Università di Pisa
Dipartimento di Informatica

Technical Report: TR-08-27

Minimization Algorithm for Symbolic Bisimilarity

Filippo Bonchi Ugo Montanari

December 17, 2008

Address: Largo B. Pontecorvo 3, 56127 Pisa, Italy. Tel: +39 050 2212700 Fax: +39 050 2212726

Minimization Algorithm for Symbolic Bisimilarity

Filippo Bonchi and Ugo Montanari

Dipartimento di Informatica, Università di Pisa

Abstract. The operational semantics of interactive systems is usually described by labeled transition systems. Abstract semantics is defined in terms of bisimilarity that, in the finite case, can be computed via the well-known *partition refinement algorithm*. However, the behaviour of interactive systems is in many cases infinite and thus checking bisimilarity in this way is unfeasible. *Symbolic semantics* allows to define smaller, possibly finite, transition systems, by employing symbolic actions and avoiding some sources of infiniteness. Unfortunately, the standard partition refinement algorithm does not work with symbolic bisimilarity. In a previous paper, we have introduced an abstract theory of symbolic semantics. In this paper, we introduce a partition refinement algorithm for symbolic bisimilarity.

1 Introduction

The operational semantics of interactive system is usually specified by labeled transition systems (LTSs). Behavioural equivalence is often defined as bisimilarity, namely the largest bisimulation. Many efficient algorithms and tools for bisimulation checking in the finite case have been developed [26, 8, 9]. Among these, the *partition refinement algorithm* [12, 20] is the best known: first it generates the state space of the LTS (i.e., the set of reachable states); then, it creates a partition equating all the states and then, iteratively, refines this partitions by splitting non equivalent states. At the end, the resulting partition equates all and only the bisimilar states.

Most importantly, the same algorithm can be used to construct the *minimal automaton*, that is the smallest (in terms of states and transitions) LTS amongst all those bisimilar. Construction of minimal automata allows to model check efficiently for several properties by eliminating redundant states once and for all. In fact most model checking logics are *adequate w.r.t. bisimilarity*, namely a formula holds in the given system iff it holds in its minimal representative.

In practical cases, *compositionality* is also very relevant, since it is the key to master complexity. Then a fundamental property is that bisimilarity be a congruence. When this is not the case, behavioural equivalence is defined either as the *largest congruence contained into bisimilarity* [15] or as the *largest bisimulation that is also a congruence* [19]. In this paper we focus on the largest bisimulation congruence and we call it *saturated bisimilarity*. Indeed it coincides

with ordinary bisimilarity on the *saturated transition system*, that is obtained by 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 π-calculus [16] are not preserved under substitution (and thus under input prefixes), in [24] 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, 11], mobile ambients calculus [6, 14] and (explicit [27]) fusion calculus [21].

The definition of saturated bisimilarity as ordinary bisimulation on the saturated LTS, while in principle operational, often makes infinite state the portion of LTS reachable by any nontrivial agent, and in any case (e.g. for the open π-calculus) is very inefficient, since it introduces a large number of additional states and transitions. Inspired by Hennessy and Lin [10], who introduced a *symbolic* semantics of value passing calculi, Sangiorgi defines in [24] a symbolic transition system and symbolic bisimilarity that efficiently characterizes open bisimilarity. After this, many formalisms have been equipped with a symbolic semantics.

In [5], 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 performs such transition. Moreover, a certain *derivation relation* $\vdash$ amongst the transitions of a systems 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 retrievied via the symbolic LTS.

However, the ordinary bisimilarity over the symbolic transition system does not coincide with 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, algorithms and tools developed for bisimilarity cannot be reused for symbolic bisimilarity. Inspired by [22, 17] who developed ad hoc partition refinement algorithm for open and asynchronous bisimilarity, in this paper we introduce a generical *symbolic partition refinement algorithm*, relying on the theoretical framework presented in [5]. The algorithm is based on the notion of *redundant symbolic transitions*. Intuitively, a symbolic transition $p \xrightarrow{c_2,\alpha_2}_\beta q$ is redundant if there exists another symbolic transition $p \xrightarrow{c_1,\alpha_1}_\beta p_1$ that logically implies it, that is $p \xrightarrow{c_1,\alpha_1}_\beta p_1 \vdash p \xrightarrow{c_2,\alpha_2} p_2$ and q is bisimilar to p_2. Now, if we consider the LTS having only not-redundant transitions, the ordinary notion of bisimilarity coincides with saturated bisimilarity. Thus, in principle, we could remove all the redundant transitions and then check bisimilarity with the stan-

dard partition refinement algorithm. But, how can we decide which transitions are redundant, if redundancy itself depends on bisimilarity?

Our solution consists in computing bisimilarity and redundancy *at the same time*. In the first step, the algorithm considers all the states bisimilar and all the transitions (that are potentially redundant) as redundant. At any iteration, states are distinguished according to (the current estimation of) not-redundant transitions and then not-redundant transitions are updated according to the new computed partition.

The main peculiarity of the algorithm is that in the initial partition, we have to insert not only the reachable states, but also those that are needed to check redundancy. This is clearly formalized in our general algorithm, but it is even more evident when considering the *coalgebras* [23] underlying this algorithm. [1]

During the whole paper we will use as running examples open Petri nets [13, 2]. In [5], we have shown that also asynchronous and open π calculus are instances of our general theory, and thus our algorithm works also in these cases. However, in these calculi, the symbolic transition system is infinite whenever a processes can create infinitely many names. Thus in order to apply our algorithm in these cases, we need to extend our approach to HD-Automata [18]. This is left as future work.

The background of the paper consists in Section 2 and Section 3. In the former we show the partition refinement algorithm in the case of CCS, while in the latter we recall the theoretical framework for symbolic bisimilarity of [5]. In Section 4, we show that both saturated and symbolic bisimilarity cannot be checked through ordinary partition refinement. In Section 5, we introduce redundant transitions that will be fundamental for the symbolic partition refinement algorithm shown in Section 6.

2 Partition Refinement and Minimal Automaton

In CCS [15], bisimilarity ($\sim$) is defined as the largest bisimulation relation, i.e., the largest relation R such that $R \subseteq \mathbf{F}(R)$ where $\mathbf{F}$ is a function such that for each relation R, $p\,\mathbf{F}(R)\,q$ iff

- if $p \xrightarrow{a} p'$ then $q \xrightarrow{a} q'$ and $p'Rq'$,
- if $q \xrightarrow{a} q'$ then $p \xrightarrow{a} p'$ and $p'Rq'$.

Since $\mathbf{F}$ is monotonic for set inclusion, $\sim\, = \bigcup\{R \mid R \subseteq \mathbf{F}(R)\}$ follows from standard results on fixed point theory. Moreover, $\sim$ is itself a fix point of $\mathbf{F}$, i.e., $\sim = \mathbf{F}(\sim)$. Alternatively, bisimilarity can be characterized as the limit of a decreasing chains of relations (none of them is a bisimulation) starting with the universal relation. Hereafter, we use κ to denote ordinals numbers, $\kappa + 1$ for

[1] [4] introduced a class of categorical models for symbolic bisimilarity that are *structured coalgebras* [7] instead of the more ordinary *bialgebras* [25]. In the latter, we can characterize bisimilarity abstracting away from the algebraic structure, while in the former we cannot. This is the reason why our algorithm relies on the algebraic structure, i.e., on the states that are not reachable but that are needed to check redundancy.

successor of κ, λ for limits ordinals and $\mathcal{O}$ for the class of all ordinals. Formally, the *terminal sequence* is defined for each ordinal κ as follow,

$$\sim^0 = \{\mathcal{P} \times \mathcal{P}\} \quad \sim^{\kappa+1} = \mathbf{F}(\sim^\kappa) \quad \sim^\lambda = \bigcap_{\kappa<\lambda} \sim^\kappa$$

where $\mathcal{P}$ is the set of all CCS processes. Bisimilarity coincides with the limit of the terminal sequence.

Proposition 1. $\sim = \bigcap_{\kappa \in \mathcal{O}} \sim^\kappa$

Given a set S, a *partition* of S is a set of *blocks*, i.e. subsets of S, that are all disjoint and whose union is S. A partition on S represents an equivalence relation, where equivalent elements belong to the same block. In the following, given a function $\mathbf{G}$ on equivalence relations, we denote by $\overline{\mathbf{G}}$ the corresponding function on partitions.

The characterization of bisimilarity through the terminal sequence suggests a procedure for checking bisimilarity of a set of initial states IS. First of all, we compute $IS^\star$, i.e., the set of all states that are reachable from IS. Then we create the partition P^0 where all the elements of $IS^\star$ belongs to the same block. After the initialization, we iteratively refine the partitions by using the function $\overline{\mathbf{F}}$ (i.e., the function equivalent to $\mathbf{F}$ on partitions): two states p and q belong to the same block in P^{n+1}, if and only if whenever $p \xrightarrow{a} p'$ then $q \xrightarrow{a} q'$ with p' and q' in the same block of P^n and viceversa.

Algorithm 1 `Partition-Refinement(`IS`)`

Initialization

1. $IS^\star$ is the set of all processes reachable from IS,
2. $P^0 := \{IS^\star\}$,

Iteration $P^{n+1} := \overline{\mathbf{F}}(P^n)$,
Termination If $P^n = P^{n+1}$ then return P^n.

The algorithm terminates whenever two consecutive partitions are equivalent. In such partition two states belong to the same block if and only if they are bisimilar.
Notice that since $\mathbf{F}$ is monotonic, any iteration splits blocks and never fuse them. For this reason if $IS^\star$ is finite, the algorithm terminates in at most $|IS^\star|$ iterations.

Proposition 2. *If $IS^\star$ is finite, then the algorithm terminates and the resulting partition equates all and only the bisimilar state.*

The partition refinement algorithm allows not only to check bisimilarity of a set of states, but also to build the *minimal automaton* of a certain state p. Intuitively, the minimal automaton is a labeled transition systems where all the bisimilar

states are identified. Hereafter, given a set A and an equivalence relation R, we write $A_{|R}$ to denote the set of equivalence classes of A w.r.t. R. Moreover, given $p \in A$, $[p]_R$ denotes the equivalence class of p w.r.t. R.

Definition 1 (Minimal Automaton). *Let $\{p\}^\star$ be the set of states reachable from the state p. The* minimal automaton of p *(denoted by $MA(p)$) is a triple $\langle i, M, tr_M \rangle$:*

- *the initial state i is equal to $[p]_\sim$,*
- *$M = \{p\}^\star_{|\sim}$ is the set of equivalence classes of $\sim$,*
- *tr_M is the transition relation defined according to the following rule.*

$$\frac{q \xrightarrow{a} r}{[q]_\sim \xrightarrow{a}_M [r]_\sim}$$

Proposition 3. *$p \sim q$ if and only if $MA(p)$ is isomorphic to $MA(q)$.*

If the set of states reachable from p is finite, we can employ the partition refinement algorithm to build the minimal automaton of p. We have just to quotient the set of reachable states $\{p\}^\star$ with the partition returned by the `Partition-Refinement(`$\{p\}$`)`.

