Reducing Weak to Strong Bisimilarity in CCP

Andres Aristizabal[*], Luis Pino[†], Frank Valencia [‡]

Comète, LIX, Laboratoire de l'École Polytechnique associé à l'INRIA

Filippo Bonchi[‡]

ENS Lyon, Université de Lyon, LIP (UMR 5668 CNRS ENS Lyon UCBL INRIA), 46 Allée d'Italie, 69364 Lyon, France

Concurrent constraint programming (ccp) is a well-established model for concurrency that singles out the fundamental aspects of asynchronous systems whose agents (or processes) evolve by posting and querying (partial) information in a global medium. Bisimilarity is a standard behavioural equivalence in concurrency theory. However, only recently a well-behaved notion of bisimilarity for ccp, and a ccp partition refinement algorithm for deciding the strong version of this equivalence have been proposed. Weak bisimiliarity is a central behavioural equivalence in process calculi and it is obtained from the strong case by taking into account only the actions that are observable in the system. Typically, the standard partition refinement can also be used for deciding weak bisimilarity simply by using Milner's reduction from weak to strong bisimilarity; a technique referred to as *saturation*. In this paper we demonstrate that, because of its involved labeled transitions, the above-mentioned saturation technique does not work for ccp. We give an alternative reduction from weak ccp bisimilarity to the strong one that allows us to use the ccp partition refinement algorithm for deciding this equivalence.

1 Introduction

Since the introduction of process calculi, one of the richest sources of foundational investigations stemmed from the analysis of *behavioural equivalences*: in any formal process language, systems which are syntactically different may denote the same process, i.e., they have the same *observable behaviour*.

A major dichotomy among behavioural equivalences concerns *strong* and *weak* equivalences. In strong equivalences, all the transitions performed by a system are deemed observable. In weak equivalences, instead, internal transitions (usually denoted by τ) are unobservable. On the one hand, weak equivalences are more abstract (and thus closer to the intuitive notion of behaviour); on the other hand, strong equivalences are usually much easier to be checked (for instance, in [18] a strong equivalence is introduced which is computable for a Turing complete formalism).

Strong bisimilarity is one of the most studied behavioural equivalence and many algorithms (e.g., [31, 12, 13]) have been developed to check whether two systems are equivalent up to strong bisimilarity. Among these, the *partition refinement algorithm* [16] is one of the best known: first it generates the state space of a labeled transition system (LTS), i.e., the set of states reachable through the transitions; then, it creates a partition equating all states and afterwards, iteratively, refines these partitions by splitting non equivalent states. At the end, the resulting partition equates all and only bisimilar states.

[*] Supported by a CNRS/DGA grant

[†] Supported by a INRIA/DGA grant

[‡] Supported by CNRS

Submitted to:
ICE 2012

Weak bisimilarity can be computed by reducing it to strong bisimilarity. Given an LTS $\overset{\cdot}{\longrightarrow}$ labeled with actions $a, b, \ldots$ one can build $\overset{\cdot}{\Longrightarrow}$ as follows.

$$\frac{P \overset{a}{\longrightarrow} Q}{P \overset{a}{\Longrightarrow} Q} \qquad \frac{}{P \overset{\tau}{\Longrightarrow} P} \qquad \frac{P \overset{\tau}{\Longrightarrow} P_1 \overset{a}{\Longrightarrow} Q_1 \overset{\tau}{\Longrightarrow} Q}{P \overset{a}{\Longrightarrow} Q}$$

Since weak bisimilarity on $\overset{a}{\longrightarrow}$ coincides with strong bisimilarity on $\overset{a}{\Longrightarrow}$, then one can check weak bisimilarity with the algorithms for strong bisimilarity on the new LTS $\overset{a}{\Longrightarrow}$.

It is worth pointing out that an alternative presentation of $\overset{\cdot}{\Longrightarrow}$ with sequences of actions as labels is also possible [20]. Nevertheless, the resulting transition system may be infinite-branching and hence not amenable to automatic verification using standard algorithms such as partition refinement.

Concurrent Constraint Programming (ccp) [27] is a formalism that combines the traditional algebraic and operational view of process calculi with a declarative one based upon first-order logic. In ccp, processes (agents or programs) interact by *adding* (or *telling*) and *asking* information (namely, constraints) in a medium (the *store*).

Inspired by [8, 7], the authors introduced in [3] both strong and weak bisimilarity for ccp and showed that the weak equivalence is *fully abstract* with respect to the standard observational equivalence of [28]. Moreover, a variant of the partition refinement algorithm is given in [4] for checking strong bisimilarity on (the finite fragment) of concurrent constraint programming.

In this paper, first we show that the standard method for reducing weak to strong bisimilarity does not work for ccp and then we provide a way out of the impasse. Our solution can be readily explained by observing that the labels in the LTS of a ccp agent are constraints (actually, they are "the minimal constraints" that the store should satisfy in order to make the agent progress). These constraints form a lattice where the least upper bound (denoted by $\sqcup$) intuitively corresponds to conjunction and the bottom element is the constraint $true$. (As expected, transitions labeled by $true$ are internal transitions, corresponding to the τ moves in standard process calculi). Now, rather than closing the transitions just with respect to $true$, we need to close them w.r.t. all the constraints. Formally we build the new LTS with the following rules.

$$\frac{P \overset{a}{\longrightarrow} Q}{P \overset{a}{\Longrightarrow} Q} \qquad \frac{}{P \overset{true}{\Longrightarrow} P} \qquad \frac{P \overset{a}{\Longrightarrow} Q \overset{b}{\Longrightarrow} R}{P \overset{a \sqcup b}{\Longrightarrow} R}$$

Note that, since $\sqcup$ is idempotent, if the original LTS $\overset{a}{\longrightarrow}$ has finitely many transitions, then also $\overset{a}{\Longrightarrow}$ is finite. This allows us to use the algorithm in [4] to check weak bisimilarity on (the finite fragment) of concurrent constraint programming. We have implemented this procedure in a tool that is available at `http://www.lix.polytechnique.fr/~andresaristi/checkers/`. To the best of our knowledge, this is the first tool for checking weak equivalence of ccp programs.

This paper is structured as follows. In Sec. 2 we recall the partition refinement method and the standard reduction from weak to strong bisimilarity. We also recall the ccp formalism, its equivalences, and the ccp partition refinement algorithm. We then show why the standard reduction does not work for ccp. Finally, in Sec. 3 we present our reduction and show its correctness.

Related Work. Ccp is not the only formalism where weak bisimilarity cannot be naively reduced to the strong one. Probably the first case in literature can be found in [31] that introduces an algorithm for checking weak open bisimilarity of π-calculus. This algorithm is rather different from ours, since it is on-the-fly [12] and thus it checks the equivalence of only two given states (while our algorithm, and more generally all algorithms based on partition refinement, check the equivalence of all the states of a given LTS). Also [5] defines weak labelled transitions following the above-mentioned standard method which does not work in the ccp case.

Analogous problems to the one discussed in this paper arise in Petri nets [29, 11], in tile transition systems [14, 10] and, more generally, in the theory of reactive systems [15] (the interested reader is referred to [30] for an overview). In all these cases, labels form a monoid where the neutral element is the label of internal transitions. Roughly, when reducing from weak to strong bisimilarity, one needs to close the transitions with respect to the composition of the monoid (and not only with respect to the neutral element). However, in all these cases, labels composition is not idempotent (as it is for ccp) and thus a finite LTS might be transformed into an infinite one. For this reason, this procedure applied to the afore mentioned cases is not effective for automatic verification.

2 From Weak to Strong CCP Bisimilarity: Saturation Approach

The problem of whether two states are weakly bisimilar in traditional labeled transitions systems is typically reduced to the problem of whether they are strongly bisimilar which can be efficiently verified using partition refinement. We shall refer to this standard reduction as Milner's saturation method [1].

In this section we shall show that this method does not work for ccp. More precisely, Milner's reduction will produce an equivalence that does not correspond to the one expected. First, we shall recall the partition refinement algorithm for strong bisimilarity and Milner's saturation method. Then we show the corresponding notions in ccp.

Standard Partition Refinement. In this section we recall the partition refinement algorithm [16] for checking bisimilarity over the states of a *labeled transition system*. Remember that an LTS can be intuitively seen as a graph where nodes represent states and arcs represent transitions between states. A transition $P \xrightarrow{a} Q$ between P and Q labeled with a can be typically thought of as an evolution from P to Q provided that a condition a is met. Transition systems can be used to represent the evolution of processes in calculi such as CCS and the π-calculus [20, 21]. In this case states correspond to processes and transitions are given by the operational semantics of the calculus.

Let us now introduce some notation. Given a set S, a *partition* $\mathscr{P}$ of S is a set of non-empty *blocks*, i.e., subsets of S, that are all disjoint and whose union is S. We write $\{B_1\}\dots\{B_n\}$ to denote a partition consisting of blocks $B_1,\dots,B_n$. A partition represents an equivalence relation where equivalent elements belong to the same block. We write $P\mathscr{P}Q$ to mean that P and Q are equivalent in the partition $\mathscr{P}$.

The *partition refinement algorithm* (see Alg. 1) checks bisimilarity as follows. First, it computes $IS^\star$, that is the set of all states that are reachable from the set of initial state IS. Then it creates the partition $\mathscr{P}^0$ where all the elements of $IS^\star$ belong to the same block (i.e., they are all equivalent). After the initialization, it iteratively refines the partitions by employing the function $\mathbf{F}$, defined as follows: for all partitions $\mathscr{P}$, $P\,\mathbf{F}(\mathscr{P})\,Q$ iff

- if $P \xrightarrow{a} P'$ then exists Q' s.t. $Q \xrightarrow{a} Q'$ and $P'\,\mathscr{P}Q'$.

