# To be defined

Laura Bussi[b], Fabio Gadducci[a,*], Francesco Santini[c,1,**]

[a]Dipartimento di Informatica, Università di Pisa, Italy
[b]Dipartimento di Informatica, Università di Pisa, Italy
[c]Dipartimento di Matematica e Informatica, Università di Perugia, Italy

**Abstract**

To be added

## 1. Introduction

To be defined

## 2. An introduction to Residuated Monoids

This section reports some results on residuated monoids, which are the algebraic structure adopted for modelling soft constraints in the following of the paper. These background results are mostly drawn from [12], where also proofs can be found.

### *2.1. Preliminaries on Ordered Monoids*

The first step is to define an algebraic structure for modelling preferences, where it is possible to compare values and combine them. Our choice falls into the range of *bipolar* approaches, in order to represent both positive and negative preferences: we refer to [11] for a detailed introduction and a comparison with other proposals.

*Principal corresponding author.
**Secondary corresponding author.
*Email addresses:* laura.bussi@phd.unipi.it (Laura Bussi), gadducci@di.unipi.it (Fabio Gadducci), francesco.santini@dmi.unipg.it (Francesco Santini)

**Definition 1 (Orders).** *A partial order (PO) is a pair $\langle A, \leq \rangle$ such that A is a set and $\leq \subseteq A \times A$ is a reflexive, transitive, and anti-symmetric relation. A semi-lattice (SL) is a PO such that any non-empty finite subset of A has a least upper bound (LUB).*

The LUB of a (possibly infinite or empty) subset $X \subseteq A$ is denoted $\bigvee X$, and it is clearly unique. Should they exist, $\bigvee A$ and $\bigvee \emptyset$ correspond respectively to the top, denoted as $\top$, and to the bottom, denoted as $\perp$, of the PO.

**Definition 2 (Ordered monoids).** *A (commutative) monoid is a triple $\langle A, \otimes, \mathbf{1} \rangle$ such that $\otimes : A \times A \to A$ is a commutative and associative function and $\mathbf{1} \in A$ is its* identity *element, i.e., $\forall a \in A. a \otimes \mathbf{1} = a$. A partially ordered monoid (POM) is a 4-tuple $\langle A, \leq, \otimes, \mathbf{1} \rangle$ such that $\langle A, \leq \rangle$ is a PO and $\langle A, \otimes, \mathbf{1} \rangle$ a monoid. A semi-lattice monoid (SLM) is a POM such that their underlying PO is a SL.*

As usual, we use the infix notation: $a \otimes b$ stands for $\otimes(a, b)$.

**Example 1 (Power set).** *Given a (possibly infinite) set V of variables, we consider the monoid $\langle 2^V, \cup, \emptyset \rangle$ of (finite, possibly empty) subsets of V, with union as the monoidal operator. Since the operator is idempotent (i.e., $\forall a \in A. a \otimes a = a$), the natural order ($\forall a, b \in A. a \leq b$ iff $a \otimes b = b$) is a partial order, and it coincides with subset inclusion: in fact, $\langle 2^V, \subseteq, \cup, \emptyset \rangle$ is an SLM.*

In general, the partial order $\leq$ and the multiplication operator $\otimes$ can be unrelated. This is not the case for distributive SLMs.

**Definition 3 (Distributivity).** *Let $\langle A, \leq, \otimes, \mathbf{1} \rangle$ be an SLM. It is distributive if for any non-empty finite $X \subseteq A$ it holds $\forall a \in A. a \otimes \bigvee X = \bigvee \{a \otimes x \mid x \in X\}$.*

Note that distributivity implies that $\otimes$ is monotone with respect to $\leq$.

**Remark 1.** *It is almost straightforward to show that our proposal encompasses many other formalisms in the literature. Indeed, distributive semi-lattice monoids are* tropical *semirings (also known as dioids), namely, semirings with an idempotent sum operator $a \oplus b$, which in our formalism is obtained as $\bigvee \{a, b\}$. If $\mathbf{1}$ is the top of the SL we end up in* absorptive *semirings [14], which are known as c-semirings in the soft constraint jargon [3] (see e.g. [4] for a brief survey on residuation for such semirings). Note that requiring the monotonicity of $\otimes$ and imposing $\mathbf{1}$ to be the top of the partial order means that preferences are negative, i.e., that it holds $\forall a, b \in A. a \otimes b \leq a$.*

*2.2. Remarks on residuation*

It is often needed to be able to "remove" part of a preference, due e.g. to the non-monotone nature of the language at hand for manipulating constraints. The structure of our choice is given by residuated monoids [14]. They introduce a new operator $\ominus$, which represents a "weak" (due to the presence of partial orders) inverse of $\otimes$.

**Definition 4 (Residuation).** *A residuated POM is a 5-tuple $\langle A, \leq, \otimes, \ominus, \mathbf{1}\rangle$ such that $\langle A, \leq, \otimes, \mathbf{1}\rangle$ is a partially ordered monoid and $\ominus : A \times A \to A$ is a function satisfying $\forall a, b, c \in A. b \otimes c \leq a \iff c \leq a \ominus b$. A residuated SLM is a residuated POM such that the underlying PO is a SL.*

In order to confirm the intuition about weak inverses, Lemma 1 below precisely states that residuation conveys the meaning of an approximated form of subtraction.

**Lemma 1.** *Let $\langle A, \leq, \otimes, \ominus, \mathbf{1}\rangle$ be a residuated POM with bottom. Then $a \ominus b = \bigvee\{c \mid b \otimes c \leq a\}$ for all $a, b \in A$.*

In words, the LUB of the (possibly infinite) set $\{c \mid b \otimes c \leq a\}$ exists and is equal to $a \ominus b$. In fact, residuation implies distributivity (see [11, Lemma 2.2]).

**Lemma 2.** *Let $\langle A, \leq, \otimes, \ominus, \mathbf{1}\rangle$ be a residuated POM. Then $\otimes$ is monotone. If additionally it is a SLM, then it is distributive.*

In any residuated POM the $\ominus$ operator is also monotone on the first argument and anti-monotone on the second one, i.e., $\forall a, b, c \in A. a \leq b \implies c \ominus b \leq c \ominus a$. Other easy to prove properties are $\forall a \in A. \mathbf{1} \leq a \ominus a$ and $\forall a, b \in A. a \leq b \implies b \otimes (a \ominus b) \leq a$. These latter facts suggests the definition below, which identifies sub-classes of residuated monoids that are suitable for an easier manipulation of constraints (see e.g. [4]).

**Definition 5 (Families of POMs).** *A residuated POM $\langle A, \leq, \otimes, \ominus, \mathbf{1}\rangle$ is*

- localised *if* $\forall a \in A. a \notin \{\bot, \top\} \implies a \ominus a = \mathbf{1}$*;*
- invertible *if* $\forall a, b \in A. a \leq b \implies b \otimes (a \ominus b) = a$*;*
- cancellative *if* $\forall a, b, c \in A. a \notin \{\bot, \top\} \land a \otimes b = a \otimes c \implies b = c$*.*

**Remark 2.** *When introduced in [11, Def. 2.4], localisation was equivalently stated as* $\forall a,b \in A. \bot < a \leq b < \top \implies a \ominus b \leq \mathbf{1}$*. Indeed, the latter implies* $a \ominus a \leq \mathbf{1}$*, while* $\mathbf{1} \leq a \ominus a$ *by definition. Now, assuming* $a \ominus a = \mathbf{1}$ *and* $a \leq b$*, by anti-monotonicity* $a \ominus b \leq a \ominus a = \mathbf{1}$*. Note the constraint on* $a \notin \{\bot, \top\}$*: indeed, if a residuated POM has a bottom element then it also has a top element and moreover* $a \ominus \bot = \top \ominus a = \top$ *for any* $a$*.*

*Note that being cancellative is a strong requirement. It implies e.g. some uniqueness of invertibility, that is, for any* $a,b$ *there exists at most a* $c$ *such that* $b \otimes c = a$*. It is moreover equivalent to what we could call strongly locality, that is,* $\forall a,b \in A. a \notin \{\bot, \top\} \implies (a \otimes b) \ominus a = b$*. Indeed, this property implies cancellativeness, since if* $a \otimes b = a \otimes c$ *then* $b = (a \otimes b) \ominus a = (a \otimes c) \ominus a = c$*. On the other side, it is implied, since* $((a \otimes b) \ominus a) \otimes a = a \otimes b$ *holds in residuated POMs.*

## 3. A polyadic approach to constraint manipulation

This section presents our personal take on polyadic algebras for ordered monoids: the standard axiomatisation of e.g. [19] has been completely reworked, in order to be adapted to the constraints formalism. It extends our previous description in [6] by further elaborating on the laws for the polyadic operators in residuated monoids.

