Electronic Notes in Theoretical Computer Science 67 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume67.html 20 pages

Continuous Domains and their Information System Representation as Logical Systems

Benjamín René Callejas Bedregal [1]

Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil

Ivanosca Andrade da Silva [2]

Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil

Abstract

In this work, we will introduce a finite information logic for continuous domains, as a complementary logic to continuous domain logic [BA95,Bed96,BA99,Keg99]. Here, the logical aspects of a continuous domain are analysed instead of the whole class. In order to do so we will use the continuous information systems representation of continuous domains [Hoo93,Bed96] and we relate both structures via an isomorphism. This triple (continuous domain, isomorphism and continuous information system) are called continuous logical systems. From a continuous logical system we can define several notions, such as: satisfaction relation, information content, model, theory, logical consequence, information content order, etc. Based on the constructors of continuous domains and continuous information systems, we will define the respectives constructors of continuous logical systems. Finally, we will extract a continuous domain logic from continuous logical systems and provide a soundness and completeness result for this logic based on the notion of model.

Keywords: Domain theory, continuous information systems, logical systems, models, consequence relations.

1 Introduction

Domain theory studies a class of appropriate spaces to solve recursive equations in order to obtain semantic functions to give denotational semantics of

[1] Email: bedregal@dimap.ufrn.br
[2] Email: ivanosca@dimap.ufrn.br

©2002 Published by Elsevier Science B. V.

programming languages [Sco82,Gun85]. The (algebraic) domain logic was introduced by Abramsky [Abr87,Abr91] in order to study the logical aspects (a proof system) of domains as it is used in denotational semantics for programming languages. However, algebraic domains does not support a natural and a consistent topologically interpretation of the real interval data type. Nevertheless, continuous domains have a desirable interpretation for this data type.

Analogously to the work of Scott [Sco82] to represent concretely algebraic domains, Hoofman in [Hoo93] introduced the alternative description of continuous information systems to continuous domains. Roughly speaking, a continuous information system is a triple composed by a set of tokens, a consistency predicate and an entailment relation. The tokens are finite units of information, in the sense that they are information of computations derived from observing a program execution at a finite time with a finite amount of work. Thus, the logical language associated to continuous information systems is the finite observations logic of Vickers [Vic89]. The main difference between Scott information system and continuous information systems is that, of the logical point of view, the entailment relation $\vdash$ in a Scott information system is a consequence relation [Gab94], i.e. is reflexive, monotonic and transitive. Whenever, in a continuous information system the entailment relation does not satisfy the property of reflexivity [Hoo93].

The continuous information system representation turn explicit the idea of information, in the sense that each element of a domain is seen as a collection of information that it "satisfies". Although the domains and information systems are categorically equivalent, information systems allow us to capture the logical aspects of domains, in the sense that the properties of domains can be derived from the entailment between propositions about the properties of computations. Thus, for example, the continuous domain of real interval is represented as a continuous information system whose basic information are rational intervals.

In this work, we will see continuous domains and their continuous information system representation as logical systems which allows us to explore the logical relations between elements of a continuous domain with the informations languages obtained from their continuous information system. From this logical systems, we can define several notions, such as: satisfaction relation, information content, model, theory, logical consequence, information content order, etc. This result in a complementary logic to the continuous domain logic [BA95,Bed96,BA99,Keg99], where the logical aspects of a continuous domain are analysed instead of the whole class at all. Based on the continuous domains and continuous information systems constructors, we will define the respectives constructors on continuous logical systems. Finally, we will extract a continuous domain logic from continuous logical systems and provide a soundness and completeness result for this logic based on the notion of model.

2 Continuous Domains

Since the seminal work of Scott and Strachey, several mathematical structures have been widely employed as models for denotational semantics of programming languages. Continuous domains, which are, basically, a continuous lattice [Sco72] without, possibly, the top element is one of them. Continuous domain allows us to develop a theory of interval arithmetic and numerical analysis [Aci91]. This theory has the advantage of being constructive and computational, besides of unifying the theory of programming languages semantics and computational mathematics. In what follows is formally define this.

Let $\mathbf{D} = \langle D, \leq \rangle$ be a partially ordered set (poset). A non empty set $\Delta \subseteq D$ is called **directed** if each finite subset has an upper bound, or equivalently, $\forall a, b \in \Delta \quad \exists c \in \Delta$ such that $a \leq c$ and $b \leq c$. A poset $\mathbf{D}$ is **complete** (for short cpo) if each directed set Δ has a supremum or least upper bound (denoted by $\bigsqcup \Delta$) and has a bottom element, usually denoted by $\bot$. We say that a is **way below** b (denoted by $a \ll b$) if for every directed set Δ such that $b \leq \bigsqcup \Delta$ then $a \leq x$ for some $x \in \Delta$.

Let $\dagger x$ be the set $\{y \in D \ / \ y \ll x\}$. A cpo $\mathbf{D}$ is called **continuous** if, for all $x \in D$, the set $\dagger x$ is directed and $x = \bigsqcup \dagger x$. A subset $B \subseteq D$ is a **basis** of $\mathbf{D}$ if for each $x \in D$, the set $\dagger_B x = \{y \in B \ / \ y \ll x\}$ is directed and $x = \bigsqcup \dagger_B x$. A continuous cpo $\mathbf{D}$ where each consistent set (a set has an upper bound in $\mathbf{D}$) has a supremum in D is called **continuous domain**.

An element x of a continuous domain $\mathbf{D} = \langle D, \leq, \bot \rangle$ is said to be **total** if $x \leq y$ implies $x = y$. We denote the set of total elements of a domain $\mathbf{D}$ by $Tot(\mathbf{D})$.

Example 2.1 Let $\mathbb{I}(\mathbb{R}) = \{[r, s] \ / \ r, s \in \mathbb{R} \text{ and } r \leq s\} \cup \{[-\infty, +\infty]\}$ be the set of closed real intervals with real numbers as end points or simply real intervals. Define on $\mathbb{I}(\mathbb{R})$ the follow partial order:

$$[r, s] \sqsubseteq [t, u] \text{ iff } [r, s] = [-\infty, +\infty] \text{ or } (r \leq_{\mathbb{R}} t \text{ and } u \leq_{\mathbb{R}} s).$$

The pair $\Re = \langle \mathbb{I}(\mathbb{R}), \sqsubseteq, [-\infty, +\infty] \rangle$ is a continuous domain. The way below relation associated to $\Re$ is defined by

$$[r, s] \ll [t, u] \text{ iff } [r, s] = [-\infty, +\infty] \text{ or } (r <_{\mathbb{R}} t \text{ and } u <_{\mathbb{R}} s).$$

A basis for it continuous domain is the set $\mathbb{I}(\mathbb{Q}) = \{[p, q] \ / \ p, q \in \mathbb{Q} \text{ and } p <_{\mathbb{Q}} q\}$ of rational intervals, i.e. intervals whose end points are rational numbers. The total elements of $\Re$ are the degenerate intervals, i.e. intervals with the same end points and therefore can be consider as real numbers. Thus,

$$Tot(\Re) = \{[r, r] \ / \ r \in \mathbb{R}\} \cong \mathbb{R}.$$

The morphisms between continuous domains, called continuous functions, are the monotonic functions w.r.t. the order associated to domains preserving supremum of directed sets. The continuous domains $\mathbf{D} = \langle D, \sqsubseteq_D, \bot_D \rangle$ and $\mathbf{E} = \langle E, \sqsubseteq_E, \bot_E \rangle$ are isomorphic orders, denoted by $\mathbf{D} \cong \mathbf{E}$, if there exists a

bijective continuous function $f : D \longrightarrow E$.

2.1 Constructors on continuous domains

For a mathematical structure to be considered a semantic domain it must be closed under the desired operators. Usually, the "desired" operators are the product, function space, lifting, disjoint and coalesced sum (sometimes are also considered the powerdomains constructors). We will define some of this domain contructors to continuous domains.

Lifting

Definition 2.2 Let $\mathbf{D} = \langle D, \leq, \bot \rangle$ be a continuous domain. The **lifting** of $\mathbf{D}$, denoted by $\mathbf{D}_\uparrow$, is the poset $\mathbf{D}_\uparrow = \langle D_\uparrow, \leq_\uparrow, \bot_\uparrow \rangle$ where

- $D_\uparrow = (D \times \{0\}) \cup \{\bot_\uparrow\}$
- $(x, 0) \leq_\uparrow (y, 0)$ iff $x \leq y$
- $\bot_\uparrow \leq_\uparrow x$ for each $x \in D_\uparrow$.

