Eletronic Journal on Mathematics of Computation 1-1(2003)
URL: http://ejmc.ucpel.tche.br/

A Bi-Scott Domains Logic

Benjamín René Callejas Bedregal
Federal University of Rio Grande do Norte
Department of Informatic and Applied Mathematics
Laboratory of Logic and Computational Intelligence
Email: bedregal@dimap.ufrn.br

Roberto Callejas-Bedregal
Federal University of Paraíba
Department of Mathematics
Email: roberto@mat.ufpb.br

Abstract

Logic domain was introduced by Abramsky [1] in order to study the logical aspects (a proof system) of Scott domains as used in the denotational semantics of programming languages. Thus, logic domain presents denotational semantics as a logic program. Since then, several researches have extended the Abramsky logic domain to other categories of domains, such as SFP domains [30, 2], DI-Domains [31], continuous domains [6] and Plotkin continuous domains [5].

On the other hand, the bi-Scott domain was introduced in [9] in order to study a kind of domain, which are closed under the interval constructor, and to give denotational semantic to programming languages which consider a parametric interval data type.

In this work, we will define a logic domain to bi-Scott domains which takes into account the particularities of this category of domain.

Keywords: *bi-Scott domains, Interval constructor, domain logic.*

1 Introduction

Domain theory studies a class of appropriate spaces to obtain semantic functions to give denotational semantics of programming languages. Several mathematical structures had been proposed as models for denotational semantics (semantic domains) of programming languages. Most of those mathematical structures are a kind of partial orders satisfying the following five properties [17]:

1. it must be closed under the desired operators (product, function space, etc.);
2. canonical solutions to recursive equations must exist;
3. it must define an intuitive and flexible computability theory;
4. it must be natural to motivate and simple to describe;
5. it must have the desired basic data types.

The logic domain was introduced by Abramsky [1] in order to study the logical aspects (a proof system) of Scott domains as used in denotational semantics of programming languages, in the sense that the denotation of any program coincides with the set of true assertions of it. Since then, several works defining a logic to other categories of domains were developed, for example [31] defines a logic to DI-domains, [30, 2] defines a SFP logic domain, and [5] defines a logic to continuous domains.

On the other hand, R. Moore [22] developed an interval mathematic in order to provide a control of the computational errors resulting from numeric computations involving real numbers. The Moore theory not only comprehend real intervals, but also complex intervals, matrix and array of real and complex intervals. Several subtypes of the real interval have been used in the interval literature, for example the interval of real numbers in the interval $[0, 1]$ are used in fuzzy interval logic [29, 25, 26] and interval probability [13, 8], the interval of rational numbers [4, 14] and interval of float point numbers [20, 19, 12].

In domain theory, Dana Scott in [24] saw the real intervals as a continuous domain (which is not algebraic) and suggested to be considered as the domain for the real data type. Lately, Acióly in [3] used the domain of real intervals to provide a computational foundation for the interval mathematics and in [6, 7, 15, 16] was provided a computability theory for the real intervals based in this domain and in the notion of the effectively given continuous domains. It is clear that, analogously to the real interval domain, we can also define an appropriate domain for complex intervals, matrix of real and complex intervals (which can be seen as isomorphic to interval of real and complex matrix, respectively), etc. and, yet, we can construct other types of intervals (such as interval of tuples of real numbers) by using the usual domain constructors. Nevertheless, this "solution" compel us to define a novel domain for each new interval data type which can be necessary for some problem. For example, recently have been used the notion of interval of digital images by Lyra [21] to deal with the errors account in process of digitalization of images.

A domain constructor was introduced in [9, 10] in order to define an interval theory which consider the above kind of intervals and any possible type of interval, with no need to define a new domain for the specific case. But the usual domains are not closed under this interval constructor. Nevertheless, the same authors provided in [9] a new category of domain, namely bi-Scott domains, which is closed under this constructor, and, in [10], they proved that this category of domain is cartesian closed, which is an important condition in order to conclude that this category of domains is adequate to give denotational semantic of programming languages which consider interval as a parametric data type.

In this work, we propose a logic domain for this category of domains. It will allows us to provide a proof system for bi-Scott domains which agrees with the denotational semantic of programming languages.