### *3.1. Cylindric and Polyadic operators for Ordered Monoids*

We introduce two families of operators that will be used for modelling variables hiding and substitution, which represent key features in languages for manipulating constraints. One is a well-known abstraction for existential quantifiers, the other an axiomatisation of the notion of substitution, and it is proposed as a weaker alternative to diagonals [20], the standard tool for modelling equivalence in constraint programming.[1]

#### *3.1.1. Cylindric operators.*

We fix a POM $\mathbb{S} = \langle A, \leq, \otimes, \mathbf{1} \rangle$ and a set $V$ of variables, and we define a family of cylindric operators axiomatising existential quantifiers.

**Definition 6 (Cylindrification).** *A cylindric operator* $\exists$ *over* $\mathbb{S}$ *and* $V$ *is a family of monotone functions* $\exists_x : A \to A$ *indexed by elements in* $V$ *such that for all* $a,b \in A$ *and* $x,y \in V$

[1] "Weaker alternative" here means that diagonals allow for axiomatising substitutions at the expenses of working with complete partial orders: see e.g. [12, Definition 11].

1. $a \leq \exists_x a$,

2. $\exists_x \exists_y a = \exists_y \exists_x a$,

3. $\exists_x (a \otimes \exists_x b) = \exists_x a \otimes \exists_x b$.

*Let $a \in A$. The* support *of $a$ is the set of variables $sv(a) = \{x \mid \exists_x a \neq a\}$.*

In other words, $\exists$ fixes a monoid action which is monotone and increasing.

*3.1.2. Polyadic operators.*

We now move to define a family of operators axiomatising substitutions. They interact with quantifiers, thus, beside a partially ordered monoid $\mathbb{S}$ and a set $V$ of variables, we fix a cylindric operator $\exists$ over $\mathbb{S}$ and $V$.

As for notation, $F(V)$ is the set of functions with domain and codomain $V$. For a function $\sigma$ we define its support as $sv(\sigma) = \{x \mid \sigma(x) \neq x\}$ and, for a set $X \subseteq V$, we denote by $\sigma\mid_X : X \to V$ the restriction and by $\sigma^c(X)$ the counter-image of $X$ along $\sigma$.

**Definition 7 (Polyadification).** *A polyadic operator $s$ for a cylindric operator $\exists$ is a family of monotone functions $s_\sigma : A \to A$ indexed by elements in $F(V)$ such that for all $a, b \in A$, $x \in V$, and $\sigma, \tau \in F(V)$*

1. $sv(\sigma) \cap sv(a) = \emptyset \implies s_\sigma a = a$,

2. $s_\sigma (a \otimes b) = s_\sigma a \otimes s_\sigma b$,

3. $\sigma\mid_{sv(a)} = \tau\mid_{sv(a)} \implies s_\sigma a = s_\tau a$,

4. $\exists_x s_\sigma a = \begin{cases} s_\sigma \exists_y a & \text{if } \sigma^c(x) = \{y\} \\ s_\sigma a & \text{if } \sigma^c(x) = \emptyset \end{cases}$.

A polyadic operator offers enough structure for modelling variable substitution. In the following, we fix a cylindric operator $\exists$ and a polyadic operator $s$ for it.

*3.2. Cylindric and Polyadic operators for Residuated Monoids*

We now consider the interaction of previous structures with residuation. To this end, in the following we assume that $\mathbb{S}$ is a residuated POM (see Def. 4).

**Lemma 3.** *Let $x \in V$ and $a, b \in A$. Then it holds $\exists_x(a \ominus \exists_x b) \leq \exists_x a \ominus \exists_x b \leq \exists_x(\exists_x a \ominus b)$.*

**Remark 3.** *It is clear that $\exists_x(a \ominus \exists_x b) \leq \exists_x a \ominus \exists_x b$ is actually equivalent to state that $\exists_x(a \otimes \exists_x b) \geq \exists_x a \otimes \exists_x b$.*

We can show that $\ominus$ does not substantially alter the free variables of its arguments.

**Lemma 4.** *Let $a, b \in A$. Then it holds $sv(a \ominus b) \subseteq sv(a) \cup sv(b)$.*

A result similar to Lemma 3 relates residuation and polyadic operators.

**Lemma 5.** *Let $a, b \in A$ and $\sigma \in F(V)$. Then it holds $s_\sigma(a \ominus b) \leq s_\sigma a \ominus s_\sigma b$. Furthermore, if $\sigma$ is invertible, then the equality holds.*

*3.3. Polyadic Soft Constraints*

The key example of polyadic construction comes from soft constraints: our presentation generalises [5], whose underlying algebraic structure is the one of absorptive semirings.

**Definition 8 (Soft constraints).** *Let $V$ be a set of variables, $D$ a finite domain of interpretation and $\mathbb{S} = \langle A, \leq, \otimes, \ominus, \mathbf{1} \rangle$ a residuated SLM. A* (soft) constraint *$c : (V \to D) \to A$ is a function associating a value in $A$ with each assignment $\eta : V \to D$ of the variables.*

The set of constraints forms a residuated SLM $\mathbb{C}$, with the structure lifted from $\mathbb{S}$. Denoting the application of a constraint function $c : (V \to D) \to A$ to a variable assignment $\eta : V \to D$ as $c\eta$, we e.g. have that $c_1 \leq c_2$ if $c_1\eta \leq c_2\eta$ for all $\eta : V \to D$.

**Lemma 6 (Cylindric and polyadic operators for (soft) constraints).** *The residuated SLM of constraints $\mathbb{C}$ admits cylindric and polyadic operators, defined as*

- $(\exists_x c)\eta = \bigvee\{c\rho \mid \eta\mid_{V\setminus\{x\}} = \rho\mid_{V\setminus\{x\}}\}$ *for all $x \in V$,*

- $(s_\sigma c)\eta = c(\eta \circ \sigma)$ *for all* $\sigma \in F(V)$.

**Remark 4.** *Note that* $sv(c)$ *coincides with the classical notion of support for soft constraints. Indeed, if* $x \notin sv(c)$*, then two assignments* $\eta_1, \eta_2 : V \to D$ *differing only for the image of* $x$ *coincide (i.e.,* $c\eta_1 = c\eta_2$*). The cylindric operator is called* projection *in the soft framework, and* $\exists_x c$ *is denoted* $c \Downarrow_{V \setminus \{x\}}$.

For the sake of simplicity, we will use running examples where $D$ is a finite initial segment of the naturals. A constraint can be sometimes simply expressed as an inequation $x \leq 1$, intended as $c\eta = \mathbf{1}$ if $\eta(x) \leq 1$, and $\perp$ otherwise.

## 4. Polyadic Soft CCP: Syntax and reduction semantics

This section introduces our language. We fix a set of variables $V$, ranged over by $x$, $y$, ..., and a residuated POM $\mathbb{S} = \langle \mathscr{C}, \leq, \otimes, \ominus, \mathbf{1} \rangle$, which is cylindric and polyadic over $V$ and whose elements are ranged over by $c$, $d$, ...

**Definition 9 (Agents).** *The set* $\mathscr{A}$ *of agents, parametric with respect to a set* $\mathscr{P}$ *of (unary) procedure declarations* $p(x) = A$*, is given by the following grammar*

$$A ::= \ \mathbf{stop} \mid \mathbf{tell}(c) \mid \mathbf{ask}(c) \mapsto A \mid A \parallel A \mid p(x) \mid \exists_x A$$

In the following we consider a set $\mathscr{E}$ of extended agents that uses the existential operator $\exists_x^\pi A$, where $\pi \in \mathscr{C}^*$ is meant to represent the sequence of updates performed on the local store. More precisely, the extended agent may carry some information about the hidden variable $x$ in an incremental way. We will often write $\exists_x A$ for $\exists_x^{[]} A$ and $\pi_i$ for the $i$-th element of $\pi = [\pi_0, \ldots, \pi_n]$.

We denote by $fv(A)$ the set of free variables of an (extended) agent, defined in the expected way by structural induction, assuming that $fv(\mathbf{tell}(c)) = sv(c)$, $fv(\mathbf{ask}(c) \mapsto A) = sv(c) \cup fv(A)$, and $fv(\exists_x^\pi A) = (fv(A) \cup \bigcup_i sv(\pi_i)) \setminus \{x\}$. In the following, we restrict our attention to procedure declarations $p(x) = A$ such that $fv(A) = \{x\}$.