Coalesced sum

Definition 2.3 Let $\mathbf{D}_1 = \langle D_1, \leq_1, \bot_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2, \bot_2 \rangle$ be continuous domains. The **coalesced sum** of $\mathbf{D}_1$ with $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \oplus \mathbf{D}_2$, is the poset $\mathbf{D}_1 \oplus \mathbf{D}_2 = \langle D_1 \oplus D_2, \leq, \bot \rangle$ where

- $D_1 \oplus D_2 = ((D_1 \uplus D_2) \cup \{\bot\}) - \{(\bot_1, 1), (\bot_2, 2)\}$ [3],
- $(x, i) \leq (y, j)$ iff $i = j$ and $x \leq_i y$,
- $\bot \leq x$ for each $x \in D_1 \oplus D_2$.

Cartesian product

Definition 2.4 Let $\mathbf{D}_1 = \langle D_1, \leq_1, \bot_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2, \bot_2 \rangle$ be continuous domains. The **cartesian product** of $\mathbf{D}_1$ with $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \times \mathbf{D}_2$, is the poset $\mathbf{D}_1 \times \mathbf{D}_2 = \langle D_1 \times D_2, \leq, (\bot_1, \bot_2) \rangle$ where

$$(x_1, x_2) \leq (y_1, y_2) \text{ iff } x_1 \leq_1 y_1 \text{ and } x_2 \leq_2 y_2.$$

Function space

Definition 2.5 Let $\mathbf{D}_1 = \langle D_1, \leq_1, \bot_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2, \bot_2 \rangle$ be continuous domains. The **function space** from $\mathbf{D}_1$ into $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \to \mathbf{D}_2$, is the poset $\mathbf{D}_1 \to \mathbf{D}_2 = \langle D_1 \to D_2, \leq, f_\bot \rangle$ where

(i) $D_1 \to D_2 = \{f : D_1 \longrightarrow D_2 \ / f \text{ is a continuous function}\}$,

(ii) $f \leq g$ iff for each $x \in D_1$, $f(x) \leq_2 g(x)$,

[3] $A \uplus B = (A \times \{1\}) \cup (B \times \{2\})$

(iii) $f_{\perp}(x) = \perp_2$ for each $x \in D_1$.

Proposition 2.6 *[Aci91,Bed96] Let* $\mathbf{D}$, $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *be continuous domains. Then the posets* $\mathbf{D}_{\uparrow}$, $\mathbf{D}_1 \oplus \mathbf{D}_2$, $\mathbf{D}_1 \times \mathbf{D}_2$ *and* $\mathbf{D}_1 \to \mathbf{D}_2$ *are too continuous domains.*

3 Continuous Information Systems

There exist several categories of semantic domains and the most of them have an information systems representation [Bed99,Bed01] following the spirit of the information systems introduced by D. Scott [Sco82]. The information systems are a more concrete and logic representation for Scott domains and can be viewed as prescriptions or programs saying how to build a domain. Structures consideres information systems representation of some domain category must have the follows components:

Definition 3.1 A triple $\mathcal{I} = \langle I, Con, \vdash \rangle$, where I is a non empty countable **information set**, Con is a non empty subset of $\mathcal{P}^{fin}(I)$ (finite parts of I) called **consistency predicate** and $\vdash$ is a subset of $Con \times I$ named **entailment relation**, is said an **information system structure**.

In the case of continuous domains the information system structure must satisfy the follows conditions:

Definition 3.2 [Hoo93] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be an information system structure. $\mathcal{I}$ is a **continuous information system**, CIS in short, if

1. If $X \in Con$ and $Y \subseteq X$ then $Y \in Con$
2. If $a \in I$ then, $\{a\} \in Con$
3. If $X \vdash a$ then, $X \cup \{a\} \in Con$
4. If $Y \subseteq X$, $X \in Con$ and $Y \vdash a$ then, $X \vdash a$
5. If $X \vdash Y$ and $Y \vdash a$ then, $X \vdash a$
6. If $X \vdash a$ then $\exists Y \in Con$ such that $X \vdash Y$ and $Y \vdash a$

The notation $X \vdash Y$ is a shorthand for $\forall a \in Y$, $X \vdash a$.

Example 3.3 We can to see each rational interval $[p, q]$ as a partial or finite information of some real number r belong to it. Thus, finite set of information (rational intervals) are consistent if all the intervals, inform on a same real number, i.e.

$$X \in Con_{\mathbb{I}(\mathbb{Q})} \text{ iff } X \subseteq^{fin} \mathbb{I}(\mathbb{Q}) \text{ and } Max(left(X)) < Min(right(X)),$$

where $left(X) = \{p \in \mathbb{Q} \ / \ [p, q] \in X \text{ for some } q \in \mathbb{Q}\}$, $right(X) = \{q \in \mathbb{Q} \ / \ [p, q] \in X \text{ for some } p \in \mathbb{Q}\}$, $Max(S)$ is the greater element of S w.r.t. the usual order, and $Min(S)$ is the lesser element of S.

An entailment relation can be defined as follow:

$$X \vdash_{\mathbb{I}(\mathbb{Q})} [p,q] \text{ iff } p <_{\mathbb{Q}} Max(left(X)) \text{ and } Min(right(X)) < q.$$

The structure $\mathcal{I}_{\Re} = \langle \mathbb{I}(\mathbb{Q}), Con_{\mathbb{I}(\mathbb{Q})}, \vdash_{\mathbb{I}(\mathbb{Q})} \rangle$ is a continuous information system.

A CIS informs about elements of a domain. These elements are identified with the set of information, in the system, which are satisfied by them.

Definition 3.4 [Hoo93] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be a CIS. The set $x \subseteq I$ is an **element** of $\mathcal{I}$ if the the following conditions are satisfied:

(i) If $X \subseteq^{fin} x$, then $X \in Con$;[4] .

(ii) If $X \subseteq x$ and $X \vdash a$, then $a \in x$;

(iii) If $a \in x$, then $\exists Y \subseteq^{fin} x$ such that $Y \vdash a$.

Let $|\mathcal{I}|$ be the set of elements of a CIS $\mathcal{I} = \langle I, Con, \vdash \rangle$. The poset $\langle |\mathcal{I}|, \subseteq \rangle$ is called **domain of elements** of the CIS $\mathcal{I}$.

Lemma 3.5 *[Hoo93] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be a CIS. If $X \in Con$ then $\overline{X} = \{a \in I \ / \ X \vdash a\} \in |\mathcal{I}|$.*

Theorem 3.6 *[Hoo93] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be a CIS. Then $\langle |\mathcal{I}|, \subseteq, \overline{\emptyset} \rangle$ is a continuous domain, with $\overline{Con} = \{\overline{X} \ / \ X \in Con\}$ as a countable basis*

Example 3.7 The domain of elements of the CIS $\mathcal{I}_{\Re}$ is an isomorphic order to the continuous domain $\Re$. The isomorphisms is given by the function $f : \mathbb{I}(\mathbb{R}) \longrightarrow |\mathcal{I}_{\Re}|$ defined by:

$$f([r,s]) = \{[p,q] \in \mathbb{I}(\mathbb{Q}) \ / \ [p,q] \ll [r,s]\}.$$

There are technical advantages to working with CIS rather than directly with continuous domains. First CIS uses the set theory languages and second the properties of domains can be derived rather than postulated.