3 Saturated and Symbolic Semantics

In this section we recall the general framework for symbolic bisimilarity that we have introduced in [5]. As running example, we will use open Petri nets [13, 2]. However, our theory has as special cases the abstract semantics of several formalisms such as open [24] and asynchronous [1] π-calculus and explicit fusion calculus [27].

3.1 Saturated Semantics

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

The first definition of the theoretical framework presented in [5] 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.

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

Definition 2 (Context Interactive System). *A* context interactive system $\mathcal{I}$ *is a quadruple* $\langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ *where:*

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

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

The abstract semantics of interactive systems is usually defined through behavioural equivalences. In [5] we proposed a general notion of bisimilarity that generalizes the abstract semantics of a large variety of formalisms. 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 3 (Saturated Bisimilarity). *Let* $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ *be a context interactive system. Let* $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$ *be an* S *sorted family of symmetric relations.* R *is a* saturated bisimulation *iff,* $\forall s, t \in S$, $\forall c \in \Sigma_{s,t}$, *whenever* pR_sq*:*

- $c_{\mathbb{A}}(p)\, R\, c_{\mathbb{A}}(q)$,
- *if* $p \xrightarrow{o} p'$, *then* $q \xrightarrow{o} q'$ *and* $p' R q'$.

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

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

Proposition 4. $\sim^S$ *is the coarsest bisimulation congruence.*

3.2 Running example: open Petri nets

Differently from process calculi, Petri nets have not a widely known interactive behaviour. Indeed they model concurrent systems that are closed, in the sense that they do not interact with the environment. *Open nets* [13, 2] are P/T Petri nets that can interact by exchanging tokens on *input* and *output places*.

Given a set X, we write $X^{\oplus}$ for the free commutative monoid over X. A multiset $m \in X^{\oplus}$ is a function from X to ω (the set of natural 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 $\forall x \in X$, $m_1 \oplus m_2(x) = m_1(x) + m_2(x)$, while $m_1 \cap m_2$ as $\forall x \in X$, $m_1 \cap m_2(x) = min\{m_1(x), m_2(x)\}$. We write $\varnothing$ to denote both the empty set and the empty multiset, while we write ab^n to denote the multiset that associates multiplicity 1 to a and multiplicity n to b.

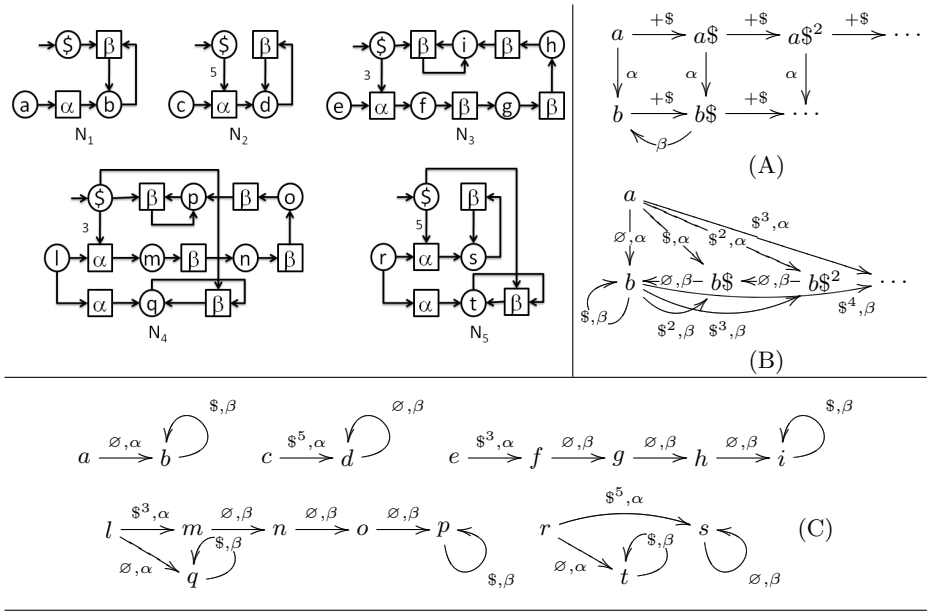


Fig. 1. The open nets N_1, N_2, N_3, N_4 and N_5.(A)Part of the infinite transition system of $\langle N_1, a\rangle$. (B)Part of the infinite saturated transition system of $\langle N_1, a\rangle$.(C)The symbolic transition systems of $\langle N_1, a\rangle$,$\langle N_2, c\rangle$,$\langle N_3, e\rangle$,$\langle N_4, l\rangle$ and $\langle N_5, r\rangle$.

Definition 4 (Open net). *An* open net *is a tuple* $N = (S, T, pre, post, l, I, O)$ *where* S *and* T *are the sets of places and transitions* ($S \cap T = \varnothing$)*;* $pre, post : T \to S^{\oplus}$ *are functions mapping each transition to its pre- and post-set;* $l : T \to \Lambda$ *is a labeling function* (Λ *is a set of labels) and* $I, O \subseteq S$ *are the sets of input and output places. A* marked open net *is a pair* $\langle N, m\rangle$ *where* N *is an open net and* $m \in S^{\oplus}$ *is a marking.*

Fig.1 shows five open nets where, as usual, circles represents places and rectangles transitions (labeled with α, β). Arrows from places to transitions represent *pre*, while arrows from transitions to places represent *post*. Input places are denoted by ingoing edges, thus the only input place of N_1 is \$. To make examples easier, hereafter we only consider *open input nets*, i.e., open nets without output places[3].

The operational semantics of marked open nets is expressed by the rules on Table 1 where, in order to make lighter the notation, we use ${}^{\bullet}t$ and $t^{\bullet}$ to denote $pre(t)$ and $post(t)$ and we avoid to put brackets around the marked net $\langle N, m\rangle$. The rule (TR) is the standard rule of P/T nets (seen as multisets rewriting). 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. Fig.1(A) shows part of the infinite transition system of $\langle N_1, a\rangle$.

[3] The extension to nets with output places is trivial as shown in [5].

$$(\text{TR})\frac{t \in T \quad l(t) = \lambda \quad m = {}^\bullet t \oplus c}{N, m \xrightarrow{\lambda} N, t^\bullet \oplus c} \qquad (\text{IN})\frac{i \in I_N}{N, m \xrightarrow{+i} N, m \oplus i}$$

Table 1. Operational Semantics of marked open nets

The abstract semantics (denoted by $\sim^N$) is defined in [3] as the ordinary bisimilarity over such an LTS. It is worth noting that $\sim^N$ can be seen as an instance of saturated semantics, where multisets over open places are contexts and transitions are only those generated by the rule (TR).

In the following we formally define $\mathcal{N} = \langle (S^{\mathcal{N}}, \Sigma^{\mathcal{N}}), \mathbb{N}, \Lambda, tr_{\mathcal{N}} \rangle$ that is the context interactive system of all open nets (labeled over the set of labels Λ).

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

- $S^{\mathcal{N}} = \{I \mid I \text{ is a set of places}\}$,
- $\forall I \in S^{\mathcal{N}}$, $\Sigma^{\mathcal{N}}_{I,I} = I^{\oplus}$, $id_I = \varnothing$ and $i_1 \circ i_2 = i_1 \oplus i_2$.

Intuitively sorts of the signature are sets of input places I, while operators of $\Sigma^{\mathcal{N}}$ are multisets of tokens on the input places. 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 nets $\langle N_1, a \rangle$ has interface $\{\$\}$. Let us define the $(S^{\mathcal{N}}, \Sigma^{\mathcal{N}})$-algebra $\mathbb{N}$. For any sort I, the carrier set N_I contains all the marked open nets with interface I. Any operator $i \in \Sigma_{I,I}$ is defined as the function that 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 1.

In [5], it is proved that saturated bisimilarity for $\mathcal{N}$ coincides with $\sim^N$.

In the remainder of the paper we will use as running example the open nets in Fig.1. 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 asynchronous π-calculus [1] where it holds that $a(x).(\overline{a}x \mid p) + \tau.p \sim \tau.p$. Appendix A proves that $l \sim^N a$.

3.3 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 [5], 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.

Definition 5 (Symbolic Context Transition System). *A* symbolic context transition system *(*SCTS *for short) for a system* $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ *is a transition system* $\beta \subseteq |\mathbb{A}| \times \Sigma \times O \times |\mathbb{A}|$.

In [5], we have introduced a SCTS for open nets. Intuitively the symbolic transition $N, m \xrightarrow{i,\lambda}_\eta N, m'$ is possible if and only if $N, m \oplus i \xrightarrow{\lambda}_{\mathcal{N}} N, m'$ and i is the smallest multiset (on input places) allowing such transition. This SCTS 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 Fig.1 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 $N_2, cd\$^2 \xrightarrow{\$^3,\alpha}_\eta N_2, dd$. Fig.1(C) shows symbolic transition systems of marked open nets discussed in the previous subsection.

Definition 6 (Inference System). *An inference system* $\mathcal{R}$ *for a context interactive system* $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ *is a set of rules of the following format, where* $s, t \in S$, $o, o' \in O$, $c \in \Sigma_{s,s'}$ *and* $d \in \Sigma_{t,t'}$.

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

The above rule states that all processes with sort s that perform a transition with observation o going into a state q_t with sort t, when inserted into the context c can perform a transition with the observation o' going into $d(q_t)$.

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 $d \circ c \xrightarrow[o'']{o} d' \circ c'$ if $d \circ c$ and $d' \circ c'$ are

defined. Given an inference system $\mathcal{R}$, $\Phi(\mathcal{R})$ is the set of all the rules derivable from $\mathcal{R}$ together with the identities rules ($\forall o \in O$ and $\forall s, t \in S$, $id_s \ \frac{o}{o} \ id_t$).

Definition 7 (Derivations, soundness and completeness). *Let $\mathcal{I}$ be a context interactive system, β an* SCTS *and $\mathcal{R}$ an inference system.*

We say that $p \xrightarrow{c_1,o_1} p_1$ derives *$p \xrightarrow{c_2,o_2} p_2$ in $\mathcal{R}$ (written $p \xrightarrow{c_1,o_1} p_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,o_2} p_2$) if there exist $d, e \in \Sigma$ such that $d \ \frac{o_1}{o_2} \ e \in \Phi(\mathcal{R})$, $d \circ c_1 = c_2$ and $e_{\mathbb{A}}(p_1) = p_2$.*

We say that β and $\mathcal{R}$ are sound and complete *w.r.t. $\mathcal{I}$ if*

$$p \xrightarrow{c,o}_S q \text{ iff } p \xrightarrow{c',o'}_\beta q' \text{ and } p \xrightarrow{c',o'}_\beta q' \vdash_{\mathcal{R}} p \xrightarrow{c,o}_S q.$$

A sound and complete SCTS could be considerably smaller than the saturated transition system, but still containing all the information needed to recover $\sim^S$. Note that the ordinary bisimilarity over SCTS (hereafter called *syntactical bisimilarity* and denoted by $\sim^W$) is usually stricter than $\sim^S$. As an example consider the marked open nets a and l. These are not syntactically bisimilar, since $l \xrightarrow{\$^3,\alpha}_\eta m$ while a cannot (Fig.1(C)). However, they are saturated bisimilar, as discussed in the previous subsection. In order to recover $\sim^S$ through the symbolic transition system we need a more elaborated definition of bisimulation.