The algorithm terminates whenever two consecutive partitions are equivalent. In such a partition two states belong to the same block iff they are bisimilar.

Standard reduction from weak to strong bisimilarity. As pointed out in the literature (Chapter 3 from [25]), in order to compute weak bisimilarity, we can use the above mentioned partition refinement.

```
Algorithm 1 Partition-Refinement($IS$)
Initialization
    1. $IS^\star$ is the set of all processes reachable from $IS$,
    2. $\mathscr{P}^0 := \{IS^\star\}$,
Iteration $\mathscr{P}^{n+1} := \mathbf{F}(\mathscr{P}^n)$,
Termination If $\mathscr{P}^n = \mathscr{P}^{n+1}$ then return $\mathscr{P}^n$.
```

$$\text{MR1}\ \frac{\gamma \xrightarrow{\alpha} \gamma'}{\gamma \overset{\alpha}{\Longrightarrow} \gamma'} \qquad \text{MR2}\ \frac{}{\gamma \overset{true}{\Longrightarrow} \gamma} \qquad \text{MR3}\ \frac{\gamma \overset{true}{\Longrightarrow} \gamma_1 \overset{\alpha}{\Longrightarrow} \gamma_2 \overset{true}{\Longrightarrow} \gamma'}{\gamma \overset{\alpha}{\Longrightarrow} \gamma'}$$

Table 1: Milner's Saturation Method

The idea is to start from the graph generated via the operational semantics and then *saturate* it using the rules described in Tab. 1 to produce a new labeled transition relation $\Longrightarrow$. Recall that $\longrightarrow^*$ is the reflexive and transitive closure of the transition relation $\longrightarrow$. Now the problem whether two states are weakly bisimilar can be reduced to checking whether they are strongly bisimilar wrt $\Longrightarrow$ using partition refinement. This approach does not work in a formalism like ccp. We shall see that the problem involves the ccp transition labels which, being constraints, can be arbitrary combined using the lub operation $\sqcup$ to form a new one. Such a situation does not arise in CCS-like labelled transitions.

Notation 1. *When the label of a transition is true we will omit it. Namely, henceforth we will use* $\gamma \longrightarrow \gamma'$ *and* $\gamma \Longrightarrow \gamma'$ *to denote* $\gamma \xrightarrow{true} \gamma'$ *and* $\gamma \overset{true}{\Longrightarrow} \gamma'$.

2.1 CCP

We shall now recall ccp and the adaptation of the partition refinement algorithm to compute bisimilarity in ccp [4].

Constraint Systems. The ccp model is parametric in a *constraint system (cs)* specifying the structure and interdependencies of the information that processes can ask or and add to a *central shared store*. This information is represented as assertions traditionally referred to as *constraints*. Following [6, 19] we regard a cs as a complete algebraic lattice in which the ordering $\sqsubseteq$ is the reverse of an entailment relation: $c \sqsubseteq d$ means d *entails* c, i.e., d contains "more information" than c. The top element *false* represents inconsistency, the bottom element *true* is the empty constraint, and the *least upper bound* (lub) $\sqcup$ is the join of information.

Definition 1 (cs)**.** *A* constraint system (cs) $\mathbf{C} = (Con, Con_0, \sqsubseteq, \sqcup, true, false)$ *is a complete algebraic lattice where* Con*, the set of constraints, is a partially ordered set wrt* $\sqsubseteq$*,* Con_0 *is the subset of* compact *elements of* Con*,* $\sqcup$ *is the lub operation defined on all subsets, and true, false are the least and greatest elements of* Con*, respectively.*

Remark 1. *We shall assume that the constraint system is well-founded and, for practical reasons, that its ordering* $\sqsubseteq$ *is decidable.*

We now define the constraint system we use in our examples.

$$\mathrm{R1}\,\langle \mathbf{tell}(c),d\rangle \longrightarrow \langle \mathbf{stop}, d\sqcup c\rangle \quad \mathrm{R2}\,\frac{c\sqsubseteq d}{\langle \mathbf{ask}\ (c)\ \rightarrow\ P,d\rangle \longrightarrow \langle P,d\rangle} \quad \mathrm{R3}\,\frac{\langle P,d\rangle \longrightarrow \langle P',d'\rangle}{\langle P\parallel Q,d\rangle \longrightarrow \langle P'\parallel Q,d'\rangle} \quad \mathrm{R4}\,\frac{\langle P,d\rangle \longrightarrow \langle P',d'\rangle}{\langle P+Q,d\rangle \longrightarrow \langle P',d'\rangle}$$

Table 2: Reduction semantics for ccp (the symmetric rules for R3 and R4 are omitted).

Example 1. *Let Var be a set of variables and ω be the set of natural numbers. A variable assignment is a function $\mu : Var \longrightarrow \omega$. We use $\mathscr{A}$ to denote the set of all assignments, $\mathscr{P}(\mathscr{A})$ to denote the powerset of $\mathscr{A}$, $\emptyset$ the empty set and $\cap$ the intersection of sets. Let us define the following constraint system: The set of constraints is $\mathscr{P}(\mathscr{A})$. We define $c \sqsubseteq d$ iff $c \supseteq d$. The constraint false is $\emptyset$, while true is $\mathscr{A}$. Given two constraints c and d, $c \sqcup d$ is the intersection $c \cap d$. We will often use a formula like $x < n$ to denote the corresponding constraint, i.e., the set of all assignments that map x to a number smaller than n.*

Processes We now recall the basic ccp process constructions. For the sake of space and simplicity we dispense with the recursion operator, which is defined in the standard way as in CCS or other process algebras, and the local/hiding operator (see [3] for further details).

Syntax. Let us presuppose a constraint system $\mathbf{C} = (Con, Con_0, \sqsubseteq, \sqcup, true, false)$. The ccp processes are given by the following syntax:

$$P,Q ::= \mathbf{stop} \mid \mathbf{tell}(c) \mid \mathbf{ask}(c) \rightarrow P \mid P \parallel Q \mid P+Q$$

where $c \in Con_0$. Intuitively, **stop** represents termination, $\mathbf{tell}(c)$ adds the constraint (or partial information) c to the store. The addition is performed regardless the generation of inconsistent information. The process $\mathbf{ask}(c) \rightarrow P$ may execute P if c is entailed from the information in the store. The processes $P \parallel Q$ and $P+Q$ stand, respectively, for the *parallel execution* and *non-deterministic choice* of P and Q.

Reduction Semantics. The operational semantics is given by transitions between configurations. A configuration is a pair $\langle P,d\rangle$ representing a *state* of a system; d is a constraint representing the global store, and P is a process, i.e., a term of the syntax. We use *Conf* with typical elements $\gamma, \gamma', \ldots$ to denote the set of configurations. The operational model of ccp is given by the transition relation $\longrightarrow \subseteq Conf \times Conf$ defined in Tab. 2. The rules in Tab. 2 are easily seen to realize the above intuitions.

Barbed Semantics. The authors in [3] introduced a barbed semantics for ccp. Barbed equivalences have been introduced in [22] for CCS, and have become the standard behavioural equivalences for formalisms equipped with unlabeled reduction semantics. Intuitively, *barbs* are basic observations (predicates) on the states of a system. In the case of ccp, barbs are taken from the underlying set Con_0 of the constraint system. A configuration $\gamma = \langle P,d\rangle$ is said to *satisfy* the barb c ($\gamma \downarrow_c$) iff $c \sqsubseteq d$. Similarly, γ satisfies a *weak barb* c ($\gamma \Downarrow_c$) iff there exist γ' s.t. $\gamma \longrightarrow^* \gamma' \downarrow_c$.

In this context, the equivalence proposed is the *saturated bisimilarity* [8, 7]. Intuitively, in order for two states to be saturated bisimilar, then (i) they should expose the same barbs, (ii) whenever one of them moves then the other should reply and arrive at an equivalent state (i.e. follow the bisimulation game), (iii) they should be equivalent under all the possible contexts of the language. In the case of ccp, it is enough to require that bisimulations are *upward closed* as in condition (iii) below.

Definition 2 (Saturated Barbed Bisimilarity)**.** *A saturated barbed bisimulation is a symmetric relation $\mathscr{R}$ on configurations s.t. whenever $(\gamma_1, \gamma_2) \in \mathscr{R}$ with $\gamma_1 = \langle P,c\rangle$ and $\gamma_2 = \langle Q,d\rangle$ implies that: (i) if $\gamma_1 \downarrow_e$ then $\gamma_2 \downarrow_e$, (ii) if $\gamma_1 \longrightarrow \gamma_1'$ then there exists γ_2' s.t. $\gamma_2 \longrightarrow \gamma_2'$ and $(\gamma_1', \gamma_2') \in \mathscr{R}$, (iii) for every $a \in Con_0$, $(\langle P, c \sqcup a\rangle, \langle Q, d \sqcup a\rangle) \in \mathscr{R}$. We say that γ_1 and γ_2 are saturated barbed bisimilar ($\gamma_1 \dot{\sim}_{sb} \gamma_2$) if there exists a saturated barbed bisimulation $\mathscr{R}$ s.t. $(\gamma_1, \gamma_2) \in \mathscr{R}$.*

$$\text{LR1}\,\langle \mathbf{tell}(c), d\rangle \xrightarrow{true} \langle \mathbf{stop}, d \sqcup c\rangle \qquad \text{LR2}\,\frac{\alpha \in \min\{a \in Con_0 \,|\, c \sqsubseteq d \sqcup a\ \}}{\langle \mathbf{ask}\ (c)\ \rightarrow\ P, d\rangle \xrightarrow{\alpha} \langle P, d \sqcup \alpha\rangle} \qquad \text{LR3}\,\frac{\langle P,d\rangle \xrightarrow{\alpha} \langle P',d'\rangle}{\langle P \parallel Q, d\rangle \xrightarrow{\alpha} \langle P' \parallel Q, d'\rangle} \qquad \text{LR4}\,\frac{\langle P,d\rangle \xrightarrow{\alpha} \langle P',d'\rangle}{\langle P+Q, d\rangle \xrightarrow{\alpha} \langle P',d'\rangle}$$

Table 3: Labeled semantics for ccp (the symmetric rules for LR3 and LR4 are omitted).