3.1 Constructors on continuous information systems

The usual constructors on domains have their analogous on CIS's:

Lifting

Definition 3.8 Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be a CIS. The **lifting** of $\mathcal{I}$ is the information system structure $\mathcal{I}_{\uparrow} = \langle I_{\uparrow}, Con_{\uparrow}, \vdash_{\uparrow} \rangle$, where

(i) $I_{\uparrow} = (I \times \{0\}) \cup \{\bot\}$

(ii) $X \in Con_{\uparrow}$ iff $\pi(X) = \{a \in I \ / \ (a,0) \in X\} \in Con$

(iii) $X \vdash_{\uparrow} a$ iff $a = \bot$ or $\pi(X) \vdash \pi(\{a\})$

[4] $X \subseteq^{fin} Y$ is an abbreviation for "$X \subseteq Y$ and X is finite"

Coalesced sum

Definition 3.9 Let $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ and $\mathcal{I}_2 = \langle I_2, Con_2, \vdash_2 \rangle$ be CIS's. The **coalesced sum** of $\mathcal{I}_1$ and $\mathcal{I}_2$ is the information system structure $\mathcal{I}_1 \oplus \mathcal{I}_2 = \langle I_1 \oplus I_2, Con_\oplus, \vdash_\oplus \rangle$, where

(i) $I_1 \oplus I_2 = (I_1 - \overline{\emptyset}) \uplus (I_2 - \overline{\emptyset})$

(ii) $X \in Con_\oplus$ iff there exists $X_1 \in Con_1$ such that $X = X_1 \times \{1\}$ or there exists $X_2 \in Con_2$ such that $X = X_2 \times \{2\}$

(iii) $X \vdash_\oplus (a, i)$ iff $\pi(X) = \{b \ / \ (b, i) \in X\} \vdash_i a$

Cartesian product

The natural idea is that the cartesian product between CIS's has as basic information set the cartesian product of their respectives basic information sets. Nevertheless, since the least element of a CIS can be the empty set and $\emptyset \times S = \emptyset$ for each set S, we need add a new information to the basic set of information.

Definition 3.10 Let $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ and $\mathcal{I}_2 = \langle I_2, Con_2, \vdash_2 \rangle$ be continuous information systems. The **cartesian product** of $\mathcal{I}_1$ and $\mathcal{I}_2$ is the information system structure $\mathcal{I}_1 \times \mathcal{I}_2 = \langle I_\times, Con_\times, \vdash_\times \rangle$, where

(i) $I_\times = ((I_1 \times \{1\}) \cup \{\bot_1\}) \times ((I_2 \times \{2\}) \cup \{\bot_2\})$

(ii) $X \in Con_\times$ iff $\pi_1(X) = \{a \ / \ ((a,1),b) \in X$ for some $b \in (I_2 \times \{2\}) \cup \{\bot_2\}\}$ in Con_1 and $\pi_2(X) = \{b \ / \ (a,(b,1)) \in X$ for some $a \in (I_1 \times \{1\}) \cup \{\bot_1\}\}$ in Con_2.

(iii) $X \vdash_\times a$ iff $\pi_1(X) \vdash_1 \pi_1(\{a\})$ and $\pi_2(X) \vdash_2 \pi_2(\{b\})$

Function space

Definition 3.11 Let $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ and $\mathcal{I}_2 = \langle I_2, Con_2, \vdash_2 \rangle$ be CIS's. The **function space** from $\mathcal{I}_1$ into $\mathcal{I}_2$, is the information system structure $\mathcal{I}_1 \to \mathcal{I}_2 = \langle Con_1 \times I_2, Con_\to, \vdash_\to \rangle$, where

(i) Let $X = \{(X_1, a_1), \ldots, (X_n, a_n)\} \subseteq^{fin} Con_1 \times I_2$. $X \in Con_\to$ iff $\forall F \subseteq \{1, \ldots, n\}$, $(\bigcup_{i \in F} X_i \in Con_1 \Rightarrow \{a_i \ / \ i \in F\} \in Con_2)$

(ii) $\{(X_1, a_1), \ldots, (X_n, a_n)\} \vdash_\to (X, a)$ iff $\{a_i \ / \ X \vdash_1 X_i\} \vdash_2 a$

Proposition 3.12 *[Bed96] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$, $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ and $\mathcal{I}_2 = \langle I_2, Con_2, \vdash_2 \rangle$ be CIS's. Then $\mathcal{I}_\uparrow$, $\mathcal{I}_1 \oplus \mathcal{I}_2$, $\mathcal{I}_1 \times \mathcal{I}_2$ and $\mathcal{I}_1 \to \mathcal{I}_2$ are CIS's.*

Theorem 3.13 *[Bed96] Let $\mathcal{I} = \langle I, Con, \vdash \rangle$, $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ and $\mathcal{I}_1 = \langle I_1, Con_1, \vdash_1 \rangle$ be CIS's such that $\langle | \mathcal{I} |, \subseteq, \overline{\emptyset} \rangle \cong \mathbf{D}$, $\langle | \mathcal{I}_1 |, \subseteq, \overline{\emptyset} \rangle \cong \mathbf{D}_1$ and $\langle | \mathcal{I}_2 |, \subseteq, \overline{\emptyset} \rangle \cong \mathbf{D}_2$. Then $\langle | \mathcal{I}_\uparrow |, \subseteq, \{\bot\} \rangle \cong \mathbf{D}_\uparrow$, $\langle | \mathcal{I}_1 \oplus \mathcal{I}_2 |, \subseteq, \emptyset \rangle \cong \mathbf{D}_1 \times \mathbf{D}_2$, $\langle | \mathcal{I}_1 \times \mathcal{I}_2 |, \subseteq, \overline{\{(\bot_1, \bot_2)\}} \rangle \cong \mathbf{D}_1 \times \mathbf{D}_2$ and $\langle | \mathcal{I}_1 \to \mathcal{I}_2 |, \subseteq, \overline{\emptyset} \rangle \cong \mathbf{D}_1 \to \mathbf{D}_2$.*

4 Language of Finite Information

An information is said to be finite, if on the hypothesis of its correctness, we can verify it in a finite amount time and work. For example, the information "there exists a blue crow", seemingly false, to verify its truth, all we need is to observe the suppose crow and determinate its blueness, which will be made in a finite amount of time. But, the information "all the crows are black", seemingly true, can not be confirmed in an finite amount of time, because we need to observe all the crows (in the past, present and future). In our examples, because always is possible to determinate in a finite time and with a finite amount of work when a real number belong or not to a rational interval, we can consider each rational interval is a finite information of some real number. The logical operators that preserve the finiteness of the information are the conjunction (finite) and disjunction (perhaps infinite). We, also consider two canonical information: the always true information (T) and the always false information (F).

Definition 4.1 Let $\mathcal{I} = \langle I, Con, \vdash \rangle$ be a continuous information system. The finite information language of $\mathcal{I}$, denoted by $\mathcal{L}(I)$, is defined recursively as follows:

(i) **The atomic information:** if $a \in I$, then $a \in \mathcal{L}(I)$;

(ii) if $\alpha, \beta \in \mathcal{L}(I)$, then $\alpha \vee \beta \in \mathcal{L}(I)$ and $\alpha \wedge \beta \in \mathcal{L}(I)$.

(iii) if $\Gamma \subseteq \mathcal{L}(I)$, then $\bigvee_{\alpha \in \Gamma} \alpha \in \mathcal{L}(I)$.

We will be use, the symbols T and F as abbreviations of the sentences $\bigvee_{a \in I} a$ and $\bigvee_{a \in \emptyset} a$, respectively. Thus $\mathsf{T}, \mathsf{F} \in \mathcal{L}(I)$.

5 Logical System of a Continuous Domain

Let $\mathbf{D} = \langle D, \leq, \perp \rangle$ be a continuous domain. A CIS $\mathcal{I} = \langle I, Con, \vdash \rangle$ is said a **representation of D** if $\mathbf{D}$ and $\langle |\mathcal{I}|, \subseteq, \overline{\emptyset} \rangle$ are isomorphic orders. If $\varphi : D \longrightarrow |\mathcal{I}|$ is such an isomorphism, then the triple $\mathcal{D} = \langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ is said a **continuous logical system** (CLS for short). The sentences in $\mathcal{L}(I)$ can be see as information on the elements of $\mathbf{D}$ and conversely, we can to see the elements of $\mathbf{D}$ as models of the information in $\mathcal{L}(I)$. Thus, a CLS determine a satisfaction relation "$\models$" between the elements of the continuous domain and the information of the CIS. The relation $\models \subseteq D \times \mathcal{L}(I)$ is defined recursively as follows:

1. $x \models \mathsf{T}$
2. $\forall a \in I \;\; x \models a$ iff $a \in \varphi(x)$
3. $x \models \alpha \wedge \beta$ iff $x \models \alpha$ and $x \models \beta$.
4. $x \models \alpha \vee \beta$ iff $x \models \alpha$ or $x \models \beta$
5. $x \models \bigvee \Gamma$ iff $x \models \alpha$ for some $\alpha \in \Gamma$

A **semantical system** is a structure $\langle A, Mod, \models \rangle$, where A is a set, Mod is a class of models (i.e. a set of objects) and $\models$ is a satisfaction relation

between sentence and models, i.e. $\models \subseteq Mod \times A$ [Par83,Car90]. The triple $\mathcal{D} = \langle \mathcal{L}(I), D, \models \rangle$ is the semantical system associated to CLS $\mathcal{D}$.