Definition 8 (Symbolic Bisimilarity). *Let $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ be an interactive system, $\mathcal{R}$ be a set of rules and β be a symbolic transition system. Let $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$ be an S sorted family of symmetric relations. R is a* symbolic bisimulation *iff $\forall s \in S$, whenever pR_sq:*

- *if $p \xrightarrow{c,o}_\beta p'$, then $q \xrightarrow{c_1,o_1}_\beta q'_1$ and $q \xrightarrow{c_1,o_1}_\beta q'_1 \vdash_{\mathcal{R}} q \xrightarrow{c,o} q'$ and $p'Rq'$.*

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

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

In the remainder of this section we focus on open Petri nets. The inference system $\mathcal{R}_\mathcal{N}$ is defined by the following parametric rule.

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

The intuitive meaning of this rule 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 $N, m \xrightarrow{i_1,\lambda_1} N, m_1 \vdash_{\mathcal{R}_\mathcal{N}} N, m \xrightarrow{i_2,\lambda_2} N, m_2$ if and only if $\lambda_2 = \lambda_1$ and there exists a multiset x on input places such that $i_2 = i_1 \oplus x$ and $m_2 = m_1 \oplus x$. *For all the nets N_i of our example, this just means that for all observations λ and for all multisets m, n, we have that $\langle N_i, m \rangle \xrightarrow{\$^i,\lambda}_\eta \langle N_i, n \rangle \vdash_{\mathcal{R}_\mathcal{N}} \langle N_i, m \rangle \xrightarrow{\$^{i+j},\lambda} \langle N_i, n\$^j \rangle$.*

From the above characterization of $\vdash_{\mathcal{R}_\mathcal{N}}$ and from Def.8, the definition of symbolic bisimulation for open nets follows.

Definition 9 (Symbolic bisimulation for marked open nets). *A symmetric relations R is a* symbolic bisimulation for nets *iff, whenever* $\langle N_1, m_1\rangle R\langle N_2, m_2\rangle$

- N_1 *and* N_2 *have the same sets of input places,*
- *if* $\langle N_1, m_1\rangle \xrightarrow{i,\lambda}_\eta \langle N_1, m_1'\rangle$ *then* $\exists i_1, x \in I^\oplus$ *such that:*
  $i = i_1 \oplus x$, $\langle N_2, m_2\rangle \xrightarrow{i_1,\lambda}_\eta \langle N_2, m_2'\rangle$ *and* $\langle N_1, m_1'\rangle R\langle N_2, m_2' \oplus x\rangle$.

In [5] we have shown that $\mathcal{R}_\mathcal{N}$ and η are sound and complete w.r.t. $\mathcal{N}$. For this reason, we can prove that two marked nets are bisimilar, by showing a symbolic bisimulation that relates them. An example is in Appendix A.

4 Saturated and Symbolic Terminal Sequences

In this section we introduce the terminal sequence for saturated and symbolic bisimilarity. They are almost straightforward adaptation of the terminal sequence for ordinary bisimilarity presented in Section 2. Hereafter we always implicitly refer to a context interactive system $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr\rangle$, a symbolic transition system β and an inference system $\mathcal{R}$, such that β and $\mathcal{R}$ are sound and complete w.r.t. $\mathcal{I}$.

The *saturated terminal sequence* is defined as follows,

$$\sim_S^0 = \{A_s \times A_s \mid s \in S\} \quad \sim_S^{\kappa+1} = \mathbf{SAT}(\sim_S^\kappa) \quad \sim_S^\lambda = \bigcap_{\kappa<\lambda} \sim_S^\kappa$$

where **SAT** is a function on S indexed families of relations such that, for all $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$, $p\mathbf{SAT}(R)q$ iff

- if $p \xrightarrow{c,o}_S p'$, then $q \xrightarrow{c,o}_S q'$ and $p'Rq'$,
- if $q \xrightarrow{c,o}_S q'$, then $p \xrightarrow{c,o}_S p'$ and $p'Rq'$.

The only difference w.r.t. the terminal sequence of ordinary bisimilarity is in the fact that we consider S indexed families of relations (recall that S is the set of sorts, and A_s is carrier set of sort s of the algebra $\mathbb{A}$).

It is easy to see that **SAT** is monotonic w.r.t. (indexed) set inclusion. From classical results of fixed point theory (analogously to ordinary bisimilarity), we have that saturated bisimilarity is the limit of the saturated terminal sequence.

Proposition 5. $\sim^S = \bigcap_{\kappa\in\mathcal{O}} \sim_S^\kappa$

The following lemma will be fundamental to prove the correctness of our algorithm. Its proof can be found in the appendix.

Lemma 1. $\forall\kappa \in \mathcal{O}$, $\sim_S^\kappa$ *is a congruence.*

In Section 2, we have shown that the terminal sequence for ordinary bisimilarity provides an effective procedure for computing bisimilarity. We would like to apply the same intuition to the saturated terminal sequence but, unfortunately, the saturated transition system is usually infinite since it considers all the possible

contexts. Instead of using the saturated transition system, we work with the symbolic transition system.

The *symbolic terminal sequence* is defined as follows,

$$\sim^0_{SYM} = \{A_s \times A_s \mid s \in S\} \quad \sim^{\kappa+1}_{SYM} = \mathbf{SYM}(\sim^{\kappa}_{SYM}) \quad \sim^{\lambda}_{SYM} = \bigcap_{\kappa<\lambda} \sim^{\kappa}_{SYM}$$

where **SYM** is a function on S indexed families of relations such that, for all $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$, $p\mathbf{SYM}(R)q$ iff

- if $p \xrightarrow{c,o}_\beta p'$, then $q \xrightarrow{c_1,o_1}_\beta q'_1$ and $q \xrightarrow{c_1,o_1}_\beta q'_1 \vdash_{\mathcal{R}} q \xrightarrow{c,o} q'$ and $p'Rq'$,
- if $q \xrightarrow{c,o}_\beta q'$, then $p \xrightarrow{c_1,o_1}_\beta p'_1$ and $p \xrightarrow{c_1,o_1}_\beta p'_1 \vdash_{\mathcal{R}} q \xrightarrow{c,o} q'$ and $p'Rq'$.

The only difference w.r.t. saturated terminal sequence is in the inductive case, where we use **SYM** instead of **SAT**. We could prove that the two terminal sequences coincide, but this is useless for our aims. In fact, imagine to adapt the partition refinement algorithm (Section 2) to the symbolic terminal sequence, by replacing the function $\overline{\mathbf{F}}$ with $\overline{\mathbf{SYM}}$. Suppose to apply this imaginary algorithm to the set of initial states $IS = \{a, r\}$ (Fig.1(C)). The set of states reachable through the symbolic transition system is $IS^\star = \{a, b, r, s, t\}$. The initial partition would be $P^0 = \{a, b, r, s, t\}$. At the first iteration, when computing $\overline{\mathbf{SYM}}(P^0)$, we should decide about splitting a and r. Since $r \xrightarrow{\$^5,\alpha}_\eta s$ and $a \xrightarrow{\varnothing,\alpha}_\eta b$ and $a \xrightarrow{\varnothing,\alpha}_\eta b \vdash_{\mathcal{R}_\mathcal{N}} a \xrightarrow{\$^5,\alpha} b\$^5$, we should check if $b\$^5$ and s belong to the same block in P^0. Unfortunately $b\$^5$ is not reachable from $IS = \{a, r\}$, and thus it has not been inserted into the initial partition. One could conjecture that we could add all those states as the above $b\$^5$. However, this immediately brings to add infinitely many states.

5 Redundant Transitions

In Section 3, we have shown that syntactical bisimilarity ($\sim^W$), i.e. the ordinary bisimilarity on the symbolic transition system, does not coincide with $\sim^S$. Here we show that this is due to the presence of *redundant transitions*. In order to better explain this phenomenon, we have to show an important property of $\vdash_{\mathcal{R}}$.

Lemma 2. *$\forall p, q$, if $p \xrightarrow{c_1,d_1} p_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,d_2} e_{\mathbb{A}}(p_1)$, then $q \xrightarrow{c_1,d_1} q_1 \vdash_{\mathcal{R}} q \xrightarrow{c_2,d_2} e_{\mathbb{A}}(q_1)$.*

Now, consider a process p that performs only the symbolic transitions $p \xrightarrow{c_1,o_1}_\beta p_1$ and $p \xrightarrow{c_2,o_2}_\beta p_2$ such that $p \xrightarrow{c_1,o_1}_\beta p_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,o_2} e_{\mathbb{A}}(p_1)$ and $p_2 \sim^S e_{\mathbb{A}}(p_1)$. The transition $p \xrightarrow{c_2,o_2}_\beta p_2$ is *redundant* and it makes $\sim^W$ different from $\sim^S$. Indeed, take a process q that performs only $q \xrightarrow{c_1,o_1}_\beta q_1$ such that $p_1 \sim^S q_1$. Clearly p and q are not syntactically bisimilar, because $p \xrightarrow{c_2,o_2}_\beta p_2$ while q cannot. However, $p \sim^S q$, because $q \xrightarrow{c_2,o_2}_S e_{\mathbb{A}}(q_1)$ (assuming that β and $\mathcal{R}$ are sound and complete and by Lemma 2) and, $p_2 \sim^S e_{\mathbb{A}}(p_1) \sim^S e_{\mathbb{A}}(q_1)$ (since $\sim^S$ is a congruence).

As an example consider the symbolic transition system of l (Fig.1). $l \xrightarrow{\varnothing,\alpha}_\eta q$ and $l \xrightarrow{\$^3,\alpha}_\eta m$. Moreover, $l \xrightarrow{\varnothing,\alpha}_\eta q \vdash_{\mathcal{R}_\mathcal{N}} l \xrightarrow{\$^3,\alpha} q\$^3$ and $q\$^3 \sim^S m$. Now consider a. $a \xrightarrow{\varnothing,\alpha}_\eta b$. Clearly $l \not\sim^W a$ but they are saturated bisimilar (as shown in Section 3). Redundant transitions appears in many other formalisms. In Appendix B, we shows some of them in the case of open and asynchronous π calculus.