**Definition 10 (Substitutions).** *Let* $[^y/_x] : V \to V$ *be the substitution defined as*

$$[^y/_x](w) = \begin{cases} y & \text{if } w = x \\ w & \text{otherwise} \end{cases}.$$

*It induces an operator $[^y/_x] : \mathscr{E} \to \mathscr{E}$ on extended agents as expected, in particular*

$$(\exists_w^\pi A)[^y/_x] = \begin{cases} \exists_w^{(s_{[^y/_x]}\pi)} A[^y/_x] & \textit{if } w \notin \{x, y\} \\ (\exists_z^{(s_{[^z/_w]}\pi)} A[^z/_w])[^y/_x] & \textit{for } z \notin fv(\exists_w^\pi A) \textit{ otherwise} \end{cases}$$

*with $s_{[^y/_x]}[\pi_1, \ldots, \pi_n]$ a shorthand for $[s_{[^y/_x]}\pi_1, \ldots, s_{[^y/_x]}\pi_n]$.*

Note that the choice of $z$ in the rule above is immaterial, since for the polyadic operator it holds $\exists_x c = \exists_y s_{[^y/_x]}(c)$ if $y \notin sv(c)$. In the following we consider terms to be equivalent up-to $\alpha$-conversion, meaning that terms differing only for hidden variables are considered equivalent, i.e., $\exists_w^\pi A = \exists_z^{(s_{[^z/_w]}\pi)} A[^z/_w]$ for $z \notin fv(\exists_w^\pi A)$.

**Lemma 7.** *Let $A \in \mathscr{E}$ and $x \notin fv(A)$. Then $A[^y/_x] = A$.*

### *4.1. Reduction semantics*

We now move to the reduction semantics of our calculus. Given a sequence $\pi = [\pi_1, \ldots, \pi_n]$, we will use $\pi_\otimes$ and $\exists_x \pi$ as shorthands for $\pi_1 \otimes \ldots \otimes \pi_n$ and $[\exists_x \pi_1, \ldots, \exists_x \pi_n]$, respectively, sometimes combining them as in $(\exists_x \pi)_\otimes$, with $[]_\otimes = \mathbf{1}$.

**Definition 11 (Reductions).** *Let $\Gamma = \mathscr{E} \times \mathscr{C}$ be the set of* configurations*. The* direct reduction semantics *for SCCP is the pair $\langle \Gamma, \to \rangle$ such that $\to \subseteq \Gamma \times \Gamma$ is the binary relations obtained by the rules in Table 1.*

*The* reduction semantics *for SCCP is the pair $\langle \Gamma, \to \rangle$ such that $\to \subseteq \Gamma \times \Gamma$ is the binary relation obtained by the rules in Table 1 and Table 2.*

The split distinguishes between the axioms and the rules guaranteeing the closure with respect to the parallel and existential operators. Indeed, rule **R1** models the interleaving of two agents in parallel, assuming for the sake of simplicity that the parallel operator is associative and commutative, as well as satisfying $\mathbf{stop} \parallel A = A$. In **A1** a constraint $c$ is added to the store $\sigma$. **A2** checks if $c$ is entailed by $\sigma$: if not, the computation is blocked. Axiom **A3** replaces a procedure identifier with the associated body, renaming the formal parameter with the actual one.

Let us instead discuss in some details the rule **R2**. The intuition is that if we reach an agent $\langle \exists_x^\pi A, \sigma \rangle$, then during the computation a sequence $\pi$ of updates has

| | | | |
|---|---|---|---|
| **A1** | $\langle \mathbf{tell}(c), \sigma \rangle \rightarrow \langle \mathbf{stop}, \sigma \otimes c \rangle$ | **Tell** |
| **A2** | $\dfrac{\sigma \leq c}{\langle \mathbf{ask}(c) \mapsto A, \sigma \rangle \rightarrow \langle A, \sigma \rangle}$ | **Ask** |
| **A3** | $\dfrac{p(x) = A \in \mathscr{P}}{\langle p(y), \sigma \rangle \rightarrow \langle A[^y/_x], \sigma \rangle}$ | **Rec** |

Table 1: Axioms of the reduction semantics for SCCP.

| | | |
|---|---|---|
| **R1** | $\dfrac{\langle A, \sigma \rangle \rightarrow \langle A', \sigma' \rangle}{\langle A \parallel B, \sigma \rangle \rightarrow \langle A' \parallel B, \sigma' \rangle}$ | **Par1** |
| **R2** | $\dfrac{\langle A, \pi_0 \otimes \sigma \rangle \rightarrow \langle B, \sigma_1 \rangle \text{ with } \pi_0 = \pi_\otimes \ominus (\exists_x \pi)_\otimes}{\langle \exists_x^\pi A, \sigma \rangle \rightarrow \langle \exists_x^{\pi\rho} B, \sigma \otimes \exists_x \rho \rangle \text{ with } \rho = \sigma_1 \ominus (\pi_0 \otimes \sigma)} \text{ for } x \notin sv(\sigma)$ | **Hide** |

Table 2: Contextual rules of the reduction semantics for SCCP.

been performed by the local agent and $(\exists_x \pi)_\otimes$ has been added to the global store. In order to evaluate $A$, the chosen store is $\pi_0 \otimes \sigma$ for $\pi_0 = \pi_\otimes \ominus (\exists_x \pi)_\otimes$: the effect $(\exists_x \pi)_\otimes$ of the sequence of updates is removed from the local store $\pi_\otimes$, which may carry information about $x$, since that effect had been previously added to the global store. Now, $\rho = \sigma_1 \ominus (\pi_0 \otimes \sigma)$ is precisely the information added by the step originating from $A$, which is then restricted and added to $\sigma$. On the local store we simply add that effect $\rho$ to the sequence of updates, with $\pi\rho = [\pi_0, \ldots, \pi_n, \rho]$.

**Lemma 8 (On monotonicity).** *Let $\langle A, \sigma \rangle \rightarrow \langle B, \rho \rangle$ be a reduction. Then $\rho = (\rho \ominus \sigma) \otimes \sigma$ and $fv(\langle B, \rho \rangle) \subseteq fv(\langle A, \sigma \rangle)$.*

**Remark 5.** *With respect to the crisp language with local variables introduced in [1], which can be recasted in our framework as absorptive POMs where the monoidal operator is idempotent, our proposal differ mostly for the the structure of rule* **R2***, which could be presented as below*

$$\frac{\langle A, \pi_0 \otimes \sigma \rangle \rightarrow \langle B, \xi \otimes \pi_0 \otimes \sigma \rangle \textit{ with } \pi_0 = \pi_\otimes \ominus (\exists_x \pi)_\otimes}{\langle \exists_x^\pi A, \sigma \rangle \rightarrow \langle \exists_x^{\pi\xi} B, \sigma \otimes \exists_x \xi \rangle} \textit{ for } x \notin sv(\sigma)$$

*The proposals coincide for e.g. cancellative monoids, since inverses are unique. However, this is not so if the monoidal operator is idempotent, thus the crisp rule* *represents in fact a schema, giving rise to a possibly infinite family of reductions departing from an agent. Our choice of a chosen witness $\exists_x \sigma_1 \ominus (\pi_0 \otimes \sigma)$ avoids such non-determinism.*

Let $\gamma = \langle A, \sigma \rangle$ be a configuration. We denote by $fv(\gamma)$ the set $fv(A) \cup sv(\sigma)$ and by $\gamma[^z/_w]$ the component-wise application of the substitution $[^z/_w]$.

**Definition 12.** *A configuration $\langle A, \sigma \rangle$ is initial if $A \in \mathscr{A}$ and $\sigma = \mathbf{1}$; it is reachable if it can be reached by an initial configuration via a sequence of reductions.*

**Lemma 9 (On monotonicity, II).** *Let $\langle A \parallel \exists_x^\pi B, \sigma \rangle$ be a reachable configuration. Then $\sigma = (\sigma \ominus (\exists_x \pi)_\otimes) \otimes (\exists_x \pi)_\otimes$.*

**Remark 6.** *An alternative solution for the the structure of rule* **R2** *would have been*

$$\frac{\langle A, \pi_\otimes \otimes \sigma_0 \rangle \rightarrow \langle B, \sigma_1 \rangle \textit{ with } \sigma_0 = \sigma \ominus (\exists_x \pi)_\otimes}{\langle \exists_x^\pi A, \sigma \rangle \rightarrow \langle \exists_x^{\pi\rho} B, \sigma \otimes \exists_x \rho \rangle \textit{ with } \rho = \sigma_1 \ominus (\pi_\otimes \otimes \sigma_0)} \textit{ for } x \notin sv(\sigma)$$

*Indeed, in the light of Lemma 9, the proposals coincide for invertible semirings, since $\pi_0 \otimes \sigma = (\pi_\otimes \ominus (\exists_x \pi)_\otimes) \otimes (\exists_x \pi)_\otimes \otimes (\sigma \ominus (\exists_x \pi)_\otimes) \leq \pi_\otimes \otimes (\sigma \ominus (\exists_x \pi)_\otimes)$, and the equality holds for invertible semirings since $\pi_\otimes \leq (\exists_x \pi)_\otimes$.*

### *4.2. Saturated bisimulation*

As proposed in [1] for crisp languages, we define a barbed equivalence between two agents [16]. Intuitively, barbs are basic observations (predicates) on the states of a system, and in our case they correspond to the constraints in $\mathscr{C}$.