Example 5.1 The function $\varphi : \mathbb{I}(\mathbb{R}) \longrightarrow |\mathcal{I}_{\Re}|$, defined by

$$\varphi([r,s]) = \{[p,q] \in \mathbb{I}(\mathbb{Q}) \ / \ [p,q] \ll [r,s]\},$$

is an isomorphism from $\Re$ into $|\mathcal{I}_{\Re}|$. Since the continuous domain $\Re$ and $|\mathcal{I}_{\Re}|$ are isomorphic orders, it have associated a relation $\models \subseteq \mathbb{I}(\mathbb{R}) \times \mathcal{L}(\mathbb{I}(\mathbb{Q}))$ which is defined as above. Notice that in the case of the atomic elements of $\mathcal{L}(\mathbb{I}(\mathbb{Q}))$, this relations has the follow behaviour

$$[r,s] \models [p,q] \text{ iff } [p,q] \ll [r,s].$$

The system $\langle \Re, \varphi, \mathcal{I}_{\Re} \rangle$ is a continuous logical system and $\langle \mathcal{L}(\mathbb{I}(\mathbb{Q})), \Re, \models \rangle$ is their respective semantical system.

According to Shannon [SW49], information reduces the uncertainty, which is relative with the contents of the information. However, Shannon information theory, is determined by a probabilistic measure [SW49]. While for us "reduction of uncertainty" is done by the reduction of the spectrum of elements. So, the **contents of an information** in a CLS is associated with the set of their non models [Car90], i.e.

$$Cont(\alpha) = D - \{x \in D \ / \ x \models \alpha\}$$

Thus, for example, the contents of an information $[p,q] \in \mathbb{I}(\mathbb{Q})$ is the set

$$\begin{aligned} Cont([p,q]) &= \mathbb{I}(\mathbb{R}) - \{[r,s] \in \mathbb{I}(\mathbb{R}) \ / \ [r,s] \models [p,q]\} \\ &= \{[r,s] \in \mathbb{I}(\mathbb{R}) \ / \ s \leq p \text{ or } q \leq r\} \end{aligned}$$

The following proposition establishes a calculus of information contents to CLS.

Proposition 5.2 *For all CLS* $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$*, if* $\alpha, \beta \in \mathcal{L}(I)$ *and* $\Gamma \subseteq \mathcal{L}(I)$*, then, the following are valid:*

1. $Cont(\alpha \wedge \beta) = Cont(\alpha) \cup Cont(\beta)$
2. $Cont(\alpha \vee \beta) = Cont(\alpha) \cap Cont(\beta)$
3. $Cont(\bigvee \Gamma) = \bigcap_{\alpha \in \Gamma} Cont(\alpha)$
4. $Cont(\mathsf{T}) = \emptyset$
5. $Cont(\mathsf{F}) = D$

The property 4 says that a true information for all element has an empty content in the system, in other words these information is *irrelevant*. Analogously, the property 5 states that a false information has a greatest content, or in other words is *absurd*.

The dual notion of information content is the notion of **models of an information**. It is the set of objects in the domain which satisfies the information, and can be seen as the function $Mod : \mathcal{L}(I) \longrightarrow \mathcal{P}(D)$ defined by

$$Mod(\alpha) = \{x \in D \ / \ x \models \alpha\}.$$

Notice that,

(i) $Mod(\alpha) \cup Cont(\alpha) = D$,

(ii) $Mod(\alpha) \cap Cont(\alpha) = \emptyset$,

(iii) $Mod(\alpha \wedge \beta) = Mod(\alpha) \cap Mod(\beta)$ and

(iv) $Mod(\alpha \vee \beta) = Mod(\alpha) \cup Mod(\beta)$.

We can extend the consistency predicate of a CIS by consider $\mathcal{L}(I)$ instead of I. A subset $\Gamma \subseteq^{fin} \mathcal{L}(I)$ is **consistent** if $Mod(\Gamma) \neq \emptyset$, (i.e. if there exists an element in D satisfying all the information in Γ). Notice, that this notion of consistency is coherent with the relation Con in $\mathcal{I}$. For example, if $X \in Con$ then, by lemma 3.5, $\overline{X} \in |\mathcal{I}|$, and therefore $\varphi^{-1}(\overline{X}) \models X$, where φ^{-1} is the inverse of the isomorphism (bijection) $\varphi : D \longrightarrow |\mathcal{I}|$. So, $Mod(X) \neq \emptyset$ and consequently, X is consistent.

The inverse notion of models determines all the information in the system about the objects of a domain, it is called **theory of the object** in the CLS, which can be see as their specification in the CLS. Thus, the theory of the objects in a CLS, is the function $Theo : D \longrightarrow \mathcal{L}(I)$ defined by

$$Theo(x) = \{\alpha \in \mathcal{L}(I) \ / \ x \models \alpha\}.$$

Thus, for example, the theory of a real interval $[r, s]$ in the logical system of our example contain the set of rational intervals which inform to it, i.e.

$$\{[p, q] \in \mathbb{I}(\mathbb{Q})) \ / \ [p, q] \ll [r, s]\} \subseteq Theo([r, s]).$$

The following proposition states that two diferent objects of a domain satisfies different sets of informations in a CLS.

Proposition 5.3 *Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS and let $x, y \in D$. If $x \neq y$ then $Theo(x) \neq Theo(y)$. If $x \leq y$ then $Theo(x) \subseteq Theo(y)$.*

This functions can be extended to a set of informations (Γ) or objects (X), as follows:

- $Cont(\Gamma) = \bigcup_{\alpha \in \Gamma} Cont(\alpha)$,
- $Mod(\Gamma) = \bigcap_{\alpha \in \Gamma} Mod(\alpha)$ and
- $Theo(X) = \bigcap_{x \in X} Theo(x)$.

Proposition 5.4 *Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS. If $\Gamma, \Delta \subseteq \mathcal{L}(I)$ and $X, Y \subseteq D$, then*

1. $Mod(\emptyset) = D$
2. $Mod(\Gamma \cup \Delta) = Mod(\Gamma) \cap Mod(\Delta)$
3. $Theo(\emptyset) = \mathcal{L}(I)$
4. $Theo(X \cup Y) = Theo(X) \cap Theo(Y)$

Proposition 5.5 *[Par83] Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS. Mod and Theo are a Galois collection. I.e. if $\Gamma, \Delta \subseteq \mathcal{L}(I)$ and $X, Y \subseteq D$, then*

(i) *If $\Gamma \subseteq \Delta$ then $Mod(\Delta) \subseteq Mod(\Gamma)$*

(ii) *If $X \subseteq Y$ then $Theo(Y) \subseteq Theo(X)$*

(iii) *$\Gamma \subseteq Theo(Mod(\Gamma))$ and $X \subseteq Mod(Theo(X))$.*

Let $\langle \mathbf{D}, \models, \mathcal{I} \rangle$ be a CLS and $\Gamma \subseteq \mathcal{L}(I)$. Γ is **complete** iff Γ has an unique model, i.e. $Mod(\Gamma) = \{x\}$ for some $x \in D$. Γ is **complete greatest** iff it becomes inconsistent on the addition of any new information to Γ, i.e. to each $\alpha \in \mathcal{L}(I)$, if $\alpha \notin \Gamma$ then $Mod(\Gamma \cup \{\alpha\}) = D$. Γ is **complete minimally** iff Γ is complete and there does not exist any proper subset $\Delta \subset \Gamma$ such that $Mod(\Delta) = Mod(\Gamma) = \{x\}$. So, Γ is the information set which are sufficient and necessary to specify x.

Proposition 5.6 *Let $\mathcal{D} = \langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS. Then, for each $x \in D$, $x \in Tot(\mathbf{D})$ iff the set $\varphi(x)$ is complete.*

The idea to characterise the notion of complete information is to emphasize that continuous information systems allows us to specify the total element of continuous domains in terms of finite information.

An important concept, which arises when we see a CIS as a logical theory, is logical consequence for information.