Definition 10 (Redundant Transition). *Let $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr\rangle$ be a context interactive system, $\mathcal{R}$ be an inference system and X be an S sorted family of relations. Let $p \xrightarrow{c_1,o_1} p_1$ and $p \xrightarrow{c_2,o_2} p_2$ be two different transitions. We say that the former* dominates *the latter in X (written $p \xrightarrow{c_1,o_1} p_1 \prec_X p \xrightarrow{c_2,o_2} p_2$) if and only if $p \xrightarrow{c_1,o_1} p_1 \vdash_\mathcal{R} p \xrightarrow{c_2,o_2} e_\mathbb{A}(p_1)$ and $p_2\, X\, e_\mathbb{A}(p_1)$. A transition is* redundant *w.r.t. X if it is dominated in X by another transition. Otherwise, it is* irredundant.

In the remainder of this section, we introduce another characterization of saturated bisimilarity that only checks irredundant symbolic transitions. The minimization algorithm that we will present in Section 6 relies on this notion.

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

- *if $p \xrightarrow{c,o}_\beta p'$ is irredundant in R, then $q \xrightarrow{c,o}_\beta q'$ and $p'Rq'$.*

We write $p \sim_s^{NR} q$ iff an irredundant bisimulation R such that pR_sq exists.

Theorem 2 states that $\sim^{NR}=\sim^S$. However, in order to have such correspondence, we have to add a constraint to our theory. Indeed, according to the actual definition of context interactive systems, there could exist infinite descending chains like: $\cdots \prec_R p \xrightarrow{c_2,o_2} p_2 \prec_R p \xrightarrow{c_1,o_1} p_1$. In this chain, all the transitions are redundant and thus none of them is considered when checking irredundant bisimilarity.

Definition 12. *A context interactive system is* well-founded *w.r.t. $\mathcal{R}$ if and only if for all relations R there are no infinite descending chains of $\prec_R$.*

All the examples that we have shown in [5] are well-founded. In particular $\mathcal{N}$ is well founded w.r.t. $\mathcal{R}_\mathcal{N}$. Indeed, for all relations R, $m \xrightarrow{i_1,\lambda_1} m_1 \prec_R m \xrightarrow{i_2,\lambda_2} m_2$ only if there exists a multiset $x \neq \varnothing$ such that $x \circ i_1 = i_2$. This means that the multiset i_1 is strictly included in the multiset i_2, and since all multisets are finite, there exist only finite descending chains of $\prec_R$.

Theorem 2. *Let $\mathcal{I}$ be a context interactive system, β an* SCTS *and $\mathcal{R}$ an inference system. If β and $\mathcal{R}$ are sound and complete w.r.t. $\mathcal{I}$ and $\mathcal{I}$ is well founded w.r.t. $\mathcal{R}$, then $\sim^{NR}=\sim^{SYM}$.*

6 A Minimization Algorithm for Symbolic Bisimilarity

In this section we first introduce the terminal sequence for irredundant bisimilarity (Subsection 6.1) then, relying on this, we introduce the symbolic partition refinement algorithm that checks saturated bisimilarity (Subsection 6.2). Finally, we prove the existence of minimal symbolic automata and we provide a procedure to compute them (Subsection 6.3). All the proofs are in Appendix C.

6.1 Irredundant Terminal Sequence

Relying on the notions introduced in the previous section, we show here the terminal sequence for irredundant bisimilarity and we prove that it coincides with saturated terminal sequence. In the remainder of the paper we always refer to a context interactive system $\mathcal{I}$, a symbolic transition system β and an inference system $\mathcal{R}$, such that β and $\mathcal{R}$ are sound and complete w.r.t. $\mathcal{I}$ and $\mathcal{I}$ is well-founded w.r.t. $\mathcal{R}$.

The *irredundant terminal sequence* is defined as follows,

$$\sim^0_{IR} = \{A_s \times A_s \mid s \in S\} \quad \sim^{\kappa+1}_{IR} = \mathbf{IR}(\sim^{\kappa}_{IR}) \quad \sim^{\lambda}_{IR} = \bigcap_{\kappa<\lambda} \sim^{\kappa}_{IR}$$

where $\mathbf{IR}$ is a function on S indexed families of relations such that, for all $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$, $p\mathbf{IR}(R)q$ iff

- if $p \xrightarrow{c,o}_{\beta} p'$ is irredundant in R, then $q \xrightarrow{c,o}_{\beta} q'$ and $p'Rq'$,
- if $q \xrightarrow{c,o}_{\beta} q'$ is irredundant in R, then $p \xrightarrow{c,o}_{\beta} p'$ and $p'Rq'$.

The only difference w.r.t. saturated terminal sequence is in the inductive case. The function $\mathbf{IR}$ is clearly different from $\mathbf{SAT}$, but they are equivalent when restricting to congruences.

Proposition 6. *Let $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$ be an S sorted family of symmetric relations. If R is a congruence, then $\mathbf{SAT}(R) = \mathbf{IR}(R)$.*

Theorem 3. $\forall \kappa \in \mathcal{O}$, $\sim^{\kappa}_{S} = \sim^{\kappa}_{IR}$.

Thus, saturated and irredundant terminal sequences are the same. However, the latter characterization allows us to define an effective procedure for checking saturated bisimilarity. This is the symbolic partition refinement algorithm that we will present in the next subsection.

6.2 Symbolic Partition Refinement

In Section 2 we have shown how the terminal sequence can be employed in order to have an effective procedure to compute bisimilarity. In this section we apply the same intuition to the irredundant terminal sequence. At the iteration n, instead of computing $\overline{\mathbf{F}}(P^n)$, we compute $\overline{\mathbf{IR}}(P^n)$: two processes p and q belong

$$(\text{IS})\frac{p \in IS \quad p \in A_s}{p \in IS^\star_s} \qquad (\text{RS})\frac{p \in IS^\star \quad p \xrightarrow{c,o}_\beta q \quad q \in A_s}{q \in IS^\star_s}$$

$$(\text{RD})\frac{p \in IS^\star \quad p \xrightarrow{c_1,o_1}_\beta q_1 \quad p \xrightarrow{c_2,o_2}_\beta q_2 \quad p \xrightarrow{c_1,o_1}_\beta q_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,o_2} e_{\mathbb{A}}(q_1) \quad e_{\mathbb{A}}(q_1) \in A_s}{e_{\mathbb{A}}(q_1) \in IS^\star_s}$$

Table 2. Closure rules

to the same block in P^{n+1}, if and only if whenever $p \xrightarrow{c,o}_\beta p'$ is not redundant in P^n then $q \xrightarrow{c,o}_\beta q'$ with p' and q' in the same block of P^n.

It is worth noting that in the computation of $\overline{\mathbf{IR}}(P^n)$ are involved also states that could be not reachable from the initial states IS. As an example consider the symbolic transition system of a and r (Fig.1(C)). The set of reachable states is $IS^\star = \{a, b, r, s, t\}$. Recall that $r \xrightarrow{\varnothing,\alpha}_\eta t \vdash_{\mathcal{R}_\mathcal{N}} r \xrightarrow{\$^5,\alpha}_\eta t\$^5$. Thus, at the generic iteration $n+1$, we need to check if the transition $j \xrightarrow{\$^5,\alpha}_\eta s$ is redundant. In order to do that we have to check if $t\$^5$ and s belong to the same block in P^n. However, the state $t\$^5$ is not reachable from $IS = \{a, r\}$.

For this reason, we have also to change the initialization step of our algorithm, by including in the set $IS^\star$ all the states that are needed to check redundancy. This is done, by using the closure rules in Table 2.The rule (RD) adds all the states that are needed to check redundancy. Indeed, if p can perform $p \xrightarrow{c_1,o_1}_\beta q_1$ and $p \xrightarrow{c_2,o_2}_\beta q_2$ such that $p \xrightarrow{c_1,o_1}_\beta q_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,o_2} e_{\mathbb{A}}(q_1)$, the latter could be redundant whenever $q_2 \sim^S e_{\mathbb{A}}(q_1)$. Thus also the state $e_{\mathbb{A}}(q_1)$ is needed. As an example, the closure of $IS = \{a, r\}$ is $IS^\star = \{a, b, r, s, t, t\$^1, t\$^2, t\$^3, t\$^4, t\$^5\}$ (Fig.2(B)). Usually, $IS^\star$ is not just a set, but an S indexed family of sets of states and for this reason the closure rules in Table 2 insert states in $IS^\star$ according to their sorts.

Algorithm 2 `Symbolic-Partition-Refinement(`IS`)`

Initialization

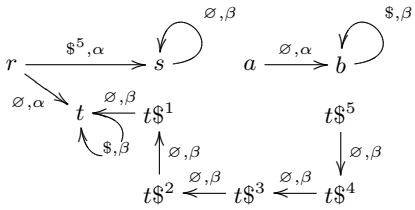
1. Compute $IS^\star$ with the rules in Table 2,
2. $P^0 := \{IS^\star_s | s \in S\}$,

Iteration $P^{n+1} := \overline{\mathbf{IR}}(P^n)$,
Termination if $P^n = P^{n+1}$ then return P^n.

Notice that in the initial partition P^0 there is one block for each sort $s \in S$. Thus P^0 equates all and the only the elements of $IS^\star$ with the same interface. Fig.2(A) shows the sequence of partitions computed by the algorithm taking as initial state $IS = \{a, r\}$. It is important to note now that in the symbolic transition system of $IS^\star$ (Fig.2(B)) the only possibly redundant transition is $r \xrightarrow{\$^5,\alpha}_\eta s$ (because $r \xrightarrow{\varnothing,\alpha}_\eta t \vdash_{\mathcal{R}_\mathcal{N}} r \xrightarrow{\$^5,\alpha} t\$^5$). Thus, in order to check redundancy,

$P^0 = \{a, b, r, s, t, t\$^1, t\$^2, t\$^3, t\$^4, t\$^5\}$
$P^1 = \{a, r\}\{b, t\}\{t\$^1, t\$^2, t\$^3, t\$^4, t\$^5, s\}$
$P^2 = \{a, r\}\{b, t\}\{t\$^1\}, \{t\$^2, t\$^3, t\$^4, t\$^5, s\}$
$P^3 = \{a, r\}\{b, t\}\{t\$^1\}, \{t\$^2\}\{t\$^3, t\$^4, t\$^5, s\}$
$P^4 = \{a, r\}\{b, t\}\{t\$^1\}, \{t\$^2\}\{t\$^3\}\{t\$^4, t\$^5, s\}$
$P^5 = \{a, r\}\{b, t\}\{t\$^1\}, \{t\$^2\}\{t\$^3\}\{t\$^4\}\{t\$^5, s\}$
$P^6 = \{a, r\}\{b, t\}\{t\$^1\}, \{t\$^2\}\{t\$^3\}\{t\$^4\}\{t\$^5\}\{s\}$
$P^7 = \{a\}\{r\}\{b, t\}\{t\$^1\}\{t\$^2\}\{t\$^3\}\{t\$^4\}\{t\$^5\}\{s\}$
$P^8 = \{a\}\{r\}\{b, t\}\{t\$^1\}\{t\$^2\}\{t\$^3\}\{t\$^4\}\{t\$^5\}\{s\}$

(A)



(B)

Fig. 2. (A)The partitions computed by `Symbolic-Partition-Refinement(`$\{a, r\}$`)`.(B)The symbolic transition systems of $\{a, r\}^\star$.

at any iteration we have only to check if $t\$^5$ and s belong to the same block. In the initial partition all the states are equivalent since they all have the same interface (recall that all the marked nets presented in Section 3 have interface \$). In P^1 there are three blocks. The states a and r are in the same block because the transition $r \xrightarrow{\$^5,\alpha}_\eta s$ is redundant since s and $t\$^5$ belong to the same block in P^0. In the second iteration, the state $t\$^1$ is separated from $\{t\$^2, t\$^3, t\$^4, t\$^5, s\}$ because the former can perform $\xrightarrow{\varnothing,\beta}_\eta \{r, b\}$ while all the others cannot. Note that a and r are still in the same block because s and $t\$^5$ belong to the same block in P^1. In each of the following iteration, a state $t\$^i$ is separated from s. In P^6, the state $t\$^5$ is separated from s and thus in P^7 the state a and r are divided because the transition $r \xrightarrow{\$^5,\alpha}_\eta s$ is not redundant anymore. Then P^8 is equivalent to P^7 and the algorithm returns such partition.