**Definition 13 (Barbs).** *Let $\langle A, \sigma \rangle$ be a configuration and $c \in \mathscr{C}$. We say that $\langle A, \sigma \rangle$ verifies $c$, or that $\langle A, \sigma \rangle \downarrow_c$ holds, if $\sigma \leq c$.*

In other terms, satisfying a barb $c$ means that $\mathbf{ask}(c)$ must be enabled in $\langle A, \sigma \rangle$. We now move to consider equivalence and, along [1] we propose the use of *saturated bisimilarity* in order to obtain a congruence.

**Definition 14 (Saturated bisimilarity).** *A saturated bisimulation is a symmetric relation $R$ on configurations such that whenever $(\langle A, \sigma \rangle, \langle B, \rho \rangle) \in R$*

1. *if $\langle A, \sigma \rangle \downarrow_c$ then $\langle B, \rho \rangle \downarrow_c$;*

| | | | |
|---|---|---|---|
| **LA1** | $\langle \mathbf{tell}(c), \sigma \rangle \xrightarrow{\mathbf{1}} \langle \mathbf{stop}, \sigma \otimes c \rangle$ | | **Tell** |
| **LA2** | $\dfrac{\alpha \leq c \ominus \sigma}{\langle \mathbf{ask}(c) \mapsto A, \sigma \rangle \xrightarrow{\alpha} \langle A, \alpha \otimes \sigma \rangle}$ | | **Ask** |
| **LA3** | $\dfrac{p(x) = A \in \mathscr{P}}{\langle p(y), \sigma \rangle \xrightarrow{\mathbf{1}} \langle A[^y/_x], \sigma \rangle}$ | | **Rec** |

Table 3: Axioms of the labelled semantics for SCCP.

2. *if* $\langle A, \sigma \rangle \to \gamma_1$ *then there is* $\gamma_2$ *such that* $\langle B, \rho \rangle \to \gamma_2$ *and* $(\gamma_1, \gamma_2) \in R$*;*
3. $(\langle A, \sigma \otimes d \rangle, \langle B, \rho \otimes d \rangle) \in R$ *for all* $d$*.*

*We say that* $\gamma_1$ *and* $\gamma_2$ *are saturated bisimilar (*$\gamma_1 \sim_s \gamma_2$*) if there exists a saturated bisimulation* $R$ *such that* $(\gamma_1, \gamma_2) \in R$*. We write* $A \sim_s B$ *if* $\langle A, \mathbf{1} \rangle \sim_s \langle B, \mathbf{1} \rangle$*.*

Note that $\langle A, \sigma \rangle \sim_s \langle B, \rho \rangle$ implies that $\sigma = \rho$. Moreover, it is also a congruence. Indeed, a context $C[\cdot]$, i.e., an agent with an placeholder $\cdot$, can modify the behaviour of a configuration only by adding constraints to its store.

**Proposition 1.** *Let* $A \sim_s B$ *and* $C[\cdot]$ *a context. Then* $C[A] \sim_s C[B]$*.*

## 5. Labelled reduction semantics

The definition of $\sim_s$ is unsatisfactory because of the store closure, i.e., the quantification in condition *3* of Definiton 14. This section presents a labelled version of the reduction semantics that allow to partially avoid such drawback.

**Definition 15 (Labelled reductions).** *Let* $\Gamma = \mathscr{A} \times \mathscr{C}$ *be the set of* configurations. *The* labelled direct reduction semantics *for SCCP is the pair* $\langle \Gamma, \to \rangle$ *such that* $\to \subseteq \Gamma \times \mathscr{C} \times \Gamma$ *is the ternary relation obtained by the rules in Table 3.*

*The* labelled reduction semantics *for SCCP is the pair* $\langle \Gamma, \to \rangle$ *such that* $\to \subseteq \Gamma \times \mathscr{C} \times \Gamma$ *is the ternary relation obtained by the rules in Table 3 and Table 4.*

In Table 3 and Table 4 we refine the notion of transition (respectively given in Table 1 and Table 2) by adding a label that carries additional information about the constraints that cause the reduction. Indeed, rules in Table 3 and Table 4 mimic those in Table 1 and Table 2, except for a constraint $\alpha$ that represents the

| | | | |
|---|---|---|---|
| **LR1** | $\dfrac{\langle A, \sigma \rangle \xrightarrow{\alpha} \langle A', \sigma' \rangle}{\langle A \parallel B, \sigma \rangle \xrightarrow{\alpha} \langle A' \parallel B, \sigma' \rangle}$ | | **Par** |
| **LR2** | $\dfrac{\langle A, \pi_0 \otimes \sigma \rangle \xrightarrow{\alpha} \langle B, \sigma_1 \rangle \text{ with } \pi_0 = \pi_\otimes \ominus (\exists_x \pi)_\otimes}{\langle \exists_x^\pi A, \sigma \rangle \xrightarrow{\alpha} \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho \rangle \text{ with } \rho = \sigma_1 \ominus (\alpha \otimes \pi_0 \otimes \sigma)}$ | for $x \notin sv(\sigma) \cup sv(\alpha)$ | **Hide** |

Table 4: Contextual rules of the labelled semantics for SCCP.

additional information that must be combined with $\sigma$ in order to fire an action from $\langle A, \sigma \rangle$ to $\langle A', \sigma' \rangle$.

For the rules in Table 3, as well as for rule **LR1**, we can restate the intuition given for their unlabelled counterparts. The difference concerns the axioms for the $\mathbf{ask}(c)$: if $c$ is not entailed from $\sigma$, then some additional information is imported from the environment, ensuring that the state $\alpha \otimes \sigma \leq c$ allows the execution of $\mathbf{ask}(c)$.

Once again, the more complex axiom is **LR2**. With respect to **R2**, the additional intuition is that $\alpha$ should not contain the restricted variable $x$: additional information can be obtained from the environment, as long as it does not interact with data that are private to the local agent. Note that by choosing $\rho = \sigma_1 \ominus (\alpha \otimes \pi_0 \otimes \sigma)$ we are removing $\alpha$ from the update to be memorised in the local store. However, since $\alpha$ is added to the global store, it will not be necessary to receive it again in the future.

**Remark 7.** *Concerning the rule* **LA2***, an alternative solution would have been to restrict the possible reductions to the one with maximal label, that is,* $\langle \mathbf{ask}(c) \mapsto A, \sigma \rangle \xrightarrow{c \ominus \sigma} \langle A, (c \ominus \sigma) \otimes \sigma \rangle$*. However, this might have been restrictive in combination with rule* **LR2***. Consider our running example and the configuration* $\langle \exists_x^{[x>1]} \mathbf{ask}(y > 2) \mapsto \mathbf{stop}, \mathbf{1} \rangle$*. The initial configuration in the premise is* $\langle \mathbf{ask}(y > 2) \mapsto \mathbf{stop}, x > 1 \rangle$ *and* $(y > 3) \ominus (x > 1) = (x \leq 1) \vee (y > 2)$*. Selecting* $\alpha = (x \leq 1) \vee (y > 2)$ *is problematic, since x occurs free. Instead, the choice of* $\alpha = (y > 2)$*, or any other value such as* $y > 3$*,* $y > 4$*, …, fits well with the intuition of information from the environment triggering the reduction.*

*Note instead that the choice of removing the requirement* $x \notin sv(\alpha)$ *and put* $\exists_x \alpha$ *as label in the conclusion of rule* **LR2** *would be too liberal. It would work in our previous example, since* $\exists_x((x \leq 1) \vee (y > 3)) = y > 3$*. However, consider e.g. the configuration* $\gamma = \langle \exists_x^{[x>1]} \mathbf{ask}(x > 2) \mapsto \mathbf{stop}, \mathbf{1} \rangle$*. We would have that* $\langle \mathbf{ask}(x > 2) \mapsto \mathbf{stop}, x > 1 \rangle \xrightarrow{x \neq 2} \langle \mathbf{stop}, x > 2 \rangle$*, and then allowing the reduction*