Definition 5.7 Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS. The function $CnI : \mathcal{L}(I) \longrightarrow \mathcal{P}(\mathcal{L}(I))$ defined by $CnI(\alpha) = Theo(Mod(\alpha))$, determines the information which can be deduced from α, i.e. their **logical consequences**.

Proposition 5.8 *Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS and $\Gamma, \Delta \subseteq \mathcal{L}(I)$. Then, $CnI(\Gamma \cup \Delta) = CnI(\Gamma) \cup CnI(\Delta)$.* [5]

The notion of logical consequence extends the entailment relation in a CIS $\mathcal{I} = \langle I, Con, \vdash \rangle$ to $\mathrel{|\!\sim}$, which consider $\mathcal{L}(I)$ instead of I. Thus, given $\Gamma \subseteq \mathcal{L}(I)$ and $\alpha \in \mathcal{L}(I)$, $\Gamma \mathrel{|\!\sim} \alpha$ if $\alpha \in CnI(\Gamma)$. If $\alpha \in CnI(\emptyset)$, then α is a **theorem**, denoted by $\mathrel{|\!\sim} \alpha$.

Proposition 5.9 *Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS. Then*

(i) *if $X \vdash a$, then $X \mathrel{|\!\sim} a$;*

(ii) *if $\alpha \in \Gamma$ then $\Gamma \mathrel{|\!\sim} \alpha$;*

(iii) *if $\Gamma \mathrel{|\!\sim} \alpha$, for each $\alpha \in \Delta$, and $\Delta \mathrel{|\!\sim} \beta$ then $\Gamma \mathrel{|\!\sim} \beta$;*

(iv) *if $\Gamma \subseteq \Delta$ and $\Gamma \mathrel{|\!\sim} \alpha$ then $\Delta \mathrel{|\!\sim} \alpha$;*

(v) *if $\Gamma \mathrel{|\!\sim} \alpha$ then $\Gamma \cup \{\alpha\}$ is consistent;*

(vi) *if Γ is not consistent then $\Gamma \mathrel{|\!\sim} \alpha$ for each $\alpha \in \mathcal{L}(I)$;*

(vii) *if $\Gamma \mathrel{|\!\sim} \alpha$ then $Mod(\Gamma) \subseteq Mod(\alpha)$.*

(viii) *if $\Gamma \mathrel{|\!\sim} \alpha$ then there exists $\Gamma_0 \subseteq^{fin} \Gamma$ such that $\Gamma_0 \mathrel{|\!\sim} \alpha$*

(ix) *if $\Gamma, \alpha, \beta \mathrel{|\!\sim} \gamma$ then $\Gamma, \alpha \wedge \beta \mathrel{|\!\sim} \gamma$*

The properties (ii), (iii) and (iv) of $\mathrel{|\!\sim}$ are knows as reflexivity, transitivity and monotonicity, respectively. So, $\mathrel{|\!\sim}$ is a **consequence relation** in $\mathcal{L}(I)$ [Gab94].

[5] The expression $CnI(A)$ is an abreviation of $Theo(Mod(A))$.

From the definition of information content we establish some interesting relations on information.

Definition 5.10 Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS and $\alpha, \beta \in \mathcal{L}(I)$. The information α is **greatest** than or **more exact** than the information β, denoted by $\beta \sqsubseteq \alpha$ iff $Cont(\beta) \subseteq Cont(\alpha)$. Because when $\beta \sqsubseteq \alpha$, all model of α is a model of β we also said that β is a **semantical consequence** of α. Analogously, α and β are **equivalent information**, denoted by $\alpha \equiv \beta$, iff $Cont(\alpha) = Cont(\beta)$.

Example 5.11 The follow information are equivalent w.r.t. CLS $\langle \Re, \varphi, \mathcal{I}_{\Re} \rangle$:

(i) if $p_1 \leq p_2$, $p_2 \leq q_1$ and $q_1 \leq q_2$ then $[p_1, q_2] \equiv [p_1, q_1] \vee [p_2, q_2]$

(ii) if $p_1 \leq p_2$, $p_2 \leq q_1$ and $q_1 \leq q_2$ then $[p_2, q_1] \equiv [p_1, q_1] \wedge [p_2, q_2]$

(iii) $[-\infty, +\infty] \equiv \mathsf{T}$

The *information content order* $\sqsubseteq$ is an intensive order. On the other hand, the dual order, $\sqsupseteq$, models-based, is an extensional information content order in the sense of [Bun74].

Proposition 5.12 *Let $\langle \mathbf{D}, \varphi, \mathcal{I} \rangle$ be a CLS, $\alpha, \beta \in \mathcal{L}(I)$ and $\Gamma \subseteq \mathcal{L}(I)$. Then*

1. $\alpha \vee \beta \equiv \beta \vee \alpha$ *and* $\alpha \wedge \beta \equiv \beta \wedge \alpha$
2. $\alpha \wedge \bigvee_{\beta \in \Gamma} \beta \equiv \bigvee_{\beta \in \Gamma} (\alpha \wedge \beta)$
3. $\alpha \equiv \beta$ *iff* $\alpha \sqsubseteq \beta$ *and* $\beta \sqsubseteq \alpha$
4. $\alpha \sqsubseteq \alpha \wedge \beta$
5. $\alpha \vee \beta \sqsubseteq \alpha$
6. $\mathsf{T} \sqsubseteq \alpha$ *and* $\alpha \sqsubseteq \mathsf{F}$
7. *if* $\alpha \mid\sim \beta$ *then* $\beta \sqsubseteq \alpha$

5.1 Constructors on Continuous Logical Systems

Based on the constructors on continuous domains and their respectives constructors on continuous information systems, we can naturally define constructors on CLS. I.e. if $\mathcal{D} = \langle \mathbf{D}, \varphi, \mathcal{I} \rangle$, $\mathcal{D}_1 = \langle \mathbf{D}_1, \varphi_1, \mathcal{I}_1 \rangle$ and $\mathcal{D}_2 = \langle \mathbf{D}_2, \varphi_2, \mathcal{I}_2 \rangle$ are CLS, then we define

The lifting constructor by

$\mathcal{D}_\uparrow = \langle \mathbf{D}_\uparrow, \varphi_\uparrow, \mathcal{I}_\uparrow \rangle$, where

$$\varphi_\uparrow((x, 0)) = (\varphi(x) \times \{0\}) \cup \{\bot\} \text{ and } \varphi_\uparrow(\bot_\uparrow) = \{\bot\}$$

The coalesced sum constructor by

$\mathcal{D}_1 \oplus \mathcal{D}_2 = \langle \mathbf{D}_1 \oplus \mathbf{D}_2, \varphi, \mathcal{I}_1 \oplus \mathcal{I}_2 \rangle$, where

$$\varphi((x, i)) = (\varphi_i(x) - \overline{\emptyset}) \times \{i\} \text{ and } \varphi(\bot) = \overline{\emptyset},$$

where the first $\overline{\emptyset}$ belong to $|\mathcal{I}_i|$ and the second belong to $|\mathcal{I}_1 \oplus \mathcal{I}_2|$.

The cartesian product constructor by

$\mathcal{D}_1 \times \mathcal{D}_2 = \langle \mathbf{D}_1 \times \mathbf{D}_2, \varphi, \mathcal{I}_1 \times \mathcal{I}_2 \rangle$, where

$\varphi(x,y) = \{(u,v) \ / \ (u = \bot_1 \text{ or } \pi(u) \in \varphi_1(x)) \text{ and } (v = \bot_2 \text{ or } \pi(v) \in \varphi_2(y))\}$

where $\pi(a,i) = a$.

The function space constructor by

$\mathcal{D}_1 \to \mathcal{D}_2 = \langle \mathbf{D}_1 \to \mathbf{D}_2, \varphi, \mathcal{I}_1 \to \mathcal{I}_2 \rangle$, where

$$\varphi(f) = \{(X,a) \ / \ \exists x \in D_1 \text{ such that } X \subseteq^{fin} x \text{ and } a \in f(x)\}$$

6 CLS as a domain logic

CLS allows us to develope a logic of continuous domain following another line than the one used by Abramsky in [Abr91] to SFP domains.