In order to prove the soundness of our algorithm we define the *irredundant terminal sequence for the set of initial states* IS,

$$\sim^0_{IS} = \sim^0_\beta \restriction IS^\star \qquad \sim^{\kappa+1}_{IS} = \mathbf{IR}(\sim^\kappa_{IS}) \qquad \sim^\lambda_{IS} = \bigcap_{\kappa<\lambda} \sim^\kappa_{IS}$$

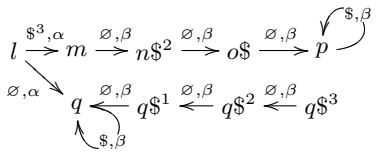
where $R \restriction A$ denotes the restriction of the relation R to the set A, $IS^\star$ is the closure of IS w.r.t. rules in Table 2.

The only difference with respect to the irredundant terminal sequence is in the first element. Here instead of taking the whole state space of $\mathcal{I}$, we restrict to $IS^\star$. The following theorem guarantees that this is enough in order to characterize the restriction of the irredundant terminal sequence to $IS^\star$. This is not trivial and it strongly relies on the fact that we close IS w.r.t. the rule (RD) in Table 2. Indeed whenever we remove such rule, it does not hold anymore.

Theorem 4. $\forall \kappa \in \mathcal{O}$, $\sim^\kappa_{ND} \restriction IS^\star = \sim^\kappa_{IS}$.

Theorem 5. *If* $\sim^\kappa_{IS} = \sim^{\kappa+1}_{IS}$, *then* $\forall k' \geq k+1$, $\sim^\kappa_{IS} = \sim^{\kappa'}_{IS}$.

Corollary 1. *If* $IS^\star$ *is finite, then the algorithm terminates and the resulting partition equates all and only saturated bisimilar states.*



$$
\begin{aligned}
P^0 &= \{l, p, q, q\$^1, o\$, q\$^2, n, q\$^3, m\} \\
P^1 &= \{l\}\{p, q\}\{q\$^1, o\$, q\$^2, n, q\$^3, m\} \\
P^2 &= \{l\}\{p, q\}\{q\$^1, o\$\}\{q\$^2, n, q\$^3, m\} \\
P^3 &= \{l\}\{p, q\}\{q\$^1, o\$\}\{q\$^2, n\}\{q\$^3, m\} \\
P^4 &= \{l\}\{p, q\}\{q\$^1, o\$\}\{q\$^2, n\}\{q\$^3, m\}
\end{aligned}
$$

Fig. 3. The partions computed by `Symbolic-Partition-Refinement(`$\{l\}$`)`

Since the algorithm applies to a lot of different formalisms, it is hard to provide a meaningful complexity analysis. However, we want to remark that the operation of checking redundancy is not expensive, since all the possible redundancies can be computed during the initialization (when using the rule (RD) of Table 2) and at any iteration, only those redundancies must be checked. Instead, the closure $IS^\star$ can be much larger than the set of reachable states (that is used by the ordinary partition refinement). Even worst, in our general theory, nothing guarantees that if the set of reachable states (through the symbolic transition system) is finite then also the closure $IS^\star$ is finite. However, we conjecture that this holds for many formalisms. The following proposition states that this holds in our running example.

Proposition 7. *Let $\mathcal{N}$, η and $\mathcal{R}_\mathcal{N}$ be the context interactive system, the symbolic transition system and the inference system for open nets that we have introduced in Section 3. Let $\langle N, m\rangle$ be a marked open net. If the symbolic transition system of $\langle N, m\rangle$ is finite, then also the closure w.r.t. rules in Table 2 is finite.*

6.3 Minimal Symbolic Automaton

In this section we introduce minimal symbolic automata, i.e. minimal automata having only irredundant symbolic transitions. We show that they are canonical representatives for equivalence classes of saturated bisimilar states. Moreover, we provide an algorithm to compute them. Hereafter, given an S sorted family of sets $X = \{X_s \mid s \in S\}$ and an S sorted family of equivalence relations $R = \{R_s \subseteq X_s \times X_s \mid s \in S\}$, we write $X_{s\,|R}$ to mean the set of equivalence classes of X_s w.r.t. R_s and for each $p \in X$, $[p]_R$ to mean the equivalence class of p w.r.t. R.

Definition 13 (Minimal Symbolic Automaton). *Let $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr\rangle$ be a context interactive system, β a symbolic transition system and $\mathcal{R}$ an inference system. Let p be a state of $\mathcal{I}$ and $\{p\}^\star = \{\{p\}^\star_s \mid s \in S\}$ be the S sorted family of sets of states obtained by closing $\{p\}$ with the rules in Table 2. The* minimal symbolic automaton of p *(denoted by $MSA(p)$) is a triple $\langle i, M, tr_M\rangle$:*

- *the initial state i is equal to $[p]_{\sim^S}$,*
- *$M = \{M_s \subseteq \{p\}^\star_{s\,|\sim^S} \mid s \in S\}$ is an S indexed family of set of equivalence classes of $\sim^S$,*
- *$tr_M \subseteq M \times \Sigma \times O \times M$ is a transition relation,*

defined according to the following two rules.

$$\frac{p \in A_s}{[p]_{\sim^S} \in M_s} \qquad \frac{[q]_{\sim^S} \in M \quad q \xrightarrow{c,o}_{\beta} r \textit{ is irredundant in } \sim^S \quad r \in A_s}{[q]_{\sim^S} \xrightarrow{c,o}_{M} [r]_{\sim^S} \qquad [r]_{\sim_S} \in M_s}$$

The leftmost rule states that the equivalence class of the initial state p belongs to the states of the minimal automaton. The other rule adds all the equivalence classes that are reachable from p trough symbolic irredundant transitions. Notice that in the minimal automaton for standard bisimilarity (Def.1) the set of states consisted of all the equivalence classes of reachable states, and thus in order to compute the minimal automata, we just needed to quotient the set of reachable states. For minimal symbolic automata we have also to remove all those states that are not reachable through irredundant symbolic transitions. As an example

Algorithm 3 `Symbolic-Minimization`(p)

1. P :=`Symbolic-Partition-Refinement`($\{p\}$),
2. Quotient $\{p\}^\star$ w.r.t. P,
3. Remove the redundant transitions,
4. Remove the states that are not reachable.

consider the symbolic transition system of l (Fig.1(C)). Fig.3 shows the closure $\{l\}^\star$ and the partitions computed by `Symbolic-Partition-Refinement`($\{l\}$). The minimal automata of l can be constructed as follows. First, we quotient the states in $\{l\}^\star$ with respect to the partition P^4 returned by the algorithm.

$$\begin{tikzcd}
\{l\} \arrow[r, "\varnothing,\alpha"] \arrow[rrr, bend left, "\$^3,\alpha"] & \{p,q\} \arrow[loop below, "\$,\beta"] & \{q\$^1, o\$\} \arrow[l, "\varnothing,\beta"'] & \{q\$^2, n\} \arrow[l, "\varnothing,\beta"'] & \{q\$^3, m\} \arrow[l, "\varnothing,\beta"']
\end{tikzcd}$$

Then we remove the redundant transitions.

$$\begin{tikzcd}
\{l\} \arrow[r, "\varnothing,\alpha"] & \{p,q\} \arrow[loop below, "\$,\beta"] & \{q\$^1, o\$\} \arrow[l, "\varnothing,\beta"'] & \{q\$^2, n\} \arrow[l, "\varnothing,\beta"'] & \{q\$^3, m\} \arrow[l, "\varnothing,\beta"']
\end{tikzcd}$$

Finally we take the set of states reachable from l: $\{l\} \xrightarrow{\varnothing,\alpha} \{p,q\} \circlearrowleft \$,\beta$. This is the minimal symbolic automaton of l. Notice that it is isomorphic to the symbolic transition system of a (Fig.1(C)). This is an alternative proof of $a \sim^S l$. Indeed, for minimal symbolic automata, analogously to minimal automata, two states p and q are saturated bisimilar if and only if their minimal symbolic automata are isomorphic, where by isomorphism we mean a bijection on states that preserves sorts, transitions and initial states.

Proposition 8. *$p \sim^S q$ if and only if $MSA(p)$ is isomorphic to $MSA(q)$.*

7 Conclusions and related works

Relying on the theoretical framework of [5], we have introduced the symbolic partition refinement algorithm that allows to efficiently check saturated bisimilarity. Moreover, we have provided a procedure for constructing the minimal symbolic automata. The existence of minimal symbolic automata is another non-trivial contribute of this work.

Our approach is absolutely general and it can be applied to a large variety of formalisms. However, when considering nominal calculi where systems are able to generate and communicate names, the symbolic transition system is often infinite. Indeed, 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 [18] are peculiar labeled transition systems that allow to garbage collect names and avoid this other source of infiniteness. As future work, we will 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 this algorithm will generalize both [22] and [17] that provide a partition refinement algorithm for open [24] and asynchronous [1] bisimilarity.

Indeed, both our approach and [22, 17] rely on *irredundant transitions*. In all these algorithms, first the closure of the set of initial states is computed by adding, not only the reachable states, but also those states that are needed to check redundancy. Then, at any iteration, only irredundant transitions are considered. In [22], the closure is called *saturated state graph* and it is computed analogously to our approach. Instead, in [17], the closure is computed by adding *negative transitions* whenever there is a possible redundancy. Roughly, if $p \overset{a}{\rightsquigarrow} q$ is a negative transition, then a transition $p \xrightarrow{a} q'$ is redundant whenever the arriving state q and q' are the same. A novel notion of bisimilarity is introduced for these kind of transition systems, but it fails to be transitive. In our context interactive systems we just rely on the algebraic structure of contexts and irredundant bisimilarity coincides with the saturated one.