$\gamma \xrightarrow{\mathbf{1}} \langle \exists_x^{[x>1,x\neq 2]} \mathbf{stop}, \mathbf{1} \rangle$*, which clashes with the intuition that receiving information should not enable reductions involving (necessarily) the restricted variable.*

**Lemma 10 (On labelled monotonicity).** *Let* $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \rho \rangle$ *be a labelled reduction. Then* $\rho = (\rho \ominus (\alpha \otimes \sigma)) \otimes \alpha \otimes \sigma$ *and* $fv(\langle B, \rho \rangle) \subseteq fv(\langle A, \sigma \rangle) \cup sv(\alpha)$*.*

**Remark 8.** *We will later prove that if* $\mathbb{S}$ *is localised and* $\alpha \neq \mathbf{1}$ *then* $\rho \ominus (\alpha \otimes \sigma) = \mathbf{1}$*. In other terms, if* $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \rho \rangle$ *is a labelled reduction and* $\alpha \neq \mathbf{1}$*, then* $\rho = \alpha \otimes \sigma$*. Indeed, since* $\alpha \neq \mathbf{1}$ *its derivation must use the axiom* **LA2***. Consider e.g. a labelled reduction* $\langle \exists_x^{\pi} A, \sigma \rangle \xrightarrow{\alpha} \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho \rangle$*. If* $\alpha \neq \mathbf{1}$*, then* $\rho = \mathbf{1}$*. Indeed, this is the expected behaviour: if an input from the context is needed, there is no contribution by the agent to the local store, hence the update is correctly* $\mathbf{1}$*.*

**Definition 16.** *A configuration is l-reachable if it can be reached by an initial configuration via a sequence of labelled reductions.*

**Lemma 11 (On labelled monotonicity, II).** *Let* $\langle B \parallel \exists_x^{\pi} C, \sigma \rangle$ *be an l-reachable configuration. Then* $\sigma = (\sigma \ominus (\exists_x \pi)_{\otimes}) \otimes (\exists_x \pi)_{\otimes}$*.*

## 6. Semantics correspondence and labelled bisimilarity

We collect further formal results in two different subsections: Section 6.1 proves the correspondence between the two unlabelled and labelled semantics, while Section 6.2 propose a bisimilarity reduction for the labelled semantics.

### *6.1. On the correspondence between reduction semantics*

This section shows the connection between labelled and unlabelled reduction semantics.

**Theorem 1 (Soundness).** *If* $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \sigma' \rangle$ *then* $\langle A, \alpha \otimes \sigma \rangle \rightarrow \langle B, \sigma' \rangle$*.*

**Proof 1.** *We proceed by induction and we will prove a slightly stronger proposition, namely, that the two reductions have equivalent proofs, namely, they use axioms and rules in the same order, up-to the obvious renaming (i.e.,* **LA1** *for* **A1***, and so on).*

*The property holds for the axioms, since e.g. for* **LA2** *we know that* $\alpha \leq c \ominus \sigma$ *implies* $\alpha \otimes \sigma \leq c$ *for residuated POMs. We then proceed by induction on rule derivations, presenting only the proof for rule* **LR2**. *We have*

$$\frac{\langle A, \pi_0 \otimes \sigma\rangle \xrightarrow{\alpha} \langle B, \sigma_1\rangle \text{ with } \pi_0 = \pi_{\otimes} \ominus (\exists_x \pi)_{\otimes}}{\langle \exists_x^{\pi} A, \sigma\rangle \xrightarrow{\alpha} \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho\rangle \text{ with } \rho = \sigma_1 \ominus (\alpha \otimes \pi_0 \otimes \sigma)}$$

*for* $x \notin sv(\sigma) \cup sv(\alpha)$. *By induction hypothesis* $\langle A, \alpha \otimes \pi_0 \otimes \sigma\rangle$ *is reachable and*

$$\langle A, \alpha \otimes \pi_0 \otimes \sigma\rangle \rightarrow \langle B, \sigma_1\rangle$$

*From this it follows by* **LR2** *that* $\langle \exists_x^{\pi} A, \alpha \otimes \sigma\rangle$ *is reachable and*

$$\langle \exists_x^{\pi} A, \alpha \otimes \sigma\rangle \rightarrow \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho\rangle$$

*and we are done.* □

The theorem above can be easily reversed, saying that if a configuration $\langle A, \sigma\rangle$ is reachable, then it is also l-reachable via a sequence of reductions labelled by $\mathbf{1}$.

**Proposition 2.** *If* $\langle A, \sigma\rangle \rightarrow \langle B, \sigma'\rangle$ *then* $\langle A, \sigma\rangle \xrightarrow{\mathbf{1}} \langle B, \sigma'\rangle$.

These results also ensure that a configuration is reachable iff it is l-reachable. However, we are interested in a more general notion of completeness, possibly taking into account reductions needing a label. For this, we first need some technical lemmas.

Now, note that the proof of every (labelled) reduction is given by the choice of an axiom and a series of applications of the rules **LR1** and **LR2**. Also, note that if $\langle A, \sigma\rangle \xrightarrow{\alpha} \langle B, \sigma'\rangle$ is a reduction via the axiom **LA1**, then $\alpha = \mathbf{1}$.

**Lemma 12 (Completeness, I).** *Let* $\langle A, \tau\rangle \xrightarrow{\mathbf{1}} \langle B, \tau'\rangle$ *be a reduction via the axiom* **LA1**. *If* $\mathscr{C}$ *is cancellative then for every* $\sigma$ $\langle A, \sigma\rangle \xrightarrow{\mathbf{1}} \langle B, \sigma'\rangle$ *and* $\tau' \ominus \tau = \sigma' \ominus \sigma$.

**Lemma 13 (Completeness, II).** *Let* $\langle A, \tau\rangle \xrightarrow{\beta} \langle B, \tau'\rangle$ *be a reduction via the axiom* **LA2**. *If* $\mathscr{C}$ *is localised then* $\tau' = \beta \otimes \tau$ *and for every* $\sigma$ *if* $\alpha \leq (\beta \otimes \tau) \ominus \sigma$ *then* $\langle A, \sigma\rangle \xrightarrow{\alpha} \langle B, \alpha \otimes \sigma\rangle$.

Clearly $\alpha = (\beta \otimes \tau) \ominus \sigma$ is a possible witness. Note however that it might be that $\beta \otimes \tau \not\leq \alpha \otimes \sigma$, e.g. if $\sigma = \bot$, in which case $\alpha = \top$.

*6.2. Labelled bisimulation*

We now exploit the labelled reductions in order to define suitable notion of bisimilarity without the upward closure condition. As it occurs with the crisp language [1] and the soft variant with global variables [6], barbs cannot be removed from the definition of bisimilarity because they cannot be inferred by the reductions.

**Definition 17 (Strong bisimilarity).** *A strong bisimulation is a symmetric relation $R$ on configurations such that whenever $(\langle A, \sigma\rangle, \langle B, \rho\rangle) \in R$*

1. *if $\langle A, \sigma\rangle \downarrow_c$ then $\langle B, \rho\rangle \downarrow_c$;*
2. *if $\langle A, \sigma\rangle \xrightarrow{\alpha} \gamma_1$ then there is $\gamma_2$ such that $\langle B, \alpha \otimes \rho\rangle \rightarrow \gamma_2$ and $(\gamma_1, \gamma_2) \in R$;*
3. *$(\langle A, \sigma \otimes d\rangle, \langle B, \rho \otimes d\rangle) \in R$ for all $d$ such that $\sigma \otimes d \not\leq \sigma$.*

*We say that $\gamma_1$ and $\gamma_2$ are strongly bisimilar ($\gamma_1 \sim \gamma_2$) if there exists a strong bisimulation $R$ such that $(\gamma_1, \gamma_2) \in R$. We write $A \sim B$ if $\langle A, \mathbf{1}\rangle \sim \langle B, \mathbf{1}\rangle$.*

Note that $\langle A, \sigma\rangle \sim \langle B, \rho\rangle$ implies $\sigma = \rho$, as for saturated bisimilarity. We improved on the feasibility of $\sim$ by requiring that the equivalence is upward closed only whenever the store may be worsened. Note that in some cases, e.g. when $\mathscr{C}$ is absorptive (as in [1]), the clause is vacuous. However, thanks to the correspondence results in Section 6.1, it can be proved upward closed for all $d$, and thus also a congruence.