$T = \mathbf{tell}(true)$ $\quad P = \mathbf{ask}\ (x<7)\ \rightarrow\ T$ $\quad Q = \mathbf{ask}\ (x<5)\ \rightarrow\ T$ $\quad R = \mathbf{ask}\ (z<5)\ \rightarrow\ (P+Q)$
$T' = \mathbf{tell}(y=1)$ $\quad S = \mathbf{ask}\ (z<7)\ \rightarrow\ P$ $\quad Q' = \mathbf{ask}\ (x<5)\ \rightarrow\ T'$ $\quad R' = \mathbf{ask}\ (z<5)\ \rightarrow\ (P+Q')$

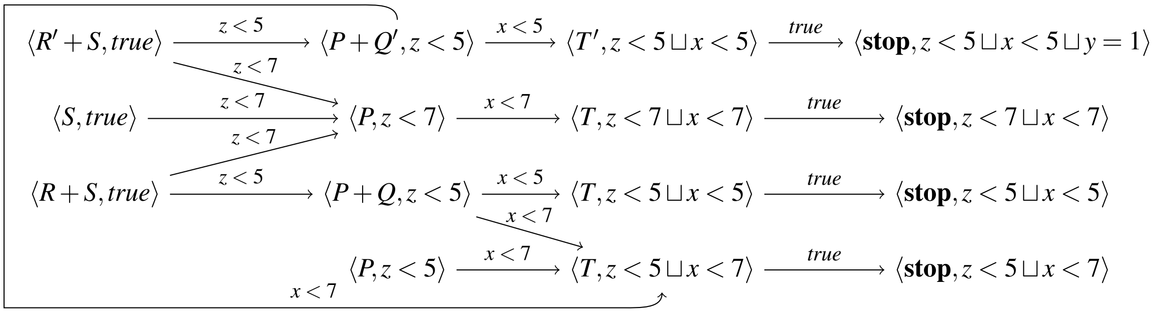

Figure 1: The LTS of the running example ($IS = \{\langle R'+S, true\rangle, \langle S, true\rangle, \langle R+S, true\rangle\}$).

We use the term "saturated" to be consistent with the original idea in [8, 7]. However, "saturated" in this context has nothing to do with the Milner's "saturation" for weak bisimilarity. In the following, we will continue to use "saturated" and "saturation" to denote these two different concepts.

Example 2. *Take $T = \mathbf{tell}(true)$, $P = \mathbf{ask}\ (x<7)\ \rightarrow\ T$ and $Q = \mathbf{ask}\ (x<5)\ \rightarrow\ T$. You can see that $\langle P, true\rangle \not\dot{\sim}_{sb} \langle Q, true\rangle$, since $\langle P, x<7\rangle \longrightarrow$, while $\langle Q, x<7\rangle \not\longrightarrow$. Consider now the configuration $\langle P+Q, true\rangle$ and observe that $\langle P+Q, true\rangle \dot{\sim}_{sb} \langle P, true\rangle$. Indeed, for all constraints e, s.t. $x<7 \sqsubseteq e$, both the configurations evolve into $\langle T, e\rangle$, while for all e s.t. $x<7 \not\sqsubseteq e$, both configurations cannot proceed. Since $x<7 \sqsubseteq x<5$, the behaviour of Q is somehow absorbed by the behaviour of P.*

As we mentioned before, we are interested in deciding the weak version of the notion above. Then, *weak saturated barbed bisimilarity* ($\dot{\approx}_{sb}$) is obtained from Def. 2 by replacing the strong barbs in condition (i) for its weak version ($\Downarrow$) and the transitions in condition (ii) for the reflexive and transitive closure of the transition relation ($\longrightarrow^*$).

Labeled Semantics. As explained in [3], in a transition of the form $\langle P, d\rangle \xrightarrow{\alpha} \langle P', d'\rangle$ the label α represents a *minimal* information (from the environment) that needs to be added to the store d to evolve from $\langle P, d\rangle$ into $\langle P', d'\rangle$, i.e., $\langle P, d \sqcup \alpha\rangle \longrightarrow \langle P', d'\rangle$. The labeled transition relation $\longrightarrow \subseteq Conf \times Con_0 \times Conf$ is defined by the rules in Tab. 3. The rule LR2, for example, says that $\langle \mathbf{ask}\ (c)\ \rightarrow\ P, d\rangle$ can evolve to $\langle P, d \sqcup \alpha\rangle$ if the environment provides a minimal constraint α that added to the store d entails c, i.e., $\alpha \in \min\{a \in Con_0 \,|\, c \sqsubseteq d \sqcup a\}$. Note that assuming that $(Con, \sqsubseteq)$ is well-founded (Remark 1) is necessary to guarantee that α exists whenever $\{a \in Con_0 \,|\, c \sqsubseteq d \sqcup a\ \}$ is not empty. The other rules are easily seen to realize the above intuition. Fig. 1 illustrates the LTSs of our running example.

The labeled semantics can be related to the reduction semantics via the following lemma, which is divided in two: soundness and completeness. The first one deals with the fact that if a process is able to perform an action using some (minimal) information from the environment, then providing such information to the process (in its store) will allow to perform a reduction instead. Now completeness goes from reductions to labeled transitions: if a configuration can perform a reduction then the idea is that one could use a piece of its store as a label, and the process should be able to arrive at the same result by means of a labeled transition. In this case, such label might not be exactly the piece of information we took from the store but something smaller, thus the resulting store could also be smaller. The following lemma is an extension of the one in [3] which considers nondeterministic ccp (Proof in the Appendix, Lemma 7 and Lemma 8).

Lemma 1 (Correctness of $\longrightarrow$). *(Soundness) If $\langle P,c\rangle \xrightarrow{\alpha} \langle P',c'\rangle$ then $\langle P,c\sqcup\alpha\rangle \longrightarrow \langle P',c'\rangle$. (Completeness) If $\langle P,c\sqcup a\rangle \longrightarrow \langle P',c'\rangle$ then there exists α and b s.t. $\langle P,c\rangle \xrightarrow{\alpha} \langle P',c''\rangle$ where $\alpha\sqcup b = a$ and $c''\sqcup b = c'$.*

The above lemma is central for deciding bisimilarity in ccp. In fact, we will show later that for the weak (saturated) semantics the completeness direction does not hold. From this we will show that the standard reduction from weak to strong does not work.

2.1.1 Equivalences: Saturated Barbed, Irredundant and Symbolic Bisimilarity

In this section we recall how to check $\dot{\sim}_{sb}$ with a modified version of partition refinement introduced in [4]. Henceforth, we shall refer to this version as *ccp partition refinement (ccp-PR)*.

The main problem with checking $\dot{\sim}_{sb}$ is the quantification over all contexts. This problem is addressed in [4] following the abstract approach in [9]. More precisely, we use an equivalent notion, namely *irredundant bisimilarity* $\dot{\sim}_I$, which can be verified with ccp-PR. As its name suggests, $\dot{\sim}_I$ only takes into account those transitions deemed irredundant.[1] However, technically speaking, going from $\dot{\sim}_{sb}$ to $\dot{\sim}_I$ requires one intermediate notion, so-called *symbolic bisimilarity*. These three notions are shown to be equivalent, i.e., $\dot{\sim}_{sb} = \dot{\sim}_{sym} = \dot{\sim}_I$. In the following we recall all of them.

Let us first give some auxiliary definitions. The first concept is that of *derivation*. Consider the following transitions (taken from Fig. 1):

$$\text{(a) } \langle P+Q, z<5\rangle \xrightarrow{x<7} \langle T, z<5\sqcup x<7\rangle \qquad \text{(b) } \langle P+Q, z<5\rangle \xrightarrow{x<5} \langle T, z<5\sqcup x<5\rangle$$

Transition (a) means that for all constraints e s.t. $x<7$ is entailed by e (formally $x<7\sqsubseteq e$), the transition (c) $\langle P+Q, z<5\sqcup e\rangle \longrightarrow \langle T, z<5\sqcup e\rangle$ can be performed, while transition (b) means that the reduction (c) is possible for all e s.t. $x<5\sqsubseteq e$. Since $x<7\sqsubseteq x<5$, transition (b) is “redundant”, in the sense that its meaning is “logically derived” by transition (a). The following notion captures the above intuition:

Definition 3 (Derivation $\vdash_D$). *We say that the transition $t = \langle P,c\rangle \xrightarrow{\alpha} \langle P',c'\rangle$ derives $t' = \langle P,c\rangle \xrightarrow{\beta} \langle P',c''\rangle$ (written $t\vdash_D t'$) iff there exists e s.t. $\alpha\sqcup e = \beta$ and $c'\sqcup e = c''$.*

One can verify in the above example that (a) $\vdash_D$ (b), and notice that both transitions arrive at the same process P', the difference lies in the label and the store. Now imagine the situation where the initial configuration is able to perform another transition with β (as in t'), let us also assume that such transition arrives at a configuration which is equivalent to the result of t'. Therefore, it is natural to think that, since t dominates t', such new transition should also be dominated by t. Let us explain with an example, consider the two following transitions:

$$\text{(e) } \langle R+S, true\rangle \xrightarrow{z<7} \langle P, z<7\rangle \qquad \text{(f) } \langle R+S, true\rangle \xrightarrow{z<5} \langle P+Q, z<5\rangle$$

Note that transition (f) cannot be derived by other transitions, since (e) $\nvdash_D$ (f). Indeed, P is syntactically different from $P+Q$, even if they have the same behaviour when inserted in the store $z<5$, i.e., $\langle P, z<5\rangle \dot{\sim}_{sb} \langle P+Q, z<5\rangle$ (since $\dot{\sim}_{sb}$ is upward closed). Transition (f) is also “redundant”, since its behaviour “does not add anything” to the behavior of (e). The following definition encompasses this situation:

[1] Redundancy itself is not trivial to check, for more information go to [4].