Moreover, the functions on relations Φ and Φ_A, that are used during the iteration of the algorithm in [22, 17], are not monotone and, as a consequence, the convergency of the corresponding terminal sequences have to be proven by hand. Instead in our approach the function **IR** generates exactly the same terminal sequence of saturated bisimilarity and thus convergence and coincidence with saturated bisimilarity are for free. Moreover, we have shown that the correspondence between irredundant bisimilarity and saturated bisimilarity is not by chance, but because **IR** and **SAT** behaves exactly in the same way when restricted to congruences.

References

1. R. M. Amadio, I. Castellani, and D. Sangiorgi. On bisimulations for the asynchronous π-calculus. In *Proc. of CONCUR '96*, volume 1119 of *LNCS*, pages 147–162.

2. P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional semantics for open Petri nets based on deterministic processes. *M.S.C.S*, 15(1):1–35, 2005.
3. P. Baldan, A. Corradini, H. Ehrig, R. Heckel, and B. König. Bisimilarity and behaviour-preserving reconfiguration of open petri nets. In *Proc. of CALCO '07*, volume 4624 of *LNCS*, pages 126–142.
4. F. Bonchi. *Abstract Semantics by Observable Contexts*. PhD thesis, 2008.
5. F. Bonchi and U. Montanari. Symbolic semantics revisited. In *Proc. of FoSSaCS '08*, volume 4962 of *LNCS*, pages 395–412.
6. L. Cardelli and A. D. Gordon. Mobile ambients. *T.C.S.*, 240(1):177–213, 2000.
7. Andrea Corradini, Martin Große-Rhode, and Reiko Heckel. A coalgebraic presentation of structured transition systems. *T.C.S.*, 260:27–55, 2001.
8. J.C. Fernandez and L. Mounier. "on the fly" verification of behavioural equivalences and preorders. In *CAV*, pages 181–191, 1991.
9. G.L. Ferrari, S. Gnesi, U. Montanari, M. Pistore, and G. Ristori. Verifying mobile processes in the hal environment. In *CAV*, pages 511–515, 1998.
10. M. Hennessy and H. Lin. Symbolic bisimulations. *T.C.S.*, 138(2):353–389, 1995.
11. K. Honda and M. Tokoro. An object calculus for asynchronous communication. In *ECOOP '91*, volume 512 of *LNCS*, pages 133–147.
12. P. C. Kanellakis and S. A. Smolka. Ccs expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, 1990.
13. E. Kindler. A compositional partial order semantics for Petri net components. In *Petri Nets '97*, volume 1248 of *LNCS*, pages 235–252.
14. M. Merro and F. Zappa Nardelli. Bisimulation proof methods for mobile ambients. In *Proc.of ICALP '03*, volume 2719 of *LNCS*, pages 584–598, 2003.
15. R. Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.
16. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, i and ii. *Information and Computation*, 100(1):1–40, 41–77, 1992.
17. U. Montanari and M. Pistore. Finite state verification for the asynchronous pi-calculus. In *Proc. of TACAS '99*, volume 1579 of *LNCS*, pages 255–269.
18. U. Montanari and M. Pistore. An introduction to history dependent automata. *Electr. Notes Theor. Comput. Sci.*, 10, 1997.
19. U. Montanari and V. Sassone. Dynamic congruence vs. progressing bisimulation for ccs. *Fundamenta Informaticae*, 16(1):171–199, 1992.
20. Robert Paige and Robert Endre Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
21. J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *LICS*, pages 176–185, 1998.
22. M. Pistore and D. Sangiorgi. A partition refinement algorithm for the π-calculus. *Information and Computation*, 164(2):264–321, 2001.
23. J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *T.C.S.*, 249(1):3–80, 2000.
24. D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33(1):69–97, 1996.
25. D. Turi and G. D. Plotkin. Towards a mathematical operational semantics. pages 280–291. IEEE, 1997.
26. B. Victor and F. Moller. The mobility workbench - a tool for the pi-calculus. In *CAV*, pages 428–440, 1994.
27. L. Wischik and P. Gardner. Strong bisimulation for the explicit fusion calculus. In *Proc. of FoSSaCS '04*, volume 2987 of *LNCS*, pages 484–498, 2004.

A Saturated and Symbolic Bisimulation: Examples

In this section, we show that $a \sim^N l$ by exhibiting both a saturated bisimulation and a symbolic bisimulation relating them.

The following $S^{\mathcal{N}}$ sorted relation R is a saturated bisimulation.

- $R_I = \varnothing$ for all $I \in S^{\mathcal{N}}$ different from $\{\$\}$
- $R_{\{\$\}}$ = the symmetric closure of $\{(a\$^x, l\$^x), (b\$^x, q\$^x), (b\$^{x+3}, m\$^x), (b\$^{x+2}, n\$^x), (b\$^{x+1}, o\$^x), (b\$^x, p\$^x) \mid x \in \omega\}$

Another way (substantially equivalent) of proving bisimilarity is that of constructing a bisimulation over the saturated transition system. An example of saturated transition system is shown in Fig.1(B).

Instead, constructing symbolic bisimulation is usually more convenient. The following $S^{\mathcal{N}}$ sorted relation R is a symbolic bisimulation.

- $R_I = \varnothing$ for all $I \in S^{\mathcal{N}}$ different from $\{\$\}$
- $R_{\{\$\}}$ = the symmetric closure of $\{(a, l), (b, q), (b\$^3, m), (b\$^2, n), (b\$^1, o), (b, p)\}$

B Redundant Transitions: Examples

In Section 5, we have shown one example of redundant transition in the case of open Petri nets. In this appendix, we show also redundant transitions in the case of asynchronous [1] and open [24] π-calculus[4].

Consider the open π processes $p = [a = b]\tau.r$ and $q = p + [a = b][d = c]\tau.r'$ with $r \sim r'$. In the symbolic transition system defined in [24],

$$q \xrightarrow{[a=b],\tau} r \qquad q \xrightarrow{[a=b][c=d],\tau} r'$$

with $q \xrightarrow{[a=b],\tau} r \vdash_{\mathcal{R}} q \xrightarrow{[a=b][c=d],\tau} [c = d]r$ and $[c = d]r \sim [c = d]r'$ (where, roughly, $[c = d]p'$ denotes the process p' where all the occurrences of name c are substituted by d). Now take the symbolic transition system of p.

$$p \xrightarrow{[a=b],\tau} r$$

Clearly $p \not\sim^W q$, but they are open bisimilar.

For asynchronous π-calculus, consider the processes $p = a(x).(\overline{a}x \mid r') + \tau.r$ and $q = \tau.r$ with $r \sim r'$.

$$p \xrightarrow{\tau} r \qquad p \xrightarrow{a(x)} \overline{a}x \mid r'$$

We have that $p \xrightarrow{\tau} r \vdash_{\mathcal{R}} p \xrightarrow{a(x)} \overline{a}x \mid r$ and $\overline{a}x \mid r \sim \overline{a}x \mid r'$. Now q can perform only

$$q \xrightarrow{\tau} r$$

Clearly $p \not\sim^W q$ but they are bisimilar w.r.t. asynchronous bisimilarity [1].

[4] The reader is referred to [5] for a formal definition of their context interactive systems.

C Proofs

In this appendix, we report the proofs. The proof of Theorem 2 is not included, since it is a direct consequence of Theorem 3.

Lemma 1. *$\forall \kappa \in \mathcal{O}$, $\sim_S^\kappa$ is a congruence.*

Proof. For $\kappa = 0$, it is trivially true, since $p \sim_S^0 q$ if and only if, p and q have the same interface.

For a successor ordinal $\kappa + 1$, we prove that if $p \sim_S^{\kappa+1} q$ then also $c_{\mathbb{A}}(p) \sim_S^{\kappa+1} c_{\mathbb{A}}(q)$. Suppose that $c_{\mathbb{A}}(p) \xrightarrow{d,o}_S p'$ then $p \xrightarrow{d \circ c,o}_S p'$. Since $p \sim_S^{\kappa+1} q$, then also $q \xrightarrow{d \circ c,o}_S q'$ and $p' \sim_S^\kappa q'$. Moreover, $c_{\mathbb{A}}(q) \xrightarrow{d,o}_S q'$, by definition of saturated transition system.

For a limit ordinal λ, we assume that $\sim_S^\kappa$ is a congruence for all ordinals $\kappa < \lambda$ and we prove that $\sim_S^\kappa$ is a congruence. Suppose that $p \sim_S^\lambda q$ then, by definition of $\sim^S$ on limit ordinals, $p \sim_S^\kappa q$ for all $\kappa < \lambda$, and then, by ordinal induction, $c_{\mathbb{A}}(p) \sim_S^\kappa c_{\mathbb{A}}(q)$ for all $\kappa < \lambda$. Now, again by definition of $\sim^S$ on limit ordinals $c_{\mathbb{A}}(p) \sim_S^\lambda c_{\mathbb{A}}(q)$.

Lemma 2. *$\forall p, q$, if $p \xrightarrow{c_1,d_1} p_1 \vdash_{\mathcal{R}} p \xrightarrow{c_2,d_2} e_{\mathbb{A}}(p_1)$, then $q \xrightarrow{c_1,d_1} q_1 \vdash_{\mathcal{R}} q \xrightarrow{c_2,d_2} e_{\mathbb{A}}(q_1)$.*

Proof. It follows immediately from the definition of $\vdash_{\mathcal{R}}$.

Lemma 3. *If $\mathcal{I}$ is well founded w.r.t. $\mathcal{R}$ and if β and $\mathcal{R}$ are sound and complete w.r.t. $\mathcal{I}$ then, for all congruences R:*

1. *$\prec_R$ is transitive,*
2. *if $p \xrightarrow{c,o}_\beta p'$ then, exists $p \xrightarrow{c_x,o_x}_\beta p'_x$ irredundant in R that either dominates it or $c_x = c$, $o_x = o$, $p'_x = p'$*
3. *if $p \xrightarrow{c,o}_S p'$, then exists $p \xrightarrow{c_x,o_x}_\beta p'_x$ irredundant in R that either dominates it or $c_x = c$, $o_x = o$, $p'_x = p'$.*

Proof.
1. If $p \xrightarrow{c_1,o_1} p'_1 \prec_R p \xrightarrow{c_1,o_1} p'_1 \prec_R p \xrightarrow{c_3,o_3} p'_3$ then there exists $d_1 \xrightarrow[o_2]{o_1} e_1$, $d_2 \xrightarrow[o_3]{o_2} e_2 \in \Phi(\mathcal{R})$ such that $d_1 \circ c_1 = c_2$, $d_2 \circ c_2 = c_3$, $e_1(p_1)\, R\, p_2$ and $e_2(p_2)\, R\, p_3$. Then it follows that $d_2 \circ d_1 \xrightarrow[o_3]{o_1} e_2 \circ e_1 \in \Phi(\mathcal{R})$, $(d_2 \circ d_1) \circ c_1 = c_3$ and that $e_2(e_1(p_1))\, R\, p_3$, i.e., $p \xrightarrow{c_1,o_1} p'_1 \prec_R p \xrightarrow{c_3,o_3} p'_3$ (the two transitions are different otherwise the system is not well founded).
2. If $p \xrightarrow{c,o}_\beta p'$ is irredundant, then $c_x = c$, $o_x = o$, $p'_x = p'$. If it is redundant, then there exists a descending chain of $\prec_R$. Since $\mathcal{I}$ is well-founded, there is a last element of this chain that we call $p \xrightarrow{c_x,o_x}_\beta p'_x$. Since by the previous point $\prec_R$ is transitive, then $p \xrightarrow{c_x,o_x}_\beta p'_x \prec_R p \xrightarrow{c,o}_\beta p'$.
3. It follows from the completeness of β and $\mathcal{R}$ and from the previous point.