**Proposition 3.** *Let $\langle A, \sigma\rangle \sim \langle B, \rho\rangle$ and $d \in \mathscr{C}$. If $\mathscr{C}$ is cancellative then $\langle A, \sigma \otimes d\rangle \sim \langle B, \rho \otimes d\rangle$.*

**Proof 2.** *We need to show that the relation $R = \{(\langle A, \sigma \otimes d\rangle, \langle B, \sigma \otimes d\rangle) \mid \langle A, \sigma\rangle \sim \langle B, \sigma\rangle\}$ is a labelled bisimulation. We then assume that $\langle A, \sigma \otimes d\rangle \xrightarrow{\beta} \langle A', \sigma'\rangle$: we need to prove that there exists $B'$ such that $\langle B, \beta \otimes \sigma \otimes d\rangle \rightarrow \langle B', \sigma'\rangle$ and $(\langle A', \sigma'\rangle, \langle B', \sigma'\rangle) \in R$.*

*By soundness $\langle A, \beta \otimes \sigma \otimes d\rangle \rightarrow \langle A', \sigma'\rangle$ with the same proof. We then distinguish two cases on the axiom used.*

*[**LA1**] By Lemma 13 (completeness for **LA1**) we have $\langle A, \sigma\rangle \xrightarrow{\mathbf{1}} \langle A', \sigma''\rangle$ and $\sigma' \ominus (\beta \otimes \sigma \otimes d) = \sigma'' \ominus \sigma$, and by Lemma 9 (monotonicity) we have that $\sigma' = \sigma'' \otimes \beta \otimes d$. Since $\langle A, \sigma\rangle \sim \langle B, \sigma\rangle$, there exists $B'$ such that $\langle B, \sigma\rangle \rightarrow$*

$\langle B', \sigma'' \rangle$ *and* $\langle A', \sigma'' \rangle \sim \langle B', \sigma'' \rangle$*, and it suffices to look at the cases where* $\beta \otimes d \leq \mathbf{1}$*. Now* $\langle B, \sigma \rangle \xrightarrow{\mathbf{1}} \langle B', \sigma'' \rangle$*, and we may now check the proof of such reduction. If it is* **LA1***, we retrace the same steps as before and* $\langle B, \beta \otimes \sigma \otimes d \rangle \xrightarrow{\mathbf{1}} \langle B', \sigma' \rangle$ *with* $\sigma' = \sigma'' \otimes \beta \otimes d$*, and we are done. If it is* **LA2***, it suffices to note that* $\beta \otimes \sigma \otimes d \leq \sigma$*, hence* $\langle B, \sigma \otimes \beta \otimes d \rangle \xrightarrow{\mathbf{1}} \langle B', \sigma'' \otimes \beta \otimes d \rangle$ *and we are done.*

*[***LA2***] By Lemma 14 (completeness for* **LA2***) we have* $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle A', \alpha \otimes \sigma \rangle$ *for any* $\alpha \leq (\beta \otimes \sigma \otimes d) \ominus \sigma$*. Since* $\langle A, \sigma \rangle \sim \langle B, \sigma \rangle$*, there exists* $B'$ *such that* $\langle B, \alpha \otimes \sigma \rangle \rightarrow \langle B', \alpha \otimes \sigma \rangle$*, and by taking* $\alpha = (\beta \otimes \sigma \otimes d) \ominus \sigma$ *we are done.*

As for the unlabelled case (Proposition 1), strong bisimilarity is a congruence.

**Proposition 4.** *Let* $A \sim B$ *and* $C[\cdot]$ *a context. If* $\mathscr{C}$ *is cancellative then* $C[A] \sim C[B]$*.*

Finally, we can state the correspondence between our bisimilarity semantics.

**Theorem 2.** $\sim_s \subseteq \sim$*. Moreover, if* $\mathscr{C}$ *is cancellative, then the equality holds.*

## 7. Related works

As it is possible to appreciate from the survey in [17], the literature on CCP languages is quite ample. In the following of this section we briefly summarise proposals that consider both local and global stores, and information mobility.

One of the most related work is represented by [1]. Anyhow, the differences are significant: in that work the underlying constraint systems is crisp, as it can only deal with hard constraints (which indeed we can do as well). For this reason, there is no need for defining a *residuated* preference system, which allows us to use bipolar preferences. Furthermore, the authors of [1] adopt a cylindric algebra instead of a polyadic one, as introduced in Section 1. Finally, as already noted in Remark 5 in this paper, the use of the local store is different with respect to our approach. Since the monoidal operator is idempotent, in [1] the semantics of the hiding operator is simply presented as $\langle \exists_x^e A, \sigma \rangle \rightarrow \langle \exists_x^{e'} B, \sigma \otimes \exists_x e' \rangle$ if $\langle A, e \otimes \exists_x \sigma \rangle \rightarrow \langle B, e' \otimes \exists_x \sigma \rangle$. Since we have introduced polyadic operators, with their simpler representation of substitutions, and thus we consider agents up-to $\alpha$-conversion, we can replace $\exists_x \sigma$ with $\sigma$ by requiring that $x \not\in sv(\sigma)$. Most importantly, in [1] the local store $e$ is used to fire a step that only changes the local store to $e'$, and this change is visible in the global store except for the effect on variable $x$. However, this rule is intrinsically non-deterministic, since many such $e'$ can exist. Moreover, since we are not idempotent we cannot add the whole $e'$ to both the local and the global stores, but only the "difference" between $e'$ and $e$ at each step.

In [15] the authors describe a *spatial* constraint systems with operators to specify information and processes moving from a space to another. Such a language provides for the specification of spatial mobility and epistemic concepts such as belief, utterance and lies: besides local stores for agents (representing belief), it can express the epistemic notion of knowledge by means of a derived spatial operator that specifies global shared information. Differently from this work, our approach focuses on preferences, on the concurrent language on top of the system, and on process equivalences.

The process calculi in [2, 9] provide to agents the use of assertions within $\pi$-like processes. A soft language is adopted in [9]: from a variant of $\pi$-calculus it inherits *explicit fusions*, i.e. simple constraints expressing name equalities, in order to pass constraints from an agent to another. However the algebraic structure is neither residuated nor polyadic; in addition, no process-equivalence relation is proposed. In [18, 10] processes can send constraints using communication channels much like in the $\pi$-calculus.

A further language that uses $\pi$-calculus features to exchange constraints between agents, but this time with a probabilistic semantics, is shown in [7]. A congruence relation and a labelled transition system are also shown in the paper.

In [13] the authors propose an extension of the CCP language with the purpose to model process migration within a hierarchical network. Agents brings their local store when they migrate. In [8] the authors enrich a CCP language with the possibility to share (read/write) the information in the global store, and communicate with other agents (via multi-party or handshake).

All the systems described in this section are based on hard constraints, and they do not consider preferences associated with constraints (except [9], whose algebraic structure is however less general). In addition, a very few proposals formalise process equivalences by providing a deeper investigation of the semantics.

## 8. Conclusions and further works

To be finished

## References


[1] Aristizábal, A., Bonchi, F., Palamidessi, C., Pino, L.F., Valencia, F.D.: Deriving labels and bisimilarity for concurrent constraint programming. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 138–152. Springer (2011)

[2] Bengtson, J., Johansson, M., Parrow, J., Victor, B.: Psi-calculi: Mobile processes, nominal data, and logic. In: LICS 2009. pp. 39–48. IEEE Computer Society (2009)

[3] Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. Journal of the ACM 44(2), 201–236 (1997)

[4] Bistarelli, S., Gadducci, F.: Enhancing constraints manipulation in semiring-based formalisms. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) ECAI 2006. FAIA, vol. 141, pp. 63–67. IOS Press (2006)

[5] Bistarelli, S., Montanari, U., Rossi, F.: Soft concurrent constraint programming. ACM Transactions on Computational Logic 7(3), 563–589 (2006)

[6] Bonchi, F., Bussi, L., Gadducci, F., Santini, F.: Polyadic soft constraints. In: Alvim, M.S., Chatzikokolakis, K., Olarte, C., Valencia, F. (eds.) The Art of Modelling Computational Systems. LNCS, vol. 11760, pp. 241–257. Springer (2019)

[7] Bortolussi, L., Wiklicky, H.: A distributed and probabilistic concurrent constraint programming language. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 143–158. Springer (2005)

[8] Brim, L., Kretínský, M., Jacquet, J., Gilbert, D.R.: Modelling multi-agent systems as synchronous concurrent constraint processes. Computers and Artificial Intelligence 21(6) (2002)