6.1 Typed language and formation rules

We shall introduce a metalanguage for denotational semantics of programs, whose language of the typed expressions has the following syntax:

$$\sigma ::= \mathbf{1} \mid \sigma \times \tau \mid \sigma \to \tau \mid \sigma \oplus \tau \mid \sigma_{\uparrow} \mid \sigma^H \mid \sigma^S \mid \mathbf{rec}\ t.\sigma$$

where t is a variable type, and σ and τ ranges over type expressions. Each type expression σ is interpreted by a continuous domain, denoted by $\mathbf{D}_\sigma$. The basic expression type $\mathbf{1}$ is interpreted by the continuous domain consisting of an unique element $\mathbf{D_1} = \langle \{\bot\}, =, \bot \rangle$. The $\times$, $\to$, $\oplus$, $_{\uparrow}$, ^{H} and ^{S} are interpreted by the constructors on continuous domains: product, function space, coalesced sum, lifting, Hoare power domain and Smyth power domain, respectively. $\mathbf{rec}\ t.\sigma$ is the initial solution of the associated domain equation.

By using the above metalanguage we can provide denotational semantics for a large class of programming languages. To each programming language L can be associated a typed expression σ, where each program is denoted by an element of $\mathbf{D}_\sigma$ and their assertion language is $\mathcal{L}(I_\sigma)$, where $\mathcal{I}_\sigma$ is the information system representation of $\mathbf{D}_\sigma$.

6.2 Proof system

In order to provide axioms of the program logic, we will use the relation of logical consequence $\mid\!\sim_\sigma$ determined from the CLS $\mathcal{D}_\sigma$ for every type σ. Thus, $\alpha \mid\!\sim_\sigma \beta$ mean that each object (program) satisfying α also satisfies β in σ. For notational simplicity, we will eliminate subscripts if no confusion arise.

Logical Axioms and Logical Rules

$$(\mathsf{T})\ \frac{}{\alpha \mid\sim \mathsf{T}} \qquad (\mathsf{F})\ \frac{}{\mathsf{F} \mid\sim \alpha}$$

$$(Ref)\ \frac{}{\alpha \mid\sim \alpha} \qquad (trans)\ \frac{\alpha \mid\sim \beta \quad \beta \mid\sim \gamma}{\alpha \mid\sim \gamma}$$

$$(Sym)\ \frac{\alpha \mid\sim \beta \quad \beta \mid\sim \alpha}{\alpha \equiv \beta} \qquad (=)\ \frac{\alpha \equiv \beta}{\alpha \mid\sim \beta} \quad \frac{\alpha \equiv \beta}{\beta \mid\sim \alpha}$$

$$(\wedge_1)\ \frac{\alpha \mid\sim \beta}{\alpha \wedge \gamma \mid\sim \beta} \qquad (\wedge_2)\ \frac{\alpha \mid\sim \beta \quad \alpha \mid\sim \gamma}{\alpha \mid\sim \beta \wedge \gamma}$$

$$(\vee_1)\ \frac{\alpha \mid\sim \beta}{\alpha \mid\sim \beta \vee \gamma} \qquad (\vee_2)\ \frac{\alpha \mid\sim \gamma \quad \beta \mid\sim \gamma}{\alpha \vee \beta \mid\sim \gamma}$$

$$(Comm_1)\ \frac{}{\alpha \wedge \beta \mid\sim \beta \wedge \alpha} \qquad (Dist_1)\ \frac{}{\alpha \wedge (\beta \vee \gamma) \mid\sim (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)}$$

$$(Comm_2)\ \frac{}{\alpha \vee \beta \mid\sim \beta \vee \alpha} \qquad (Dist_2)\ \frac{}{(\alpha \vee \beta) \wedge (\alpha \vee \gamma) \mid\sim \alpha \vee (\beta \wedge \gamma)}$$

6.2.1 Conventions to assertion languages of complex types

In order to easily the notation of the assertion language to complex types, i.e. types obtained from others types via the constructors, is advicely to consider:

(i) $\mathcal{L}(I_\sigma)_\uparrow = (\mathcal{L}(I_\sigma) \times \{0\}) \cup \{\bot\}$ instead of $\mathcal{L}(I_{\sigma_\uparrow})$.

(ii) $\mathcal{L}(I_\sigma) \oplus \mathcal{L}(I_\tau) = \mathcal{L}(\widehat{I}_\sigma) \uplus \mathcal{L}(\widehat{I}_\tau)$ instead of $\mathcal{L}(I_{\sigma\oplus\tau}) = \mathcal{L}(I_\sigma \oplus I_\tau)$, where $\widehat{I} = I - \overline{\emptyset}$.

(iii) $\mathcal{L}(I_\sigma) \times \mathcal{L}(I_\tau)$ instead of $\mathcal{L}(I_{\sigma\times\tau}) = \mathcal{L}(I_\sigma \times I_\tau)$.

(iv) $\mathcal{L}(I_\sigma) \to \mathcal{L}(I_\tau) = \mathcal{P}^{fin}(\mathcal{L}(I_\sigma)) \times \mathcal{L}(I_\tau)$ instead of $\mathcal{L}(I_{\sigma\to\tau}) = \mathcal{L}(Con_\sigma \times I_\tau)$.

We can see it notations as codifications. For each constructor, this codification can be obtained via a codification function defined recursively as follow:

Lifting

$$cod_\uparrow : \mathcal{L}(I_\sigma)_\uparrow \longrightarrow \mathcal{L}(I_{\sigma_\uparrow})$$

- $cod_\uparrow(a, 0) = (a, 0)$ for each $a \in I_\sigma$
- $cod_\uparrow(\bot) = \mathsf{T}_{\sigma_\uparrow}$ or $cod_\uparrow(\bot) = \bot_{\sigma_\uparrow}$
- $cod_\uparrow(\mathsf{T}_\sigma, 0) = \bigvee_{a\in I_\sigma}(a, 0)$

- $cod_{\uparrow}(\mathsf{F}_{\sigma}, 0) = \mathsf{F}_{\sigma_{\uparrow}}$
- $cod_{\uparrow}(\alpha \wedge \beta, 0) = cod_{\uparrow}(\alpha, 0) \wedge cod_{\uparrow}(\beta, 0)$
- $cod_{\uparrow}(\alpha \vee \beta, 0) = cod_{\uparrow}(\alpha, 0) \vee cod_{\uparrow}(\beta, 0)$

Coalesced sum

$$cod_{\oplus} : \mathcal{L}(I_{\sigma}) \oplus \mathcal{L}(I_{\tau}) \longrightarrow \mathcal{L}(I_{\sigma} \oplus I_{\tau})$$

- $cod_{\oplus}(a, 1) = (a, 1)$ for each $a \in \widehat{I}_{\sigma}$
- $cod_{\oplus}(\mathsf{T}_{\sigma}, 1) = \mathsf{T}_{\sigma\oplus\tau}$
- $cod_{\oplus}(\mathsf{F}_{\sigma}, 1) = \mathsf{F}_{\sigma\oplus\tau}$
- $cod_{\oplus}(\alpha \wedge \beta, 1) = cod_{\oplus}(\alpha, 1) \wedge cod_{\oplus}(\beta, 1)$
- $cod_{\oplus}(\alpha \vee \beta, 1) = cod_{\oplus}(\alpha, 1) \vee cod_{\oplus}(\beta, 1)$
- $cod_{\oplus}(b, 2) = (b, 2)$ for each $b \in \widehat{I}_{\tau}$
- $cod_{\oplus}(\mathsf{T}_{\tau}, 2) = \mathsf{T}_{\sigma\oplus\tau}$
- $cod_{\oplus}(\mathsf{F}_{\tau}, 2) = \mathsf{F}_{\sigma\oplus\tau}$
- $cod_{\oplus}(\alpha \wedge \beta, 2) = cod_{\oplus}(\alpha, 2) \wedge cod_{\oplus}(\beta, 2)$
- $cod_{\oplus}(\alpha \vee \beta, 2) = cod_{\oplus}(\alpha, 2) \vee cod_{\oplus}(\beta, 2)$

Cartesian product

$$cod_{\times} : \mathcal{L}(I_{\sigma}) \times \mathcal{L}(I_{\tau}) \longrightarrow \mathcal{L}(I_{\sigma} \times I_{\tau})$$