Definition 4 (Derivation w.r.t $\mathscr{R}$, $\vdash_{\mathscr{R}}$)**.** *We say that the transition $t = \gamma \xrightarrow{\alpha} \gamma_1$ derives $t' = \gamma \xrightarrow{\beta} \gamma_2$ w.r.t. to $\mathscr{R}$ (written $t \vdash_{\mathscr{R}} t'$) iff there exists γ_2' s.t. $t \vdash_D \gamma \xrightarrow{\beta} \gamma_2'$ and $\gamma_2' \mathscr{R} \gamma_2$.*

Then, when $\mathscr{R}$ represents some sort of equivalence, this notion will capture the situation above mentioned. Notice that $\vdash_D$ is $\vdash_{\mathscr{R}}$ with $\mathscr{R}$ being the identity relation (*id*). Now we introduce the concept of *domination*, which consists in strengthening the notion of derivation by requiring labels to be different.

Definition 5 (Domination $\succ_D$)**.** *We say that the transition $t = \langle P, c\rangle \xrightarrow{\alpha} \langle P', c'\rangle$ dominates $t' = \langle P, c\rangle \xrightarrow{\beta} \langle P', c''\rangle$ (written $t \succ_D t'$) iff $t \vdash_D t'$ and $\alpha \neq \beta$.*

Similarly, as we did for derivation, we can define domination depending on a relation. Again, $\succ_D$ is just $\succ_{\mathscr{R}}$ when $\mathscr{R}$ is the identity relation (*id*).

Definition 6 (Redundancy and Domination w.r.t $\mathscr{R}$, $\succ_{\mathscr{R}}$)**.** *We say that the transition $t = \langle P, c\rangle \xrightarrow{\alpha} \langle P', c'\rangle$ dominates $t' = \langle P, c\rangle \xrightarrow{\beta} \langle Q, d\rangle$ w.r.t. to $\mathscr{R}$ (written $t \succ_{\mathscr{R}} t'$) iff there exists c'' s.t. $t \succ_D \langle P, c\rangle \xrightarrow{\beta} \langle P', c''\rangle$ and $\langle P, c''\rangle \mathscr{R} \langle Q, d\rangle$. Also, a transition is said to be redundant when it is dominated by another, otherwise it is said to be irredundant.*

We are now able to introduce symbolic bisimilarity. Intuitively, two configurations γ_1 and γ_2 are symbolic bisimilar iff (i) they have the same barbs and (ii) whenever there is a transition from γ_1 to γ_1' using α, then we require that γ_2 must reply with a similar transition $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ (where γ_1' and γ_2' are now equivalent) or some other transition that derives it. In other words, the move from the defender does not need to use exactly the same label, but a transition that is "stronger" (in terms of derivation $\vdash_D$) could also do the job. Formally we have the definition below.

Definition 7 (Symbolic Bisimilarity)**.** *A symbolic bisimulation is a symmetric relation $\mathscr{R}$ on configurations s.t. whenever $(\gamma_1, \gamma_2) \in \mathscr{R}$ with $\gamma_1 = \langle P, c\rangle$ and $\gamma_2 = \langle Q, d\rangle$ implies that: (i) if $\gamma_1 \downarrow_e$ then $\gamma_2 \downarrow_e$, (ii) if $\langle P, c\rangle \xrightarrow{\alpha} \langle P', c'\rangle$ then there exists a transition $t = \langle Q, d\rangle \xrightarrow{\beta} \langle Q', d''\rangle$ and a store d' s.t. $t \vdash_D \langle Q, d\rangle \xrightarrow{\alpha} \langle Q', d'\rangle$ and $\langle P', c'\rangle \mathscr{R} \langle Q', d'\rangle$ We say that γ_1 and γ_2 are symbolic bisimilar ($\gamma_1 \dot{\sim}_{sym} \gamma_2$) if there exists a symbolic bisimulation $\mathscr{R}$ s.t. $(\gamma_1, \gamma_2) \in \mathscr{R}$.*

Example 3. To illustrate the notion of $\dot{\sim}_{sym}$ we take $\langle P+Q, true\rangle$ and $\langle P, true\rangle$ from Ex. 2. We provide a symbolic bisimulation $\mathscr{R} = \{(\langle P+Q, true\rangle, \langle P, true\rangle), (\langle T, x<7\rangle, \langle T, x<7\rangle), (\langle T, x<5\rangle, \langle T, x<5\rangle), (\langle \mathbf{stop}, x<7\rangle, \langle \mathbf{stop}, x<7\rangle), (\langle \mathbf{stop}, x<5\rangle, \langle \mathbf{stop}, x<5\rangle)\}$ to prove $\langle P+Q, true\rangle \dot{\sim}_{sym} \langle P, true\rangle$. We take the pair $(\langle P+Q, true\rangle, \langle P, true\rangle)$. The first condition in Def. 7 is trivial. For the second one, we take $\langle P+Q, true\rangle \xrightarrow{x<5} \langle T, x<5\rangle$ and one can find transitions $t = \langle P, true\rangle \xrightarrow{x<7} \langle T, x<7\rangle$ and $t' = \langle P, true\rangle \xrightarrow{x<5} \langle T, x<5\rangle$ s.t. $t \vdash_D t'$ and $\langle T, x<5\rangle \mathscr{R} \langle T, x<5\rangle$. The restant pairs are trivially verified.

And finally, the irredundant version, which follows the standard bisimulation game where labels need to be matched, however only those transitions so-called irredundant must be considered.

Definition 8 (Irredundant Bisimilarity)**.** *An irredundant bisimulation is a symmetric relation $\mathscr{R}$ on configurations s.t. whenever $(\gamma_1, \gamma_2) \in \mathscr{R}$ implies that: (i) if $\gamma_1 \downarrow_e$ then $\gamma_2 \downarrow_e$, (ii) if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ and it is irredundant in $\mathscr{R}$ then there exists γ_2' s.t. $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $(\gamma_1', \gamma_2') \in \mathscr{R}$. We say that γ_1 and γ_2 are irredundant bisimilar ($\gamma_1 \dot{\sim}_I \gamma_2$) if there exists an irredundant bisimulation $\mathscr{R}$ s.t. $(\gamma_1, \gamma_2) \in \mathscr{R}$.*

Example 4. We can verify that the relation $\mathscr{R}$ in Ex. 3 is an irredundant bisimulation to show that $\langle P+Q, true\rangle \dot{\sim}_I \langle P, true\rangle$. We take the pair $(\langle P+Q, true\rangle, \langle P, true\rangle)$. The first item in Def. 8 is obvious. Then take $\langle P+Q, true\rangle \xrightarrow{x<7} \langle T, x<7\rangle$, which is irredundant according to Def. 6, then there exists a $\langle T, x<7\rangle$ s.t. $\langle P, true\rangle \xrightarrow{x<7} \langle T, x<7\rangle$ and $(\langle T, x<7\rangle, \langle T, x<7\rangle) \in \mathscr{R}$. The other pairs are trivially proven. Notice that $\langle P+Q, true\rangle \xrightarrow{x<7} \langle T, x<7\rangle \succ_{\mathscr{R}} \langle P+Q, true\rangle \xrightarrow{x<5} \langle T, x<5\rangle$ hence $\langle P+Q, true\rangle \xrightarrow{x<5} \langle T, x<5\rangle$ is redundant, thus it does not need to be matched by $\langle P, true\rangle$.

As we said at the beginning, the above-defined equivalences coincide with $\dot{\sim}_{sb}$. The proof, given in [4], strongly relies on Lemma 1.

Theorem 1. *$\langle P,c\rangle \dot{\sim}_I \langle Q,d\rangle$ iff $\langle P,c\rangle \dot{\sim}_{sym} \langle Q,d\rangle$ iff $\langle P,c\rangle \dot{\sim}_{sb} \langle Q,d\rangle$*

2.1.2 Partition Refinement for CCP

In [4] the authors introduced an algorithm for checking $\dot{\sim}_{sb}$, by modifying the partition refinement algorithm so that to exploit $\dot{\sim}_I$. First, since configurations satisfying different barbs are surely different, it can be safely started with a partition that equates all and only those states satisfying the same barbs. Note that two configurations satisfy the same barbs iff they have the same store. Thus, we take as initial partition $\mathscr{P}^0 = \{IS^{\star}_{d_1}\}\dots\{IS^{\star}_{d_n}\}$, where $IS^{\star}_{d_i}$ is the subset of the configurations of $IS^{\star}$ with store d_i.[2] Secondly, instead of using the function **F** of Alg. 1, the partitions are refined by employing the function **IR** defined as follows: for all partitions $\mathscr{P}$, $\gamma_1\, \mathbf{IR}(\mathscr{P})\, \gamma_2$ iff

- if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ is irredundant in $\mathscr{P}$, then there exists γ_2' s.t. $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $\gamma_1' \mathscr{P} \gamma_2'$.