[9] Buscemi, M.G., Montanari, U.: Open bisimulation for the concurrent constraint pi-calculus. In: Drossopoulou, S. (ed.) ESOP 2008. LNCS, vol. 4960, pp. 254–268. Springer (2008)

[10] Díaz, J.F., Rueda, C., Valencia, F.D.: Pi+- calculus: A calculus for concurrent processes with constraints. CLEI Electronic Journal 1(2) (1998)

[11] Gadducci, F., Santini, F.: Residuation for bipolar preferences in soft constraints. Information Processing Letters 118, 69–74 (2017)

[12] Gadducci, F., Santini, F., Pino, L.F., Valencia, F.D.: Observational and behavioural equivalences for soft concurrent constraint programming. Journal of Logic and Algebraic Methods in Programming 92, 45–63 (2017)

[13] Gilbert, D.R., Palamidessi, C.: Concurrent constraint programming with process mobility. In: Lloyd, J.W., Dahl, V., Furbach, U., Kerber, M., Lau, K., Palamidessi, C., Pereira, L.M., Sagiv, Y., Stuckey, P.J. (eds.) CL 2000. LNCS, vol. 1861, pp. 463–477. Springer (2000)

[14] Golan, J.: Semirings and Affine Equations over Them. Kluwer (2003)

[15] Guzmán, M., Haar, S., Perchy, S., Rueda, C., Valencia, F.D.: Belief, knowledge, lies and other utterances in an algebra for space and extrusion. Journal of Logic and Algebraic Methods in Programming 86(1), 107–133 (2017)

[16] Milner, R., Sangiorgi, D.: Barbed bisimulation. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 685–695. Springer (1992)

[17] Olarte, C., Rueda, C., Valencia, F.D.: Models and emerging trends of concurrent constraint programming. Constraints 18(4), 535–578 (2013)

[18] Réty, J.: Distributed concurrent constraint programming. Fundamenta Informaticae 34(3), 323–346 (1998)

[19] Sági, G.: Polyadic algebras. In: Andréka, H., Ferenczi, M., Németi, I. (eds.) Cylindric-like algebras and algebraic logic, Bolyai Society Mathematical Studies, vol. 22, pp. 367–389. Springer (2013)

[20] Saraswat, V.A., Rinard, M.C., Panangaden, P.: Semantic foundations of concurrent constraint programming. In: Wise, D.S. (ed.) POPL 1991. pp. 333–352. ACM Press (1991)

## Appendix

**Lemma 8 (On monotonicity).** *Let* $\langle A, \sigma \rangle \rightarrow \langle B, \rho \rangle$ *be a reduction. Then* $\rho = (\rho \ominus \sigma) \otimes \sigma$ *and* $fv(\langle B, \rho \rangle) \subseteq fv(\langle A, \sigma \rangle)$.

**Proof 3.** *Thanks to the equivalence $((a \otimes b) \ominus b) \otimes b = a \otimes b$, which holds in residuated POMs, it is easy to check that the first law is preserved by reductions, in particular by axiom* **A1** *and rule* **R2**. *The second law is proved by induction on rule application.* $\square$

**Lemma 9 (On monotonicity, II).** *Let $\langle A \parallel \exists_x^\pi B, \sigma \rangle$ be a reachable configuration. Then $\sigma = (\sigma \ominus (\exists_x \pi)_\otimes) \otimes (\exists_x \pi)_\otimes$.*

**Proof 4 (to be done better).** *We prove an equivalent property, namely, that there is $\sigma'$ such that $\sigma = \sigma' \otimes (\exists_x \pi)_\otimes$. It holds for initial configurations, and since it is clearly preserved by reductions, in particular by those obtained by axiom* **A1** *and rule* **R2**, *then we are done.* $\square$

**Lemma 10 (On labelled monotonicity).** *Let $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \rho \rangle$ be a labelled reduction. Then $\rho = (\rho \ominus (\alpha \otimes \sigma)) \otimes \alpha \otimes \sigma$ and $fv(\langle B, \rho \rangle) \subseteq fv(\langle A, \sigma \rangle) \cup sv(\alpha)$.*

**Proof 5.** *Immediate as for Lemma* **??**. $\square$

**Lemma 11 (On labelled monotonicity, II).** *Let $\langle B \parallel \exists_x^\pi C, \sigma \rangle$ be an l-reachable configuration. Then $\sigma = (\sigma \ominus (\exists_x \pi)_\otimes) \otimes (\exists_x \pi)_\otimes$.*

**Proof 6.** *As for unabelled reductions, it is easy to show that the equivalent prop-* riguardare
*erty on the existence of $\sigma'$ such that $\sigma = \sigma' \otimes (\exists_x \pi)_\otimes$ is preserved by labelled reductions.* $\square$

**Theorem 1 (Soundness).** *If $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \sigma' \rangle$ then $\langle A, \alpha \otimes \sigma \rangle \rightarrow \langle B, \sigma' \rangle$.*

**Proof 7.** *We proceed by induction and we will prove a slightly stronger proposition, namely, that the two reductions have equivalent proofs, namely, they use axioms and rules in the same order, up-to the obvious renaming (i.e.,* **LA1** *for* **A1**, *and so on).*

*The property holds for the axioms, since e.g. for* **LA2** *we know that $\alpha \leq c \ominus \sigma$ implies $\alpha \otimes \sigma \leq c$ for residuated POMs. We then proceed by induction on rule derivations, presenting only the proof for rule* **LR2**. *We have*

$$\frac{\langle A, \pi_0 \otimes \sigma \rangle \xrightarrow{\alpha} \langle B, \sigma_1 \rangle \text{ with } \pi_0 = \pi_\otimes \ominus (\exists_x \pi)_\otimes}{\langle \exists_x^\pi A, \sigma \rangle \xrightarrow{\alpha} \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho \rangle \text{ with } \rho = \sigma_1 \ominus (\alpha \otimes \pi_0 \otimes \sigma)}$$

*for $x \notin sv(\sigma) \cup sv(\alpha)$. By induction hypothesis $\langle A, \alpha \otimes \pi_0 \otimes \sigma \rangle$ is reachable and*

$$\langle A, \alpha \otimes \pi_0 \otimes \sigma \rangle \rightarrow \langle B, \sigma_1 \rangle$$

*From this it follows by* **LR2** *that* $\langle \exists_x^\pi A, \alpha \otimes \sigma \rangle$ *is reachable and*

$$\langle \exists_x^\pi A, \alpha \otimes \sigma \rangle \rightarrow \langle \exists_x^{\pi\rho} B, \alpha \otimes \sigma \otimes \exists_x \rho \rangle$$

*and we are done.* □

**Lemma 12.** *If* $\langle A, \sigma \rangle \rightarrow \langle B, \sigma' \rangle$ *then* $\langle A, \sigma \rangle \xrightarrow{\mathbf{1}} \langle B, \sigma' \rangle$.

**Proof 8.** *Only axiom* **LA2** *needs to be checked, and it is obvious, since* $\sigma \leq c$ *implies that* $\mathbf{1} \leq c \ominus \sigma$ *and thus* $\langle \mathbf{ask}(c) \mapsto A, \sigma \rangle \xrightarrow{\mathbf{1}} \langle A, \mathbf{1} \otimes \sigma \rangle$.

**Lemma 13.** *Let* $\langle A, \tau \rangle \xrightarrow{\mathbf{1}} \langle B, \tau' \rangle$ *be a reduction via the axiom* **LA1***. If* $\mathscr{C}$ *is cancellative then for every* $\sigma$ $\langle A, \sigma \rangle \xrightarrow{\mathbf{1}} \langle B, \sigma' \rangle$ *and* $\tau' \ominus \tau = \sigma' \ominus \sigma$.