- $cod_{\times}(a, b) = (a, b)$ for each $a \in I_{\sigma}$ and $b \in I_{\tau}$
- $cod_{\times}(\mathsf{T}_{\sigma}, \mathsf{T}_{\tau}) = \mathsf{T}_{\sigma\times\tau}$
- $cod_{\times}(\mathsf{F}_{\sigma}, \beta) = \mathsf{F}_{\sigma\times\tau}$
- $cod_{\times}(\alpha, \mathsf{F}_{\tau}) = \mathsf{F}_{\sigma\times\tau}$
- $cod_{\times}(\mathsf{T}_{\sigma}, b) = \bigvee_{a\in I_{\sigma}}(a, b)$
- $cod_{\times}(\alpha \wedge \beta, \gamma) = cod_{\times}(\alpha, \gamma) \wedge cod_{\times}(\beta, \gamma)$
- $cod_{\times}(\alpha, \beta \wedge \gamma) = cod_{\times}(\alpha, \beta) \wedge cod_{\times}(\alpha, \gamma)$
- $cod_{\times}(\alpha \vee \beta, \gamma) = cod_{\times}(\alpha, \gamma) \vee cod_{\times}(\beta, \gamma)$
- $cod_{\times}(\alpha, \beta \vee \gamma) = cod_{\times}(\alpha, \beta) \vee cod_{\times}(\alpha, \gamma)$

Function space

$$cod_{\rightarrow} : (\mathcal{L}(I_{\sigma}) \rightarrow \mathcal{L}(I_{\tau})) \longrightarrow \mathcal{L}(Con_{\sigma} \times I_{\tau})$$

- $cod_{\rightarrow}(X, a) = (X, a)$ for each $X \in Con_{\sigma}$ and $a \in I_{\tau}$
- $cod_{\rightarrow}(X, \alpha) = \mathsf{F}_{\sigma\rightarrow\tau}$ if $X \in \mathcal{P}^{fin}(I_{\sigma}) - Con_{\sigma}$

- $cod_{\rightarrow}(\Gamma, \mathsf{T}_{\tau}) = \mathsf{T}_{\sigma\rightarrow\tau}$
- $cod_{\rightarrow}(\Gamma, \mathsf{F}_{\tau}) = \mathsf{F}_{\sigma\rightarrow\tau}$
- $cod_{\rightarrow}(\{\alpha \wedge \beta, \alpha_1, \ldots, \alpha_n\}, \gamma) = cod_{\rightarrow}(\{\alpha, \beta, \alpha_1, \ldots, \alpha_n\}, \gamma)$
- $cod_{\rightarrow}(\{\alpha \vee \beta, \alpha_1, \ldots, \alpha_n\}, \gamma) = cod_{\rightarrow}(\{\alpha, \alpha_1, \ldots, \alpha_n\}, \gamma) \vee cod_{\rightarrow}(\{\beta, \alpha_1, \ldots, \alpha_n\}, \gamma)$
- $cod_{\rightarrow}(\Gamma, \alpha \wedge \beta) = cod_{\rightarrow}(\Gamma, \alpha) \wedge cod_{\rightarrow}(\Gamma, \beta)$
- $cod_{\rightarrow}(\Gamma, \alpha \vee \beta) = cod_{\rightarrow}(\Gamma, \alpha) \vee cod_{\rightarrow}(\Gamma, \beta)$
- $cod_{\rightarrow}(\{\mathsf{T}_{\sigma}, \alpha_1, \ldots, \alpha_n\}, \beta) = cod_{\rightarrow}(\{\alpha_1, \ldots, \alpha_n\}, \beta)$
- $cod_{\rightarrow}(\{\mathsf{F}_{\sigma}, \alpha_1, \ldots, \alpha_n\}, \beta) = \mathsf{F}_{\sigma\rightarrow\tau}$
- $cod_{\rightarrow}(\emptyset, \alpha) = \mathsf{T}_{\sigma\rightarrow\tau}$

6.2.2 Constructor logical rules

The logical rules for this constructors are defined considering this codifications of their assertion languages.

Lifting

$$(\uparrow - \mid\sim) \ \frac{\alpha \mid\sim_{\sigma} \beta}{(\alpha, 0) \mid\sim_{\sigma_{\uparrow}} (\beta, 0)}$$

$$(\uparrow - \wedge) \ \frac{\alpha_1, 0) \mid\sim_{\sigma_{\uparrow}} (\beta_1, 0) \quad (\alpha_2, 0) \mid\sim_{\sigma_{\uparrow}} (\beta_2, 0)}{(\alpha_1 \wedge \beta_1, 0) \mid\sim_{\sigma_{\uparrow}} (\alpha_2 \wedge \beta_2, 0)}$$

$$(\uparrow - \vee) \ \frac{(\alpha_1, 0) \mid\sim_{\sigma_{\uparrow}} (\beta_1, 0) \quad (\alpha_2, 0) \mid\sim_{\sigma_{\uparrow}} (\beta_2, 0)}{(\alpha_1 \vee \alpha_2, 0) \mid\sim_{\sigma_{\uparrow}} (\beta_1 \vee \beta_2, 0)}$$

Coalesced sum

$$(\oplus - \mid\sim_1) \ \frac{\alpha \mid\sim_{\sigma} \beta}{(\alpha, 1) \mid\sim_{\sigma\oplus\tau} (\beta, 1)}$$

$$\oplus - \mid\sim_2) \ \frac{\alpha \mid\sim_{\tau} \beta}{(\alpha, 2) \mid\sim_{\sigma\oplus\tau} (\beta, 2)}$$

$$(\oplus - \wedge_1) \ \frac{(\alpha_1, 1) \mid\sim_{\sigma\oplus\tau} (\beta_1, 1) \quad (\alpha_2, 1) \mid\sim_{\sigma\oplus\tau} (\beta_2, 1)}{(\alpha_1 \wedge \alpha_2, 1) \mid\sim_{\sigma\oplus\tau} (\beta_1 \wedge \beta_2, 1)}$$

$$(\oplus - \wedge_2) \ \frac{(\alpha_1, 2) \mid\sim_{\sigma\oplus\tau} (\beta_1, 2) \quad (\alpha_2, 2) \mid\sim_{\sigma\oplus\tau} (\beta_2, 2)}{(\alpha_1 \wedge \alpha_2, 2) \mid\sim_{\sigma\oplus\tau} (\beta_{\wedge}\beta_2, 2)}$$

$$(\oplus - \vee_1)\ \frac{(\alpha_1, 1) \mid\sim_{\sigma\oplus\tau} (\beta_1, 1) \quad (\alpha_2, 1) \mid\sim_{\sigma\oplus\tau} (\beta_2, 1)}{(\alpha_1 \vee \alpha_2, 1) \mid\sim_{\sigma\oplus\tau} (\beta_1 \vee \beta_2, 1)}$$

$$(\oplus - \vee_2)\ \frac{(\alpha_1, 2) \mid\sim_{\sigma\oplus\tau} (\beta_1, 2) \quad (\alpha_2, 2) \mid\sim_{\sigma\oplus\tau} (\beta_2, 2)}{(\alpha_1 \vee \alpha_2, 2) \mid\sim_{\sigma\oplus\tau} (\beta_1 \vee \beta_2, 2)}$$

Cartesian product

$$(\times - \mid\sim_1)\ \frac{\alpha \mid\sim_\sigma \beta \quad \gamma \mid\sim_\tau \delta}{(\alpha, \gamma) \mid\sim_{\sigma\times\tau} (\beta, \delta)}$$

$$(\times - \mid\sim_2)\ \frac{(\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta)}{(\beta, \alpha) \mid\sim_{\tau\times\sigma} (\delta, \gamma)}$$

$$(\times - \wedge_1)\ \frac{(\alpha_1, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta) \quad (\alpha_2, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta)}{(\alpha_1 \wedge \alpha_2, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta)}$$

$$(\times - \wedge_2)\ \frac{(\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_1, \delta) \quad (\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_2, \delta)}{(\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_1 \wedge \gamma_2, \delta)}$$

$$(\times - \vee_1)\ \frac{(\alpha_1, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta) \quad (\alpha_2, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta)}{(\alpha_1 \vee \alpha_2, \beta) \mid\sim_{\sigma\times\tau} (\gamma, \delta)}$$

$$(\times - \vee_2)\ \frac{(\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_1, \delta) \quad (\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_2, \delta)}{(\alpha, \beta) \mid\sim_{\sigma\times\tau} (\gamma_1 \vee \gamma_2, \delta)}$$