Proposition 6. *Let $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$ be an S sorted family of symmetric relations. If R is a congruence, then $\mathbf{SAT}(R) = \mathbf{IR}(R)$.*

Proof. We first prove that if $p\,\mathbf{IR}(R)\,q$ then $p\,\mathbf{SAT}(R)\,q$. If $p \xrightarrow{c,o}_S p'$, then by Lemma 3.3 there exists $p \xrightarrow{c_x,o_x}_\beta p'_x$ irredundant in R. From this, and from $p\,\mathbf{IR}(R)\,q$ follows that $q \xrightarrow{c_x,o_x}_\beta q'_x$ and $p'_x\,R\,q'_x$. Recall that by Lemma 3.3, $p \xrightarrow{c_x,o_x}_\beta p'_x$ either dominates $p \xrightarrow{c,o}_S p'$ P^n_S or $c_x = c$, $o_x = o$ and $p'_x = p'$.

- In the latter case $q \xrightarrow{c,o}_\beta q'_x$ and then $q \xrightarrow{c,o}_S q'_x$ and $q'_x\,R\,p'_x = p'$.
- In the former case, there exists $d_x, e_x \in \Sigma$ such that $d_x \circ c_x = c$, $d_x \;\frac{o_x}{o}\; e_x \in \Phi(\mathcal{R})$ and $e_{x_\mathbb{A}}(p'_x) = p'$. From this and from the fact that β and $\mathcal{R}$ are sound with respect to $\mathcal{I}$, $q \xrightarrow{c,o}_S e_{x_\mathbb{A}}(q'_x)$ follows. Moreover, $p'\,R\,(e_{x_\mathbb{A}}(p'_x))R(e_{x_\mathbb{A}}(q'_x))$.

All this proves that if $p\,\mathbf{IR}(R)\,q$ then $p\,\mathbf{SAT}(R)\,q$.

Now, we prove the other direction, i.e., if $p\,\mathbf{SAT}(R)\,q$ then $p\,\mathbf{IR}(R)\,q$. For any transition $p \xrightarrow{c,o}_\beta p'$ that is irredundant in R, we have that $p \xrightarrow{c,o}_S p'$ and since $p\,\mathbf{SAT}(R)\,q$, then there exists q' such that $p'\,R\,q'$ and $q \xrightarrow{c,o}_S q'$. By Lemma 3.3, $q \xrightarrow{c_x,o_x}_\beta q'_x$ such that either dominates $q \xrightarrow{c,o}_S q'$ in R or $c_x = c, o_x = o, q'_x = q'$.

- In the latter case $q \xrightarrow{c,o}_\beta q'$ and then $p\,\mathbf{IR}(R)\,q$.
- In the former case we have an absurdum. Indeed, if $q \xrightarrow{c_x,o_x}_\beta q'_x \prec_R q \xrightarrow{c,o}_S q'$ then there exists $d_x \;\frac{o_x}{o}\; e_x \in \Phi(\mathcal{R})$ and $e_{x_\mathbb{A}}(q'_x)P^n_\beta q'$ and $d_x \circ c_x = c$. Since $q \xrightarrow{c_x,o_x}_\beta q'_x$ then also $q \xrightarrow{c_x,o_x}_S q'_x$ and since $p\,\mathbf{SAT}(R)\,q$ then $p \xrightarrow{c_x,o_x}_S p'_x$ with $p'_x\,Rq'_x$. By Lemma 3.3 there exists a transition $p \xrightarrow{c_2,o_2}_\beta p'_2$ such that $d_2 \;\frac{o_2}{o_x}\; e_2 \in \Phi(\mathcal{R})$ and $e_{2_\mathbb{A}}(p'_2)\,R\,p'_x$ and $d_2 \circ c_2 = c_x$. From this we can derive that $d_x \circ d_2 \;\frac{o_2}{o}\; e_x \circ e_2 \in \Phi(\mathcal{R})$ and that $d_x \circ d_2 \circ c_2 = c$. Moreover, $e_{x_\mathbb{A}}(e_{2_\mathbb{A}}(p'_2))\,R\,e_{x_\mathbb{A}}(p'_x)\,R\,e_{x_\mathbb{A}}(q'_x)\,R\,q'\,R\,p'$. Summarizing $p \xrightarrow{c_2,o_2}_\beta p'_2 \prec_R p \xrightarrow{c,o}_\beta p'$, but from the previous hypothesis the latter transition is irredundant.

Summarizing $\forall p, q$, $p\,\mathbf{SAT}(R)\,q$ if and only if $p\,\mathbf{IR}(R)\,q$, i.e., $\mathbf{SAT}(R) = \mathbf{IR}(R)$.

Theorem 3. $\forall \kappa \in \mathcal{O}$, $\sim^\kappa_S = \sim^\kappa_{IR}$.

Proof. For $\kappa = 0$, $\sim^0_{IR} = \sim^0_S$ by definition.

For a successor ordinal $\kappa+1$, we have that $\sim^{\kappa+1}_{IR} = \mathbf{IR}(\sim^\kappa_{IR})$. By ordinal induction $\sim^\kappa_{IR} = \sim^\kappa_S$, and by Lemma 1 $\sim^\kappa_{IR}$ is a congruence. Thus, applying Proposition 6, we get that $\mathbf{IR}(\sim^\kappa_{IR}) = \mathbf{SAT}(\sim^\kappa_{IR}) = \mathbf{SAT}(\sim^\kappa_S) = \sim^{\kappa+1}_S$.

For a limit ordinal λ, $\sim^\lambda_{IR} = \bigcap_{\kappa<\lambda} \sim^\kappa_{IR}$ by definition. By ordinal induction, for all $\kappa < \lambda$, $\sim^\kappa_{IR} = \sim^\kappa_S$ and thus $\bigcap_{\kappa<\lambda} \sim^\kappa_{IR} = \bigcap_{\kappa<\lambda} \sim^\kappa_S = \sim^\lambda_S$.

Theorem 4. $\forall \kappa \in \mathcal{O}$, $\sim^\kappa_{IR} \upharpoonright IS^\star = \sim^\kappa_{IS}$.

Proof. For $\kappa = 0$, $\sim^0_{IS} = \sim^\kappa_{IR} \upharpoonright IS^\star$ by definition.

For a successor ordinal $\kappa + 1$, we assume that $\sim^\kappa_{IR} \upharpoonright IS^\star = \sim^\kappa_{IS}$ and we prove that $\sim^{\kappa+1}_{IR} \upharpoonright IS^\star = \sim^{\kappa+1}_{IS}$.

We first prove that $\forall p, q \in IS^\star$ if $p \sim^{\kappa+1}_{IR} q$ then also $p \sim^{\kappa+1}_{IS} q$. If $p \xrightarrow{c,o}_\beta p'$ is irredundant in $\sim^\kappa_{IS}$, then $q \xrightarrow{c,o}_\beta q'$ and $p \sim^\kappa_{IS} q$. Now there are two cases:

– if $p \xrightarrow{c,o}_\beta p'$is also irredundant in $\sim^\kappa_{IR}$, then $q \xrightarrow{c,o}_\beta q'$ and $p' \sim^\kappa_{IR} q'$ (because $p \sim^{\kappa+1}_{IR} q$). Since $p', q' \in IS^\star$ (using the rule (TR) of Table 2), by inductive hypothesis we have that $p' \sim^\kappa_{IS} q'$.
– if $p \xrightarrow{c,o}_\beta p'$ is redundant in $\sim^\kappa_{IR}$ then there exists $p \xrightarrow{c_1,o_1}_\beta p'_1$ such that $p \xrightarrow{c_1,o_1}_\beta p'_1 \vdash_{\mathcal{R}} p \xrightarrow{c,o} d_{\mathbb{A}}(p'_1)$ and $d_{\mathbb{A}}(p'_1) \sim^\kappa_{IR} p'$. Now notice that, by the rule (RS) of Table 2 $d_{\mathbb{A}}(p'_1) \in IS^\star$. Since by inductive hypothesis $\sim^\kappa_{IR}\restriction IS^\star = \sim^\kappa_{IS}$, then $d_{\mathbb{A}}(p'_1) \sim^\kappa_{IS} p'$. This means that $p \xrightarrow{c,o}_\beta p'$ is redundant in $\sim^\kappa_{IS}$, against the previous hypothesis.

Note that in the second branch of the proof the assumption that $IS^\star$ is closed w.r.t. the rule (RS) of Table 2 is fundamental. Indeed, otherwise $d_{\mathbb{A}}(p'_1)$ could not belong to $IS^\star$.

Now we can prove the other direction $\forall p, q \in IS^\star$, if $p \sim^{\kappa+1}_{IS} q$ then also $p \sim^{\kappa+1}_{IR} q$. If $p \xrightarrow{c,o}_\beta p'$ is irredundant in $\sim^\kappa_{IR}$, then it is also irredundant in $\sim^\kappa_{IS}$ (by inductive hypothesis). Thus $q \xrightarrow{c,o}_\beta q'$ and $p' \sim^\kappa_{IS} q'$ (because $p \sim^{\kappa+1}_{IS} q$). Since $p', q' \in IS^\star$ (using the rule (TR) of Table 2), by inductive hypothesis we have that $p' \sim^\kappa_{IR} q'$.

Theorem 5. *If $\sim^\kappa_{IS} = \sim^{\kappa+1}_{IS}$, then $\forall k' \geq k+1$, $\sim^\kappa_{IS} = \sim^{\kappa'}_{IS}$.*

Proof. For a successor ordinal $\kappa'+1$, we assume that $\sim^\kappa_{IS} = \sim^{\kappa_x}_{IS}$ for all $\kappa < \kappa_x < \kappa'+1$ and we prove that $\sim^{\kappa'+1}_{IS} = \sim^{\kappa'}_{IS}$. Suppose that $p \sim^{\kappa'}_{IS} q$ and $p \xrightarrow{c,o}_\beta p'$ is irredundant in $\sim^{\kappa'}_{IS}$. Then it also irredundant in $\sim^\kappa_{IS}$. Since $p \sim^{\kappa+1}_{IS} q$, $q \xrightarrow{c,o}_\beta q'$ and $p' \sim^\kappa_{IS} q'$. By hypothesis $\sim^\kappa_{IS} = \sim^{\kappa'}_{IS}$, thus $p' \sim^{\kappa'}_{IS} q'$.