These two steps are the main idea behind the computation of $\dot{\sim}_I$ (Alg. 2).

```
Algorithm 2 CCP-Partition-Refinement ($IS$)
Initialization
  1. Compute $IS^{\star}_{new}$
  2. $\mathscr{P}^0 := \{IS^{\star}_{d_1}\}\dots\{IS^{\star}_{d_n}\}$,
Iteration $\mathscr{P}^{n+1} := \mathbf{IR}(\mathscr{P}^n)$
Termination If $\mathscr{P}^n = \mathscr{P}^{n+1}$ then return $\mathscr{P}^n$.
```

2.2 Incompleteness of Milner's saturation method in ccp

As mentioned at the beginning of this section, the standard approach for deciding weak equivalences is to add some transitions to the original processes, so-called *saturation*, and then check for the strong equivalence. In calculi like CCS, such saturation consists in forgetting about the internal actions that make part of a sequence containing one observable action (Tab. 1). However, for ccp this method does not work. The problem is that the transition relation proposed by Milner is not complete for ccp, hence the relation among the saturated, symbolic and irredundant equivalences is broken. In the next section we will provide a stronger saturation, which is complete, and allow us to use the ccp-PR to compute $\dot{\approx}_{sb}$.

Let us show why Milner's approach does not work. First, we need to introduce formally the concept of *completeness* for a given transition relation.

Definition 9. *We say that a transition relation $\rightsquigarrow \subseteq Conf \times Con_0 \times Conf$ is complete iff whenever $\langle P, c \sqcup a\rangle \rightsquigarrow \langle P', c'\rangle$ then there exist $\alpha, b \in Con_0$ s.t. $\langle P,c\rangle \overset{\alpha}{\rightsquigarrow} \langle P', c''\rangle$ where $\alpha \sqcup b = a$ and $c'' \sqcup b = c'$.*

Notice that $\longrightarrow$ is complete, and it corresponds to the second item of Lemma 1. Now Milner's method defines a new transition relation $\Longrightarrow$ using the rules in Tab. 1, but it turns out not to be complete.

Proposition 1. *Milner's saturation method ($\Longrightarrow$) is not complete.*

[2] In fact, in order to check redundancy, some new states should be added the initial ones $IS^{\star}_{new}$. The details of the computation are omitted given that they are not relevant for this paper, however the interested reader is referred to [4] for more information.

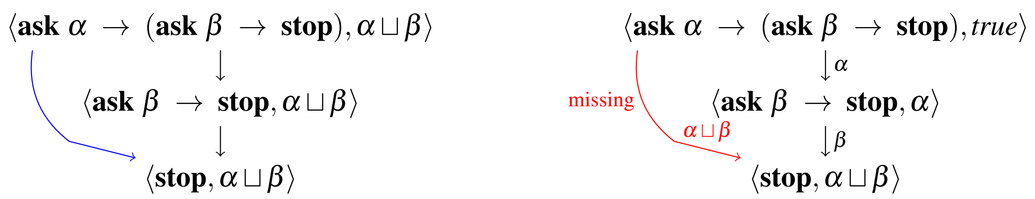
$\langle \mathbf{ask}\ \alpha \rightarrow (\mathbf{ask}\ \beta \rightarrow \mathbf{stop}), \alpha \sqcup \beta \rangle$
$\langle \mathbf{ask}\ \beta \rightarrow \mathbf{stop}, \alpha \sqcup \beta \rangle$
$\langle \mathbf{stop}, \alpha \sqcup \beta \rangle$
$\langle \mathbf{ask}\ \alpha \rightarrow (\mathbf{ask}\ \beta \rightarrow \mathbf{stop}), true \rangle$
α
missing
$\langle \mathbf{ask}\ \beta \rightarrow \mathbf{stop}, \alpha \rangle$
$\alpha \sqcup \beta$
β
$\langle \mathbf{stop}, \alpha \sqcup \beta \rangle$

Figure 2: Counterexample for completeness using Milner's saturation method (cycles from MR2 omitted). Both graphs are obtained by applying the rules in Tab. 1

Proof. We will show a counter-example where the completeness for $\Longrightarrow$ does not hold. Let $P = \mathbf{ask}\ \alpha \rightarrow (\mathbf{ask}\ \beta \rightarrow \mathbf{stop})$ and $d = \alpha \sqcup \beta$. Now consider the transition $\langle P, d \rangle \Longrightarrow \langle \mathbf{stop}, d \rangle$ and let us apply the completeness lemma, we can take $c = true$ and $a = \alpha \sqcup \beta$, therefore by completeness there must exist b and λ s.t. $\langle P, true \rangle \overset{\lambda}{\Longrightarrow} \langle \mathbf{stop}, c'' \rangle$ where $\lambda \sqcup b = \alpha \sqcup \beta$ and $c'' \sqcup b = d$. However, notice that the only transition possible is $\langle P, true \rangle \overset{\alpha}{\Longrightarrow} \langle \mathbf{ask}\ \beta \rightarrow \mathbf{stop}, \alpha \rangle$, hence completeness does not hold since there is no transition from $\langle P, true \rangle$ to $\langle \mathbf{stop}, c'' \rangle$ for some c''. Fig. 2 illustrates the problem. □

We can now use this fact to see why the method does not work for computing $\dot{\approx}_{sb}$ using ccp-PR. First, let us redefine some concepts using the new transition relation $\Longrightarrow$. Because of condition (i) in $\dot{\approx}_{sb}$, we need a new definition of barbs, namely *weak barbs w.r.t.* $\Longrightarrow$. We say γ has a weak barb e w.r.t. $\Longrightarrow$, denoted by $\gamma \lightning_e$, iff $\gamma \Longrightarrow^* \gamma' \downarrow_e$.

Using this notion, we introduce Symbolic and Irredundant bisimilarity w.r.t. $\Longrightarrow$, denoted by $\dot{\sim}^{\Longrightarrow}_{sym}$ and $\dot{\sim}^{\Longrightarrow}_{I}$ respectively. They are defined as in Def. 7 and 8 where in condition (i) weak barbs ($\Downarrow$) are replaced with $\lightning$ and in condition (ii) the transition relation is now $\Longrightarrow$.

One would expect that since $\dot{\sim}_{sb} = \dot{\sim}_{sym} = \dot{\sim}_{I}$ then the natural consequence will be that $\dot{\approx}_{sb} = \dot{\sim}^{\Longrightarrow}_{sym} = \dot{\sim}^{\Longrightarrow}_{I}$, given that these new notions are supposed to be the weak versions of the former ones when using the saturation method. However, completeness is necessary for proving $\dot{\sim}_{sb} = \dot{\sim}_{sym} = \dot{\sim}_{I}$, and from Proposition 1 we know that $\Longrightarrow$ is not complete hence we might expect $\dot{\approx}_{sb} \neq \dot{\sim}^{\Longrightarrow}_{sym} \neq \dot{\sim}^{\Longrightarrow}_{I}$. In fact, the following counter-example shows these inequalities.

Example 5. *Let P, P' and Q as in Fig. 4. The figure shows $\langle P, true \rangle$ and $\langle Q, true \rangle$ after we saturate them using Milner's method. First, notice that $\langle P, true \rangle \dot{\approx}_{sb} \langle Q, true \rangle$, since there exists a saturated weak barbed bisimulation $\mathscr{R} = \{(\langle P, true \rangle, \langle Q, true \rangle), (\langle P', \alpha \rangle, \langle P', \alpha \rangle), (\langle \mathbf{tell}(c), \alpha \sqcup \beta \rangle, \langle \mathbf{tell}(c), \alpha \sqcup \beta \rangle), (\langle \mathbf{stop}, \alpha \sqcup \beta \sqcup c \rangle, \langle \mathbf{stop}, \alpha \sqcup \beta \sqcup c \rangle), (\langle \mathbf{tell}(d), \alpha \rangle, \langle \mathbf{tell}(d), \alpha \rangle), (\langle \mathbf{stop}, \alpha \sqcup d \rangle, \langle \mathbf{stop}, \alpha \sqcup d \rangle), (\langle \mathbf{tell}(c), \alpha \sqcup \beta \rangle, \langle \mathbf{tell}(c), \alpha \sqcup \beta \rangle), (\langle \mathbf{stop}, \alpha \sqcup \beta \sqcup c \rangle, \langle \mathbf{stop}, \alpha \sqcup \beta \sqcup c \rangle)\}$. However, $\langle P, true \rangle \not\dot{\sim}^{\Longrightarrow}_{I} \langle Q, true \rangle$. To prove that, we need to pick an irredundant transition from $\langle P, true \rangle$ or $\langle Q, true \rangle$ (after saturation) s.t. the other cannot match. Thus, take $\langle Q, true \rangle \xrightarrow{\alpha \sqcup \beta} \langle \mathbf{tell}(c), \alpha \sqcup \beta \rangle$ which is irredundant and given that $\langle P, true \rangle$ does not have a transition with $\alpha \sqcup \beta$ then we know that there is no irredundant bisimulation containing $(\langle P, true \rangle, \langle Q, true \rangle)$ therefore $\langle P, true \rangle \not\dot{\sim}^{\Longrightarrow}_{I} \langle Q, true \rangle$. Using the same reasoning we can also show that $\dot{\approx}_{sb} \neq \dot{\sim}^{\Longrightarrow}_{sym}$.*

3 Reducing weak bisimilarity to Strong in CCP

In this section we shall provide a method for deciding weak bisimilarity in ccp. As shown in Sec. 2.2, the usual method for deciding weak bisimilarity (introduced in Sec. 2) does not work for ccp. We shall proceed by redefining $\Longrightarrow$ in such a way that it is sound and complete for ccp. Then we prove that, w.r.t.