Function space

$$(\to - \mid\sim_1)\ \frac{\alpha_1 \mid\sim_\sigma \beta_1, \ldots, \alpha_n \mid\sim_\sigma \beta_n \quad \gamma \mid\sim_\tau \delta}{(\{\alpha_1, \ldots, \alpha_n\}, \gamma) \mid\sim_{\sigma\to\tau} (\{\beta_1, \ldots, \beta_n\}, \delta)}$$

$$(\to - \mid\sim_2)\ \frac{(\Gamma_1, \alpha) \mid\sim_{\sigma\to\tau} (\Theta, \beta)}{(\Gamma_1 \cup \Gamma_2, \alpha) \mid\sim_{\sigma\to\tau} (\Theta, \beta)}$$

$$(\to - \mid\sim_3)\ \frac{(\Gamma, \alpha) \mid\sim_{\sigma\to\tau} (\Theta_1, \beta)}{(\Gamma, \alpha) \mid\sim_{\sigma\to\tau} (\Theta_1 \cap \Theta_2, \beta)}$$

$$(\to - \wedge_1)\ \frac{(\{\alpha\} \cup \Gamma, \gamma) \mid\sim_{\sigma\to\tau} (\Theta, \delta) \quad (\{\beta\} \cup \Gamma, \gamma) \mid\sim_{\sigma\to\tau} (\Theta, \delta)}{(\{\alpha_1 \wedge \beta\} \cup \Gamma, \beta) \mid\sim_{\sigma\to\tau} (\Theta, \delta)}$$

$$(\rightarrow -\wedge_2)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)\quad(\Gamma,\beta)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}{(\Gamma,\alpha\wedge\beta)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}$$

$$(\rightarrow -\wedge_3)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta_1,\beta)\quad(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta_2,\beta)}{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta_1\cap\Theta_2,\beta)}$$

$$(\rightarrow -\wedge_4)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\beta)\quad(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\gamma)}{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\beta\wedge\gamma)}$$

$$(\rightarrow -\vee_1)\ \frac{(\{\alpha\}\cup\Gamma,\gamma)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)\quad(\{\beta\}\cup\Gamma,\gamma)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}{(\{\alpha\vee\beta\}\cup\Gamma,\gamma)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}$$

$$(\rightarrow -\vee_2)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)\quad(\Gamma,\beta)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}{(\Gamma,\alpha\vee\beta)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\delta)}$$

$$(\rightarrow -\vee_3)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\{\gamma_1\}\cup\Theta,\beta)\quad(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\{\gamma_2\}\cup\Theta,\beta)}{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\{\gamma_1\vee\gamma_2\}\cup\Theta,\beta)}$$

$$(\rightarrow -\vee_4)\ \frac{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\beta)\quad(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\gamma)}{(\Gamma,\alpha)\mid\sim_{\sigma\rightarrow\tau}(\Theta,\beta\vee\gamma)}$$

7 Soundness and Completness

The semantics of an information $\alpha\in\mathcal{L}(I_\sigma)$ is provide by their models in $\mathbf{D}_\sigma$, i.e. $[\![\alpha]\!]=Mod(\alpha)$

Considering this semantical notion, the completeness and soundness results can be expressed as follow:

Theorem 7.1 (Soundness and Completeness) *Let σ be a type and $\alpha,\beta\in\mathcal{L}(I_\sigma)$. Then*

$$\vdash\alpha\mid\sim\beta \text{ if, and only if, } \models\beta\sqsubseteq\alpha.\blacksquare$$

In other word, the information α entail the information β ($\alpha\mid\sim\beta$) iff all model of α is a model of β ($\beta\sqsubseteq\alpha$ or equivalently $Mod(\alpha)\subseteq Mod(\beta)$).

8 Conclusion

In this work, we used continuous information system to give a logical characterisation for continuous domains. Perhaps it is not a novelty, because that original motivation of Scott and Hoofman, latter, to introduce the Scott and the continuous information systems, respectively, was to present domains in a logical form. However, these approach was limited to deal with information system structures as a concrete representation of domains. Since it just

explore the categorical relations between domains and information systems without both, a treatment of the deep implicit logical aspects of information system and a logical conection between domains and information systems. Thus, this work gives the relation between a continuous domain and their continuous information system representation, by using a satisfaction notion. This notion, reveals several logical concepts, such as: consequence relation, information content, models of an information, theory of an elements, theorems, information content order, equivalence between information, etc. Thus CLS gives a logic to continuous domains which is complementar with the continuous domain logic of Abramsky [Abr91,BA95,Bed96,Keg99]. Moreover, we can recover the continuous domain logic from CLS.

References

[Abr87] Samson Abramsky. *Domain Theory in Logic Form.* In Symposium on Logic in Computer Science, pages 47-53. Computer Society Press of the IEEE, 1987.

[Abr91] Samson Abramsky. *Domain Theory in Logic Form.* In Annals of Pure and Applied Logic, 51:1-77, 1991.

[Aci91] Benedito Melo Acióly. *Fundamentos Computacionais da Matemática Intervalar.* Ph.D. Thesis, CPGCC-UFRGS, Porto Alegre-RS, Brazil, 1991.

[AJ94] Samson Abramsky and Achim Jung. Domain Theory. In *Handbook of Logic in Computer Science*, Vol. 3, Oxford university Press., 1994.

[BA95] Benjamín R. Callejas Bedregal and Benedito Melo Acióly. *Logic of Plotkin Continuous Domain.* In LNCS 911, pages 195-206, Springer-Verlag, 1995.

[BA99] Benjamín R. Callejas Bedregal and Benedito Melo Acióly. *Using Continuous Domain Logic to Solve Interval Constraint Satisfaction Problems.* In Memorias de la XXV Conferencia Latinoamericana de Informática - CLEI'99, Asunción, Paraguay, 1999. pages 661-672.

[Bed96] Benjamín R. Callejas Bedregal. *Sistemas de Informação Contínuos: Uma abordagem lógica e computacional para a matemática intervalar.* Ph.D. Thesis, UFPE-Depto. de Informática, Recife-PE, Brazil, 1996.

[Bed99] Benjamín R. Callejas Bedregal. *ω-Continuous cpo's, ω-algebraic cpo's, SFP Domains and Retracts of SFP Domains as Information Systems.* In Proceeding of 6^{th} Workshop on Logic, Language, Information and Computation - WoLLIC' 99. Itatiaia National Park-RJ, Brazil, 1999. pages 51-60.

[Bed01] Benjamín R. Callejas Bedregal. *Representing Some Categories of Domains as Information Systems Structures.* In Preliminary Conference Proceeding of Second International Symposium on Domain Theory, Chengdu-China, october 22-26, 2001.

[BL94] Benjamín R. Callejas Bedregal and Manoel Agamemnon Lopes. *Sistemas de Informação via Sistemas Semânticos*. In Memorias de la XX Conferencia Latinoamericana de Informática - CLEI-PANEL'94, Ciudad de México, September 19-23, 1994. Pages 467-478.

[Bun74] Mario Bunge. *Tratado de Filosofia Básica*. EPU, São Paulo, 1974.

[Car90] Roberto Lins de Carvalho. *Abstract Semantical Systems*. Technical report 008/90, LNCC, Rio de Janeiro, 1990.

[Gab94] Dov M. Gabbay (ed.). *What is a Logical System?*. Oxford Science Publications, 1994.

[GHK+80] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove and D. Scott. *A Compemdium of Continuous Lattices*. Springer-Verlag, Berlim, 1980.

[Gun85] Carl A. Gunter. *Comparing categories of domains*. In LNCS 239, pages 101-121. Springer-Verlag, 1985.

[Hoo93] R. Hoofman. *Continuous Information Systems*. Information and Computation, 105(1):42-71, 1993.

[Keg99] Mathias Kegelmann. *Continuous Domains in Logical Form*. Ph.D. Thesis, University of Birmingham, 1999.

[Par83] Rohit Parikh. Some applications of topology to program semantics. *Mathematical Systems Theory*, 16:111-131, 1983.

[Sco72] Dana Scott. *Continuous Lattices*. Lectures Notes in Mathematics, 274, pages 97-136, Springer-Verlag, 1972.

[Sco82] Dana Scott. *Domain for denotational semantics*. In LNCS, 140, Springer-Verlag, 1982.

[SW49] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. The University of Illinois Press, Cambridge-UK, 1949.

[Vic89] S.J. Vickers. *Topology via Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press., Cambridge-UK, 1989.

[Zha89] G.Q. Zhang. *Logics of Domains*. Ph.D. Thesis, University of Cambridge, 1989.