For a limit ordinal λ, we assume that $\sim^\kappa_{IS} = \sim^{\kappa_x}_{IS}$ for all $\kappa < \kappa_x < \lambda$ and we prove that $\sim^\kappa_{IS} = \sim^\lambda_{IS}$. By definition $\sim^\lambda_{IS} = \bigcap_{\kappa_y < \lambda} \sim^{\kappa_y}_{IS}$. Since $\sim^\kappa_{IS} \subseteq \sim^{\kappa_y}_{IS}$ for all $\kappa_y < \lambda$, then $\bigcap_{\kappa_y < \lambda} \sim^{\kappa_y}_{IS} = \sim^\kappa_{IS}$.

Corollary 1. *If $IS^\star$ is finite, then the algorithm terminates and the resulting partition equates all and only saturated bisimilar states.*

Proof. First of all note that the algorithm perform at most $|IS^\star|$ iteration, because at every step at least a partition must be split (and partitions are never fused). Then note that at any iteration n, P^n coincides with $\sim^n_{IS}$ (in the symbolic terminal sequence for IS). Now let P^m be the partition returned by the algorithm, i.e. $P^m = P^{m+1}$. By the above observation $\sim^m_{IS} = \sim^{m+1}_{IS}$ and by Theorem 5 follows that for all ordinals $\kappa > m$, $\sim^m_{IS} = \sim^\kappa_{IS}$. By Theorem 4 follows that for all ordinals $\kappa > m$, $\sim^m_{IS} = \sim^\kappa_{IR}\restriction IS^\star$ and by Theorem 3 follows that for all ordinals $\kappa > m$, $\sim^m_{IS} = \sim^\kappa_{S}\restriction IS^\star$. Now by Proposition 5 follows that $\sim^m_{IS} = \sim^S\restriction IS^\star$.

Proposition 7. *Let $\mathcal{N}$, η and $\mathcal{R}_{\mathcal{N}}$ be the context interactive system, the symbolic transition system and the inference system for open nets that we have introduced in Section 3. Let $\langle N, m\rangle$ be a marked open net. If the symbolic transition system of $\langle N, m\rangle$ is finite, then also the closure w.r.t. rules in Table 2 is finite.*

Proof. First of all, notice that for each pairs of transitions $m \xrightarrow{i_1,\alpha_1}_\eta m_1$ and $m \xrightarrow{i_2,\alpha_2}_\eta m_2$ such that the latter is possibly dominated by the former, only one state is added to $IS^\star$. Since the symbolic transitions are finite, then only finitely many states are added by the rule (RD). However, the symbolic transitions system starting from each of these states could be infinite.

Suppose that we have added the state $n \oplus i$ where n belongs to the reachable states from m. If the symbolic transition system starting from $n \oplus i$ is infinite there are two cases. Either it is infinite branching or there is an infinite path:

$$n \oplus i \xrightarrow{j_1,\alpha_1}_\eta n_1 \xrightarrow{j_2,\alpha_2}_\eta n_2 \xrightarrow{j_3,\alpha_3}_\eta \dots$$

where $\forall x, y \in \omega$, $n_x \neq n_y \neq n \oplus i$.

In the former case, there must exists infinitely many transition that can be activated by $n \oplus i$. Since all the tokens in i are on open places, these infinitely many transition can be activated also by n and thus also the symbolic transition system of n is infinite branching, against the initial hypothesis.

In the latter case notice that if $n \oplus i \xrightarrow{j_1,\alpha_1}_\eta n_1$ then it is activated by some transition t. Let k be the multiset on open places such that $i = {}^\bullet t \oplus k$. It is easy to see that $n \xrightarrow{j_1 \oplus (i \cap {}^\bullet t),\alpha_1}_\eta n'_1$ where $n_1 = n'_1 \oplus k$. We can repeat the same argument for each transitions of the infinite path and we obtain that:

$$n \xrightarrow{j'_1,\alpha_1}_\eta n'_1 \xrightarrow{j'_2,\alpha_2}_\eta n'_2 \xrightarrow{j'_3,\alpha_3}_\eta \dots$$

where $\forall x \in \omega$ there exists k_x such that $n'_x \oplus k_x = n_x$. Now either the symbolic transition system of n is infinite (against the initial hypothesis) or there is a cycle, i.e., that for some $x, y \in \omega$,

$$n'_x = n'_{x+y} = n'_{x+2y} = n'_{x+3y} = n'_{x+4y} \dots$$

Notice that by definition of the symbolic transition system η, the series

$$k_x, k_{x+y}, k_{x+2y}, k_{x+3y} \dots$$

never grows. Now suppose that for some $u, v \in \omega$, $k_{x+uy} = k_{x+vy}$, then $n_{x+uy} = n_{x+vy}$ (against the assumption of the infinite path of $n \oplus i$). Thus the above series always strictly decreases. This means that the series

$$n_x, n_{x+y}, n_{x+2y}, n_{x+3y}, n_{x+4y} \dots$$

strictly decreases. But this is impossible in marked open nets.

Lemma 4. *If $p \sim^S q$ then:*

- *if $p \xrightarrow{c,o}_\beta p'$ and this is irredundant in $\sim^S$, then $q \xrightarrow{c,o}_\beta q'$ and this is irredundant in $\sim^S$ and $p' \sim^S q'$.*

Proof. If $p \sim^S q$, then there exists a symbolic bisimulation R such that $p\,R\,q$. If $p \xrightarrow{c,o}_\beta p'$ then $q \xrightarrow{c_1,o_1}_\beta q'_1$ such that $q \xrightarrow{c_1,o_1}_\beta q'_1 \vdash_{\mathcal{R}} q \xrightarrow{c,o} e_{\mathbb{A}}(q'_1)$ and $p'\,R\,e_{\mathbb{A}}(q'_1)$.

Now we prove that $c_1 = c$, $o_1 = o$ and $q'_1 \sim^S p'$. Indeed, if $q \xrightarrow{c_1,o_1}_\beta q'_1$ then also p will perform a transition $p \xrightarrow{c_2,o_2}_\beta p'_2$ such that $p \xrightarrow{c_2,o_2}_\beta p'_2 \vdash_{\mathcal{R}} p \xrightarrow{c_1,o_1} e'_{\mathbb{A}}(p'_2)$ and $e'_{\mathbb{A}}(p'_2) R q'_1$. Now, by Lemma 2, we have that $p \xrightarrow{c_1,o_1} e'_{\mathbb{A}}(p'_2) \vdash_{\mathcal{R}} p \xrightarrow{c,o}_\beta e_{\mathbb{A}}(e'_{\mathbb{A}}(p'_2))$, and since $\vdash_{\mathcal{R}}$ is transitive, $p \xrightarrow{c_2,o_2}_\beta p'_2 \vdash_{\mathcal{R}} p \xrightarrow{c,o}_\beta e_{\mathbb{A}}(e'_{\mathbb{A}}(p'_2))$. Moreover, $e_{\mathbb{A}}(e'_{\mathbb{A}}(p'_2)) \sim^S e_{\mathbb{A}}(q'_1) \sim^S p'$. Recall that by hypothesis $p \xrightarrow{c,o}_\beta p'$ is not dominated in $\sim^S$ and thus $p \xrightarrow{c_2,o_2}_\beta p'_2$ and $p \xrightarrow{c,o}_\beta p'$ must be the same transition. Now also $p \xrightarrow{c_1,o_1} e'_{\mathbb{A}}(p'_2)$ must be the same, otherwise

$$\cdots \prec_{\sim^S} p \xrightarrow{c,o}_\beta p' = p \xrightarrow{c_2,o_2}_\beta p'_2 \prec_{\sim^S} p \xrightarrow{c_1,o_1} e'_{\mathbb{A}}(p'_2) \prec_{\sim^S} p \xrightarrow{c,o}_\beta p'$$

Thus $c_1 = c$ and $o_1 = o$. Moreover, $q'_1 \sim^S e'_{\mathbb{A}}(p'_2) = p'$.

Summarizing, $q \xrightarrow{c_1,o_1}_\beta q'_1$ and $q'_1 \sim^S p'$. Now we have to prove that the latter transition is irredundant. By using similar arguments to those above, we can prove that if $q \xrightarrow{c_1,o_1}_\beta q'_1$ is dominated then also $p \xrightarrow{c,o}_\beta p'$ is dominated, against the initial hypothesis.

Proposition 8. *$p \sim^S q$ if and only if $MSA(p)$ is isomorphic to $MSA(q)$.*

Proof. We prove that if $p \sim^S q$ then $MSA(p)$ is isomorphic to $MSA(q)$. The converse is obvious.

Assume that the symbolic terminal sequence for $\{p\}$ converges at ordinal κ, while the the symbolic terminal sequence for $\{q\}$ converges at ordinal κ'. Then $\forall p' \in \{p\}^\star$ and $q' \in \{q\}^\star$ it holds that $[p']_{\sim^\kappa_{\{p\}}} = [p']_{\sim^S}$ and $[q']_{\sim^{\kappa'}_{\{q\}}} = [q']_{\sim^S}$.

We can construct a function f from the states of $MSA(p)$ to the state of $MSA(q)$ as follows: $f([p']_{\sim^S} = [q']_{\sim^S}$ if $p' \sim^S q'$. Notice that this is a function because if $p' \sim^S q_1$ and $p' \sim^S q_2$ then $q_1 \sim^S q_2$ and thus $[q_1]_{\sim^S} = [q_2]_{\sim^S}$. For analogous reasons, f is injective.

Moreover, notice that f is total in the states of minimal automata $MSA(p)$. Indeed, if a state p' belongs to $MSA(p)$, then it is reachable from p, and since $p \sim^S q$, there exists a state q' in $MSA(q)$ such that $p' \sim^S q'$. Converting the argumentation we can prove that f is surjective.

Thus f is an isomorphism between the states of $MSA(p)$ and $MSA(q)$.

Now we have to prove that whenever $[p']_{\sim^S} \xrightarrow{c,o}_M [p'']_{\sim^S}$ then $f([p']_{\sim^S}) \xrightarrow{c,o}_M f([p'']_{\sim^S})$.

Notice that $\forall q' \in f([p']_{\sim^S})\ p' \sim^S q'$. Now, if $[p']_{\sim^S} \xrightarrow{c,o}_M [p'']_{\sim^S}$, then $p' \xrightarrow{c,o}_\beta p''$ and this transition is irredundant in $\sim^S$. By lemma **??**, follows that $q' \xrightarrow{c,o}_\beta q''$ and $p'' \sim^S q''$ and the latter transition is not dominated in $\sim^S$. Then, by definition of $MSA(q)$, $[q']_{\sim^S} \xrightarrow{c,o}_M [q'']_{\sim^S}$. Since $p'' \sim^S q''$, then $f([p'']_{\sim^S}) = [q'']_{\sim^S}$.