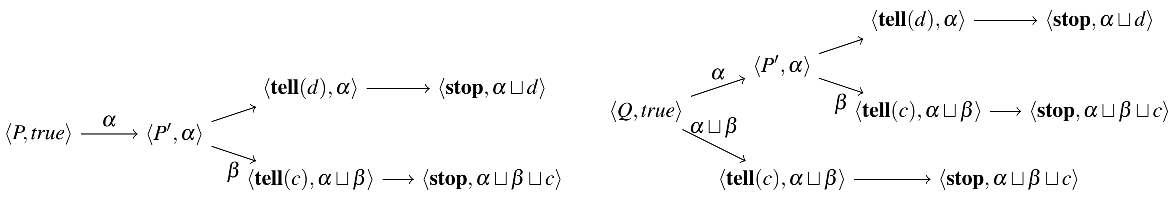

Figure 3: Execution of $\langle P, true\rangle$ and $\langle Q, true\rangle$

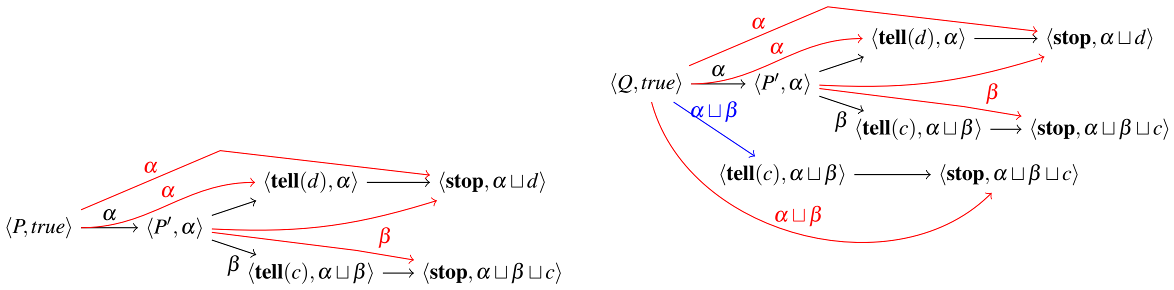

Figure 4: Let $P = \textbf{ask}\ (\alpha) \rightarrow P'$, $P' = (\textbf{ask}\ (\beta) \rightarrow \textbf{tell}(c)) + (\textbf{ask}\ (true) \rightarrow \textbf{tell}(d))$ and $Q = P + (\textbf{ask}\ (\alpha \sqcup \beta) \rightarrow \textbf{tell}(c))$. The figure represents $\langle P, true\rangle$ and $\langle Q, true\rangle$ after being saturated using Milner's method (cycles from MR2 ommited). The red transitions are the new ones added by the rules in Tab. 1. The blue transition is the (irredundant) one that $\langle Q, true\rangle$ can take but $\langle P, true\rangle$ cannot match, therefore showing that $\langle P, true\rangle \dot{\not\sim}_I^{\Longrightarrow} \langle Q, true\rangle$

$\Longrightarrow$, symbolic and irredundant bisimilarity coincide with $\dot{\approx}_{sb}$, i.e. $\dot{\approx}_{sb} = \dot{\sim}_{sym}^{\Longrightarrow} = \dot{\sim}_I^{\Longrightarrow}$. We therefore conclude that the partition refinement algorithm in [4] can be used to verify $\dot{\approx}_{sb}$ w.r.t. $\Longrightarrow$.

3.1 Defining a new saturation method for CCP

If we analyze the counter-example to completeness (see Fig. 2), one can see that the problem arises because of the nature of the labels in ccp, namely using this method $\langle \textbf{ask}\ \alpha \rightarrow (\textbf{ask}\ \beta \rightarrow \textbf{stop}), true\rangle$ does not have a transition with $\alpha \sqcup \beta$ to $\langle \textbf{stop}, \alpha \sqcup \beta\rangle$, hence that fact can be exploited to break the relation among the weak equivalences. Following this reasoning, instead of only forgetting about the silent actions we also take into account that labels in ccp can be added together. Thus we have a new rule that creates a new transition for each two consecutive ones, whose label is the lub of the labels in them. This method can also be thought as the reflexive and transitive closure of the labeled transition relation ($\xrightarrow{\alpha}$). This transition relation turns out to be sound and complete and it can be used to decide $\dot{\approx}_{sb}$.

3.1.1 A new saturation method

Formally, our new transition relation $\Longrightarrow$ is defined by the rules in Tab. 4. For simplicity, we are using the same arrow $\Longrightarrow$ to denote this transition relation. Consequently the definitions of weak barbs, symbolic and irredundant bisimilarity are now interpreted w.r.t. $\Longrightarrow$ ($\lightning$, $\dot{\sim}_{sym}^{\Longrightarrow}$ and $\dot{\sim}_I^{\Longrightarrow}$ respectively).

R-Tau	$\dfrac{}{\gamma \Longrightarrow \gamma}$	R-Label	$\dfrac{\gamma \xrightarrow{\alpha} \gamma'}{\gamma \overset{\alpha}{\Longrightarrow} \gamma'}$	R-Add	$\dfrac{\gamma \overset{\alpha}{\Longrightarrow} \gamma' \overset{\beta}{\Longrightarrow} \gamma''}{\gamma \overset{\alpha \sqcup \beta}{\Longrightarrow} \gamma''}$

Table 4: New Labelled Transition System.

We now prove that $\lightning$ coincides with $\Downarrow$, given that a transition under the new relation corresponds to a sequence of reductions.

Lemma 2. *$\gamma \longrightarrow^* \gamma'$ iff $\gamma \Longrightarrow \gamma'$.*

Using this lemma, it is straightforward to see that the notions of weak barbs coincide.

Proposition 2. *$\gamma \Downarrow_e$ iff $\gamma \lightning_e$.*

An important property is that the new labeled transition system is finitely branching, and this is due to the fact that labels in ccp are idempotent.

Proposition 3. *For any γ, we have $|\{\gamma' \mid \gamma \Longrightarrow \gamma'\}| < \infty$.*

3.1.2 Soundness and Completeness

As mentioned before, soundness and completeness of the relation are the core properties when proving $\dot{\sim}_{sb} = \dot{\sim}_{sym} = \dot{\sim}_I$. We now proceed to show that our method enjoys of these properties and they will allow us to prove the correspondence among the equivalences for the weak case.

Lemma 3 (Soundness of $\Longrightarrow$)**.** *If $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c'\rangle$ then $\langle P, c \sqcup \alpha\rangle \Longrightarrow \langle P', c'\rangle$.*

Proof. We proceed by induction on the depth of the inference of $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c'\rangle$.

- Using R-Tau we have $\langle P, c\rangle \Longrightarrow \langle P, c\rangle$ and the result follows directly given that $\alpha = true$.
- Using R-Label we have $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c'\rangle$ then $\langle P, c\rangle \xrightarrow{\alpha} \langle P', c'\rangle$. By Lemma 1 (soundness of $\longrightarrow$) we get $\langle P, c \sqcup \alpha\rangle \longrightarrow \langle P', c'\rangle$ and finally by rule R-Label $\langle P, c \sqcup \alpha\rangle \Longrightarrow \langle P', c'\rangle$.
- Using R-Add then we have $\langle P, c\rangle \overset{\beta \sqcup \lambda}{\Longrightarrow} \langle P', c'\rangle$ then $\langle P, c\rangle \overset{\beta}{\Longrightarrow} \langle P'', c''\rangle \overset{\lambda}{\Longrightarrow} \langle P', c'\rangle$ where $\beta \sqcup \lambda = \alpha$. By induction hypothesis, $\langle P, c \sqcup \beta\rangle \Longrightarrow \langle P'', c''\rangle$ (1) and $\langle P'', c'' \sqcup \lambda\rangle \Longrightarrow \langle P', c'\rangle$ (2). By monotonicity on (1), $\langle P, c \sqcup \beta \sqcup \lambda\rangle \Longrightarrow \langle P'', c'' \sqcup \lambda\rangle$ and by rule R-Add on this transition and (2) then, given that $\beta \sqcup \lambda = \alpha$, we obtain $\langle P, c \sqcup \alpha\rangle \Longrightarrow \langle P', c'\rangle$.

□

Lemma 4 (Completeness of $\Longrightarrow$)**.** *If $\langle P, c \sqcup a\rangle \Longrightarrow \langle P', c'\rangle$ then there exist α and b s.t. $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c''\rangle$ where $\alpha \sqcup b = a$ and $c'' \sqcup b = c'$.*

Proof. Assuming that $\langle P, c \sqcup a\rangle \Longrightarrow \langle P', c'\rangle$ then, from Lemma 2, we can say that $\langle P, c \sqcup a\rangle \longrightarrow^* \langle P', c'\rangle$ which can be written as $\langle P, c \sqcup a\rangle \longrightarrow \langle P_1, c_1\rangle \longrightarrow \ldots \longrightarrow \langle P_i, c_i\rangle \longrightarrow \langle P', c'\rangle$, we will proceed by induction on i.

(Base Case) Assuming $i = 0$ then $\langle P, c \sqcup a\rangle \longrightarrow \langle P', c'\rangle$ and the result follows directly from Lemma 1 (Completeness of $\longrightarrow$) and R-Label .