**Proof 9.** *For the axiom* **LA1** *it is obvious, thanks to cancellativeness. As for the inductive step, the proof for* **LR1** *is obvious. For* **LR2***, just note that by hypothesis* $\rho = \tau_1 \ominus (\pi_0 \otimes \tau) = \sigma_1 \ominus (\pi_0 \otimes \sigma)$ *and by cancellativeness* $(\tau \otimes \exists_x \rho) \ominus \tau = (\sigma \otimes \exists_x \rho) \ominus \sigma$. □

**Lemma 14.** *Let* $\langle A, \tau \rangle \xrightarrow{\beta} \langle B, \tau' \rangle$ *be a reduction via the axiom* **LA2***. If* $\mathscr{C}$ *is localised then* $\tau' = \beta \otimes \tau$ *and for every* $\sigma$ *if* $\alpha \leq (\beta \otimes \tau) \ominus \sigma$ *then* $\langle A, \sigma \rangle \xrightarrow{\alpha} \langle B, \alpha \otimes \sigma \rangle$

**Proof 10.** *Let* $\langle \mathbf{ask}(c) \mapsto B, \tau \rangle \xrightarrow{\beta} \langle A, c \otimes \tau \rangle$*. Since* $\beta \leq c \ominus \tau$ *implies* $(\beta \otimes \tau) \ominus \sigma \leq c \ominus \sigma$*, we are done. As for the inductive step, the proof for* **LR1** *is obvious. For* **LR2***, it suffices to note that by hypothesis* $\tau_1 = \beta \otimes \tau$ *and* $\sigma_1 = \alpha \otimes \sigma$ *and by locality* $\tau_1 \ominus (\beta \otimes \pi_0 \otimes \tau) = \mathbf{1} = \sigma_1 \ominus (\alpha \otimes \pi_0 \otimes \sigma)$. □

**Proposition 2.** *Let* $\langle A, \sigma \rangle \sim \langle B, \rho \rangle$ *and* $d \in \mathscr{C}$*. If* $\mathscr{C}$ *is cancellative then* $\langle A, \sigma \otimes d \rangle \sim \langle B, \rho \otimes d \rangle$.

**Proof 11.** *We need to show that the relation* $R = \{(\langle A, \sigma \otimes d \rangle, \langle B, \sigma \otimes d \rangle) \mid \langle A, \sigma \rangle \sim \langle B, \sigma \rangle\}$ *is a labelled bisimulation. We then assume that* $\langle A, \sigma \otimes d \rangle \xrightarrow{\beta} \langle A', \sigma' \rangle$*: we need to prove that there exists* $B'$ *such that* $\langle B, \beta \otimes \sigma \otimes d \rangle \rightarrow \langle B', \sigma' \rangle$ *and* $(\langle A', \sigma' \rangle, \langle B', \sigma' \rangle) \in R$.

*By soundness* $\langle A, \beta \otimes \sigma \otimes d \rangle \rightarrow \langle A', \sigma' \rangle$ *with the same proof. We then distinguish two cases on the axiom used.*

*[$\mathbf{LA1}$] By Lemma 13 (completeness for $\mathbf{LA1}$) we have $\langle A,\sigma\rangle \xrightarrow{\mathbf{1}} \langle A',\sigma''\rangle$ and $\sigma' \ominus (\beta \otimes \sigma \otimes d) = \sigma'' \ominus \sigma$, and by Lemma 9 (monotonicity) we have that $\sigma' = \sigma'' \otimes \beta \otimes d$. Since $\langle A,\sigma\rangle \sim \langle B,\sigma\rangle$, there exists $B'$ such that $\langle B,\sigma\rangle \to \langle B',\sigma''\rangle$ and $\langle A',\sigma''\rangle \sim \langle B',\sigma''\rangle$, and it suffices to look at the cases where $\beta \otimes d \leq \mathbf{1}$. Now $\langle B,\sigma\rangle \xrightarrow{\mathbf{1}} \langle B',\sigma''\rangle$, and we may now check the proof of such reduction. If it is $\mathbf{LA1}$, we retrace the same steps as before and $\langle B,\beta \otimes \sigma \otimes d\rangle \xrightarrow{\mathbf{1}} \langle B',\sigma'\rangle$ with $\sigma' = \sigma'' \otimes \beta \otimes d$, and we are done. If it is $\mathbf{LA2}$, it suffices to note that $\beta \otimes \sigma \otimes d \leq \sigma$, hence $\langle B,\sigma \otimes \beta \otimes d\rangle \xrightarrow{\mathbf{1}} \langle B',\sigma'' \otimes \beta \otimes d\rangle$ and we are done.*

*[$\mathbf{LA2}$] By Lemma 14 (completeness for $\mathbf{LA2}$) we have $\langle A,\sigma\rangle \xrightarrow{\alpha} \langle A',\alpha \otimes \sigma\rangle$ for any $\alpha \leq (\beta \otimes \sigma \otimes d) \ominus \sigma$. Since $\langle A,\sigma\rangle \sim \langle B,\sigma\rangle$, there exists $B'$ such that $\langle B,\alpha \otimes \sigma\rangle \to \langle B',\alpha \otimes \sigma\rangle$, and by taking $\alpha = (\beta \otimes \sigma \otimes d) \ominus \sigma$ we are done.*

**Theorem 2.** *$\sim_s \subseteq \sim$. Moreover, if $\mathscr{C}$ is cancellative, then the equality holds.*

**Proof 12.** *TO DO (ma torna)*

### References

### References


[1] Aristizábal, A., Bonchi, F., Palamidessi, C., Pino, L.F., Valencia, F.D.: Deriving labels and bisimilarity for concurrent constraint programming. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 138–152. Springer (2011)

[2] Bengtson, J., Johansson, M., Parrow, J., Victor, B.: Psi-calculi: Mobile processes, nominal data, and logic. In: LICS 2009. pp. 39–48. IEEE Computer Society (2009)

[3] Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. Journal of the ACM 44(2), 201–236 (1997)

[4] Bistarelli, S., Gadducci, F.: Enhancing constraints manipulation in semiring-based formalisms. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) ECAI 2006. FAIA, vol. 141, pp. 63–67. IOS Press (2006)

[5] Bistarelli, S., Montanari, U., Rossi, F.: Soft concurrent constraint programming. ACM Transactions on Computational Logic 7(3), 563–589 (2006)

[6] Bonchi, F., Bussi, L., Gadducci, F., Santini, F.: Polyadic soft constraints. In: Alvim, M.S., Chatzikokolakis, K., Olarte, C., Valencia, F. (eds.) The Art of Modelling Computational Systems. LNCS, vol. 11760, pp. 241–257. Springer (2019)

[7] Bortolussi, L., Wiklicky, H.: A distributed and probabilistic concurrent constraint programming language. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 143–158. Springer (2005)

[8] Brim, L., Kretínský, M., Jacquet, J., Gilbert, D.R.: Modelling multi-agent systems as synchronous concurrent constraint processes. Computers and Artificial Intelligence 21(6) (2002)

[9] Buscemi, M.G., Montanari, U.: Open bisimulation for the concurrent constraint pi-calculus. In: Drossopoulou, S. (ed.) ESOP 2008. LNCS, vol. 4960, pp. 254–268. Springer (2008)

[10] Díaz, J.F., Rueda, C., Valencia, F.D.: Pi+- calculus: A calculus for concurrent processes with constraints. CLEI Electronic Journal 1(2) (1998)

[11] Gadducci, F., Santini, F.: Residuation for bipolar preferences in soft constraints. Information Processing Letters 118, 69–74 (2017)

[12] Gadducci, F., Santini, F., Pino, L.F., Valencia, F.D.: Observational and behavioural equivalences for soft concurrent constraint programming. Journal of Logic and Algebraic Methods in Programming 92, 45–63 (2017)

[13] Gilbert, D.R., Palamidessi, C.: Concurrent constraint programming with process mobility. In: Lloyd, J.W., Dahl, V., Furbach, U., Kerber, M., Lau, K., Palamidessi, C., Pereira, L.M., Sagiv, Y., Stuckey, P.J. (eds.) CL 2000. LNCS, vol. 1861, pp. 463–477. Springer (2000)

[14] Golan, J.: Semirings and Affine Equations over Them. Kluwer (2003)

[15] Guzmán, M., Haar, S., Perchy, S., Rueda, C., Valencia, F.D.: Belief, knowledge, lies and other utterances in an algebra for space and extrusion. Journal of Logic and Algebraic Methods in Programming 86(1), 107–133 (2017)

[16] Milner, R., Sangiorgi, D.: Barbed bisimulation. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 685–695. Springer (1992)

[17] Olarte, C., Rueda, C., Valencia, F.D.: Models and emerging trends of concurrent constraint programming. Constraints 18(4), 535–578 (2013)

[18] Réty, J.: Distributed concurrent constraint programming. Fundamenta Informaticae 34(3), 323–346 (1998)

[19] Sági, G.: Polyadic algebras. In: Andréka, H., Ferenczi, M., Németi, I. (eds.) Cylindric-like algebras and algebraic logic, Bolyai Society Mathematical Studies, vol. 22, pp. 367–389. Springer (2013)

[20] Saraswat, V.A., Rinard, M.C., Panangaden, P.: Semantic foundations of concurrent constraint programming. In: Wise, D.S. (ed.) POPL 1991. pp. 333–352. ACM Press (1991)