(Induction) Let us assume that $\langle P, c \sqcup a\rangle \longrightarrow^{i} \langle P_i, c_i\rangle \longrightarrow \langle P', c'\rangle$ then by induction hypothesis there exist β and b' s.t. $\langle P, c\rangle \overset{\beta}{\Longrightarrow} \langle P_i, c_i'\rangle$ (1) where $\beta \sqcup b' = a$ and $c_i' \sqcup b' = c_i$. Now by completeness on the last transition $\langle P_i, \overbrace{c_i}^{c_i' \sqcup b'}\rangle \longrightarrow \langle P', c'\rangle$, there exists λ and b'' s.t. $\langle P_i, c_i'\rangle \xrightarrow{\lambda} \langle P', c''\rangle$ where $\lambda \sqcup b'' = b'$ and $c'' \sqcup b'' = c'$, thus by rule R-Label we have $\langle P_i, c_i'\rangle \overset{\lambda}{\Longrightarrow} \langle P', c''\rangle$ (2). We can now proceed to apply rule R-Add on (1) and (2) to obtain the transition $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c''\rangle$ where $\alpha = \beta \sqcup \lambda$ and finally take $b = b''$, therefore the conditions hold $\alpha \sqcup b = \beta \sqcup \lambda \sqcup b'' = a$ and $c'' \sqcup b = c'' \sqcup b'' = c'$. □

3.2 Weak saturated bisimilarity coincides with the strong symbolic and irredundant bisimilarity

We show our main result, a method for deciding $\dot{\approx}_{sb}$. Recall that $\dot{\approx}_{sb}$ is the standard weak bisimilarity for ccp [3], and it is defined in terms of $\longrightarrow$, therefore it does not depend on $\Longrightarrow$. Roughly, we start from the fact that ccp-PR is able to check whether two configurations are irredundant bisimilar $\dot{\sim}_I$. Such configurations evolve according to a transition relation ($\longrightarrow$), then we provide a new way for them to evolve ($\Longrightarrow$) and we use the same algorithm to compute now $\dot{\sim}^{\Longrightarrow}_{I}$. Here we prove that $\dot{\approx}_{sb} = \dot{\sim}^{\Longrightarrow}_{sym} = \dot{\sim}^{\Longrightarrow}_{I}$ hence we give a reduction from $\dot{\approx}_{sb}$ to $\dot{\sim}^{\Longrightarrow}_{I}$ which has an effective decision procedure.

Given the transition relation is sound and complete, the correspondence between the symbolic and irredundant bisimilarity follows from [4].

Corollary 1. *$\gamma \dot{\sim}^{\Longrightarrow}_{sym} \gamma'$ iff $\gamma \dot{\sim}^{\Longrightarrow}_{I} \gamma'$*

Finally, in the next two lemmata, we prove that $\dot{\approx}_{sb} = \dot{\sim}^{\Longrightarrow}_{sym}$.

Lemma 5. *If $\gamma \dot{\approx}_{sb} \gamma'$ then $\gamma \dot{\sim}^{\Longrightarrow}_{sym} \gamma'$*

Proof. We need to prove that $\mathscr{R} = \{(\langle P, c\rangle, \langle Q, d\rangle) \mid \langle P, c\rangle \dot{\approx}_{sb} \langle Q, d\rangle\}$ is a symbolic bisimulation over $\Longrightarrow$. The first condition (i) of the bisimulation follows directly from Proposition 2. As for (ii), let us assume that $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c'\rangle$ then by soundness of $\Longrightarrow$ we have $\langle P, c \sqcup \alpha\rangle \Longrightarrow \langle P', c'\rangle$, now by Lemma 2 we obtain $\langle P, c \sqcup \alpha\rangle \longrightarrow^{*} \langle P', c'\rangle$. Given that $\langle P, c\rangle \dot{\approx}_{sb} \langle Q, d\rangle$ then from the latter transition we can conclude that $\langle Q, d \sqcup \alpha\rangle \longrightarrow^{*} \langle Q', d'\rangle$ where $\langle P', c'\rangle \dot{\approx}_{sb} \langle Q', d'\rangle$, hence we can use Lemma 2 again to deduce that $\langle Q, d \sqcup \alpha\rangle \Longrightarrow \langle Q', d'\rangle$. Finally, by completeness of $\Longrightarrow$, there exist β and b s.t. $t = \langle Q, d\rangle \overset{\beta}{\Longrightarrow} \langle Q', d''\rangle$ where $\beta \sqcup b = \alpha$ and $d'' \sqcup b = d'$, therefore $t \vdash_D \langle Q, d\rangle \overset{\alpha}{\Longrightarrow} \langle Q', d'\rangle$ and $\langle P', c'\rangle \mathscr{R} \langle Q', d'\rangle$. □

Lemma 6. *If $\gamma \dot{\sim}^{\Longrightarrow}_{sym} \gamma'$ then $\gamma \dot{\approx}_{sb} \gamma'$*

Proof. We need to prove that $\mathscr{R} = \{(\langle P, c \sqcup a\rangle, \langle Q, d \sqcup a\rangle) \mid \langle P, c\rangle \dot{\sim}^{\Longrightarrow}_{sym} \langle Q, d\rangle\}$ is a weak saturated bisimulation. First, condition (i) follows form Proposition 2 and (iii) by definition of $\mathscr{R}$. Let us prove condition (ii), assume $\langle P, c \sqcup a\rangle \longrightarrow^{*} \langle P', c'\rangle$ then by Lemma 2 $\langle P, c \sqcup a\rangle \Longrightarrow \langle P', c'\rangle$. Now by completeness of $\Longrightarrow$ there exist α and b s.t. $\langle P, c\rangle \overset{\alpha}{\Longrightarrow} \langle P', c''\rangle$ where $\alpha \sqcup b = a$ and $c'' \sqcup b = c'$. Since $\langle P, c\rangle \dot{\sim}^{\Longrightarrow}_{sym} \langle Q, d\rangle$ then we know there exists a transition $t = \langle Q, d\rangle \overset{\beta}{\Longrightarrow} \langle Q', d'\rangle$ s.t. $t \vdash_D \langle Q, d\rangle \overset{\alpha}{\Longrightarrow} \langle Q', d''\rangle$ and **(a)**$\langle P', c''\rangle \dot{\sim}^{\Longrightarrow}_{sym} \langle Q', d''\rangle$, by definition of $\vdash_D$ there exists b' s.t. $\beta \sqcup b' = \alpha$ and $d' \sqcup b' = d''$. Using soundness of $\Longrightarrow$ on t we get $\langle Q, d \sqcup \beta\rangle \Longrightarrow \langle Q', d'\rangle$, thus by Lemma 2 $\langle Q, d \sqcup \beta\rangle \longrightarrow^{*} \langle Q', d'\rangle$ and finally by monotonicity

$$\langle Q, d \sqcup \underbrace{\overbrace{\beta \sqcup b'}_{\alpha} \sqcup b}^{a}\rangle \longrightarrow^{*} \langle Q', \overbrace{d' \sqcup b'}^{d''} \sqcup b\rangle \qquad (1)$$

Notice that the initial transition $\langle P, c \sqcup a\rangle \longrightarrow^* \langle P', c'\rangle$ can be rewritten as $\langle P, c \sqcup a\rangle \longrightarrow^* \langle P', c'' \sqcup b\rangle$, therefore using the transition in (1), $\langle Q, d \sqcup a\rangle \longrightarrow^* \langle Q', d'' \sqcup b\rangle$, it is left to prove that $\langle P', c'' \sqcup b\rangle \mathscr{R} \langle Q', d'' \sqcup b\rangle$ which follows from (**a**). □

Using Lemma 5 and Lemma 6 we obtain the following theorem.

Theorem 2. $\langle P, c\rangle \dot{\sim}_{sym}^{\Longrightarrow} \langle Q, d\rangle$ *iff* $\langle P, c\rangle \dot{\approx}_{sb} \langle Q, d\rangle$

From the above results, we conclude that $\dot{\approx}_{sb} = \dot{\sim}_{I}^{\Longrightarrow}$. Therefore, given that using ccp-PR in combination with $\Longrightarrow$ (and $\Downarrow$) we can decide $\dot{\sim}_{I}^{\Longrightarrow}$, then we can use the same procedure to check whether two configurations are in $\dot{\approx}_{sb}$.

4 Concluding Remarks

We showed that the transition relation given by Milner's saturation method is not complete for ccp (in the sense of Definition 9). As consequence we also showed that weak saturated barbed bisimilarity $\dot{\approx}_{sb}$ [3] cannot be computed using the ccp partition refinement algorithm for (strong) bisimilarity ccp wrt to this transition relation. We then presented a new transition relation using another saturation mechanism and showed that it is complete for ccp. We also showed that the ccp partition refinement can be used to compute $\dot{\approx}_{sb}$ using the new transition relation. To the best of our knowledge, this is the first approach to verifying weak bisimilarity for ccp. As future work, we plan to investigate other calculi where the nature of their transitions systems give rise to similar situations regarding weak and strong bisimilarity, in particular Timed Concurrent Constraint Programming (tcc) [26], Non-deterministic Timed Concurrent Constraint (ntcc) [24], Universal Temporal Concurrent Constraint Programming (utcc) [23] and Epistemic Concurrent Constraint Programming (eccp) [17].

References

[1] L. Aceto, A. Ingolfsdottir & J. Srba (2011): *Advanced Topics in Bisimulation and Coinduction*, chapter The Algorithmics of Bisimilarity, pp. 100–172. Cambridge University Press.

[2] A. Aristizabal, F. Bonchi, C. Palamidessi, L. Pino & F. Valencia (2011): *Deriving Labels and Bisimilarity for Concurrent Constraint Programming (Extended Version)*. Technical Report, INRIA-CNRS.

[3] Andres Aristizabal, Filippo Bonchi, Catuscia Palamidessi, Luis Pino & Frank D. Valencia (2011): *Deriving Labels and Bisimilarity for Concurrent Constraint Programming*. In: *FOSSACS*, LNCS, Springer, pp. 138–152.

[4] Andres Aristizabal, Filippo Bonchi, Luis Pino & Frank D. Valencia (2012): *Partition Refinement for Bisimilarity in CCP*. In: *SAC*, ACM, pp. 88–93.

[5] Paolo Baldan, Andrea Bracciali & Roberto Bruni (2007): *A semantic framework for open processes*. Theor. Comput. Sci. 389(3), pp. 446–483.

[6] Frank S. de Boer, Alessandra Di Pierro & Catuscia Palamidessi (1995): *Nondeterminism and Infinite Computations in Constraint Programming*. Theor. Comput. Sci. 151(1), pp. 37–78.

[7] Filippo Bonchi, Fabio Gadducci & Giacoma Valentina Monreale (2009): *Reactive Systems, Barbed Semantics, and the Mobile Ambients*. In: *FOSSACS*, LNCS, Springer, pp. 272–287.

[8] Filippo Bonchi, Barbara König & Ugo Montanari (2006): *Saturated Semantics for Reactive Systems*. In: *LICS*, IEEE, pp. 69–80.

[9] Filippo Bonchi & Ugo Montanari (2009): *Minimization Algorithm for Symbolic Bisimilarity*. In: *ESOP*, LNCS, Springer, pp. 267–284.

[10] Roberto Bruni, Fabio Gadducci, Ugo Montanari & Pawel Sobocinski (2005): *Deriving Weak Bisimulation Congruences from Reduction Systems*. In: *CONCUR*, LNCS, Springer, pp. 293–307.

[11] Roberto Bruni, Hernán C. Melgratti & Ugo Montanari (2011): *A Connector Algebra for P/T Nets Interactions*. In: *CONCUR*, LNCS, Springer, pp. 312–326.

[12] Jean-Claude Fernandez (1989): *An Implementation of an Efficient Algorithm for Bisimulation Equivalence*. *Sci. Comput. Program.* 13(1), pp. 219–236.

[13] G.L. Ferrari, S. Gnesi, U. Montanari, M. Pistore & G. Ristori (1998): *Verifying Mobile Processes in the HAL Environment*. In: *CAV*, Springer, pp. 511–515.

[14] Fabio Gadducci & Ugo Montanari (2000): *The tile model*. In: *Proof, Language, and Interaction*, The MIT Press, pp. 133–166.

[15] O. H. Jensen. (2006): *Mobile Processes in Bigraphs*. Ph.D. thesis, University of Cambridge.

[16] Paris C. Kanellakis & Scott A. Smolka (1983): *CCS Expressions, Finite State Processes, and Three Problems of Equivalence*. In: *PODC*, ACM, pp. 228–240.

[17] Sophia Knight, Catuscia Palamidessi, Prakash Panangaden & Frank D. Valencia (2012): *Spatial Information Distribution in Constraint-based Process Calculi (Extended Version)*. Technical Report, INRIA.

[18] Ivan Lanese, Jorge A. Pérez, Davide Sangiorgi & Alan Schmitt (2011): *On the expressiveness and decidability of higher-order process calculi*. *Inf. Comput.* 209(2), pp. 198–226.

[19] N. P. Mendler, Prakash Panangaden, Philip J. Scott & R. A. G. Seely (1995): *A Logical View of Concurrent Constraint Programming*. *Nord. J. Comput.* 2(2), pp. 181–220.

[20] Robin Milner (1980): *A Calculus of Communicating Systems*. *Lecture Notes in Computer Science* 92, Springer-Verlag New York, Inc.

[21] Robin Milner (1999): *Communicating and mobile systems: the π-calculus*. Cambridge University Press.

[22] Robin Milner & Davide Sangiorgi (1992): *Barbed Bisimulation*. In: *ICALP*, LNCS, Springer, pp. 685–695.

[23] Carlos Olarte & Frank D. Valencia (2008): *Universal concurrent constraint programing: symbolic semantics and applications to security*. In: *SAC*, ACM, pp. 145–150.

[24] Catuscia Palamidessi & Frank D. Valencia (2001): *A Temporal Concurrent Constraint Programming Calculus*. In: *CP*, LNCS, Springer, pp. 302–316.

[25] Davide Sangiorgi & Jan Rutten (2012): *Advanced Topics in Bisimulation and Coinduction*. Cambridge University Press.

[26] Vijay A. Saraswat, Radha Jagadeesan & Vineet Gupta (1994): *Foundations of Timed Concurrent Constraint Programming*. In: *LICS*, IEEE, pp. 71–80.

[27] Vijay A. Saraswat & Martin C. Rinard (1990): *Concurrent Constraint Programming*. In: *POPL*, ACM Press, pp. 232–245.

[28] Vijay A. Saraswat, Martin C. Rinard & Prakash Panangaden (1991): *Semantic Foundations of Concurrent Constraint Programming*. In: *POPL*, ACM Press, pp. 333–352.

[29] Pawel Sobocinski (2010): *Representations of Petri Net Interactions*. In: *CONCUR*, LNCS, Springer, pp. 554–568.

[30] Pawel Sobocinski (2012): *Relational presheaves as labelled transition systems*. In: *In proceedings CMCS*, To appear in LNCS.

[31] Björn Victor & Faron Moller (1994): *The Mobility Workbench - A Tool for the pi-Calculus*. In: *CAV*, LNCS, Springer, pp. 428–440.

Appendix

A Correspondence between Strong and Saturated Barbed Bisimilarity

Since we are working with a non-deterministic fragment of ccp, we shall add one case (non-deterministic choice) to the proofs of soundness an completeness of the labeled semantics introduced in [3] which are consequently used to prove that strong bisimilarity and saturated barbed bisimilarity coincide. This is necessary since the algorithm calculates the notion of strong equivalence for ccp via the saturated barbed bisimilarity.

Lemma 7. *(Soundness). If $\langle P,d\rangle \xrightarrow{\alpha} \langle P',d'\rangle$ then $\langle P,d\sqcup\alpha\rangle \longrightarrow \langle P',d'\rangle$.*

Proof. By induction on (the depth) of the inference of $\langle P,d\rangle \xrightarrow{\alpha} \langle P',d'\rangle$. Here we consider just the additional case for the no-deterministic choice and refer the reader to [2] for further cases.

- Using LR4 then $P = Q+R$ and $P' = Q'$ which lead us to $\langle Q,d\rangle \xrightarrow{\alpha} \langle Q',d'\rangle$ by a shorter inference. By appeal to induction then $\langle Q,d\sqcup a\rangle \longrightarrow \langle Q',d'\rangle$. Applying Rule R4 to the Previous reduction we get $\langle Q+R,d\rangle \longrightarrow \langle Q',d'\rangle$.

□

Lemma 8. *(Completeness). If $\langle P,d\sqcup a\rangle \longrightarrow \langle P',d'\rangle$ then $\exists\alpha,b$ s.t. $\langle P,d\rangle \xrightarrow{\alpha} \langle P',d''\rangle$ and $\alpha\sqcup b = a$, $d''\sqcup b = d'$.*

Proof. The proof proceeds by induction on (the depth) of the inference of $\langle P,d\sqcup a\rangle \longrightarrow \langle P',d'\rangle$. As in Lemma 7 we consider just the case for the non-deterministic choice.

- Using Rule R4 then $P = Q+R$ and $P' = Q'$ which lead us to $\langle Q,d\sqcup a\rangle \longrightarrow \langle Q',d'\rangle$ by a shorter inference. Note that the active process generating the transition could be either an ask or a tell. Let suppose that the constraint that has been either asked or told is c. If it is generated by an ask then $d' = d\sqcup a$ and $c \sqsubseteq d\sqcup a$. Note that $a \in \{a' \in Con_0 | c \sqsubseteq d\sqcup a'\}$, and then by Remark 1 there exists $\alpha \in min(\{a' \in Con_0 | c \sqsubseteq d\sqcup a'\})$ such that $\alpha \sqsubseteq a$. If it is generated by a tell then $d' = d\sqcup a\sqcup c$, thus in both cases is safe to assume that $d' = d\sqcup a\sqcup c$. Thereafter by induction hypothesis we have that there exist α and b such that:

  $$\langle Q,d\rangle \xrightarrow{\alpha} \langle Q',d''\rangle$$

  with $a = \alpha\sqcup b$ and $d' = d''\sqcup b$. As said formerly the active process generating this transition could be either an ask or a tell. If it is generated by an ask then $d'' = d\sqcup\alpha$ and if it is generated by a tell then $\alpha = true$ and $d'' = d\sqcup c$. Hence in both cases is safe to assume that $d'' = d\sqcup c$. Now by Rule LR4 we have that

  $$\langle Q+R,d\rangle \xrightarrow{\alpha} \langle Q',d''\rangle$$

  since $d'' = d\sqcup c$ we say that $a = b$ and that $d' = d\sqcup a\sqcup c = d\sqcup b\sqcup c = d''\sqcup b$.

□

Lemma 9. *If $\langle P,d\rangle \dot{\sim} \langle Q,e\rangle$, then $\forall a \in Con_0$, $\langle P,d\sqcup a\rangle \dot{\sim} \langle Q,e\sqcup a\rangle$.*

Proof. Refer to [2] taking into account the additional case for Lemmas of Soundness and Completness of the labelled system. □

We can proceed to prove that $\dot{\sim}_{sb} = \dot{\sim}$. We split the two directions of the proof in two lemmas.

Lemma 10. $\dot{\sim} \subseteq \dot{\sim}_{sb}$

Proof. Let $\mathscr{R} = \{(\langle P,d\rangle, \langle Q,e\rangle)$ s.t $\langle P,d\rangle \dot{\sim} \langle Q,e\rangle\}$. Refer to [2] taking into account the additional cases for Lemmas of Soundness and Completness of the labelled system. □

Lemma 11. $\dot{\sim}_{sb} \subseteq \dot{\sim}$

Proof. Refer to [2] taking into account the additional case for Lemmas of Soundness and Completness of the labelled system. □

Consequently, we have shown that strong bisimilarity coincides with the strong saturated barbed bisimilarity

Theorem 3. $\dot{\sim}_{sb} = \dot{\sim}$.

Proof. Using Lemma 10 and Lemma 11 □