2 Bi-Scott domains

A binary relation $\leq$ on a set D is a pre-order, if it is reflexive ($x \leq x$ for all $x \in A$) and transitive (if for all $x, y, z \in D$, $x \leq y$ and $y \leq z$ then $x \leq z$). If $\leq$ is also antisimetric (i.e. if for all $x, y \in D$ such that $x \leq y$ and $y \leq x$ then $x = y$) it is called a partial order and the pair $\mathbf{D} = \langle D, \leq \rangle$ is called a partial ordered set (poset).

A non-empty set $\Delta \subseteq D$ is called *directed* if each non-empty finite subset of it has an upper bound, or equivalently, if $\forall a, b \in \Delta \; \exists c \in \Delta$ such that $a \preceq c$ and $b \preceq c$.

A poset $\mathbf{D}$ is *direct complete* (*dcpo* for short) if each directed subset Δ has a least upper bound, (or supremum) denoted by $\bigsqcup \Delta$, in $\mathbf{D}$.

Let $\mathbf{D} = \langle D, \leq \rangle$ be a dcpo. An element $a \in D$ is *compact* if for each directed set Δ such that $a \leq \bigsqcup \Delta$ then $a \leq x$ for some $x \in \Delta$. We will denote the set of all compact elements of $\mathbf{D}$ by $\mathbf{D}^0$. $\mathbf{D}$ is *algebraic* if for each $x \in D$, the set $\mathring{\downarrow} x = \{y \in \mathbf{D}^0 \;/\; y \leq x\}$ is directed and $x = \bigsqcup \mathring{\downarrow} x$. $\mathbf{D}$ is ω-algebraic if $\mathbf{D}$ is algebraic and $\mathbf{D}^0$ is countable. A subset $X \subseteq D$ is *consistent* if there exists an upper bound in $\mathbf{D}$, i.e. if there exists $x \in D$ such that for each $a \in X$, $a \leq x$. A dcpo $\mathbf{D}$ is *consistently complete* if each nonempty consistent set $X \subseteq D$ has a supremum in D (i.e. $\bigsqcup X$ exists in $\mathbf{D}$). $\mathbf{D}$ is a *Scott domain* if it is a consistently complete ω-algebraic dcpo[1].

An ideal of a pre-order $\mathbf{D} = \langle D, \leq \rangle$ is a set $I \subseteq D$ such that I is a directed set and satisfies: for each $x \in I$ if $y \leq x$ then $y \in I$. We write $\widehat{D}$ for the set of ideals over the pre-order $\mathbf{D}$. The poset $\widehat{\mathbf{D}} = \langle \widehat{D}, \subseteq \rangle$ is called an ideal completion of $\mathbf{D}$. The set $\downarrow x = \{y \in D \;/\; y \leq x\}$ is an ideal called the main ideal in $\mathbf{D}$ generated by x. All ideals in $\mathbf{D}$ can be obtained as an arbitrary union of main ideals of $\mathbf{D}$.

Let $\mathbf{D} = \langle D, \leq_D \rangle$ and $\mathbf{E} = \langle E, \leq_E \rangle$ be dcpo's. A function $f : E \longrightarrow D$ is called *continuous* if it is monotonic ($x \leq_E y$ implies $f(x) \leq_D f(y)$) and preserves least upper bounds ($f(\bigsqcup \Delta) = \bigsqcup f(\Delta)$). $\mathbf{D}$ is order isomorphic to $\mathbf{E}$, denoted by $\mathbf{D} \cong \mathbf{E}$, if there is a bijective continuous function $f : D \longrightarrow E$ such that their reverse ($f^{-1} : E \longrightarrow D$) is also continuous. Isomorphism preserve the properties of the posets, for example if $\mathbf{D}$ is a Scott domain, $\mathbf{E}$ is a poset and $\mathbf{D} \cong \mathbf{E}$ then $\mathbf{E}$ is also a Scot domain.

Definition 2.1
Let $\mathbf{D} = \langle D, \leq \rangle$ be a poset. The **reverse poset** of $\mathbf{D}$, denoted by $\mathbf{D}^{op}$, is the pair $\mathbf{D}^{op} = \langle D^{op}, \leq_{op} \rangle$, where $D^{op} = D$ and $x \leq_{op} y$ if $y \leq x$.

Definition 2.2
Let $\mathbf{D} = \langle D, \leq \rangle$ be a Scott domain. If $\mathbf{D}^{op}$ is also a **Scott domain**, then $\mathbf{D}$ is said a **bi-Scott domain**.

Notice that not every Scott domain is a bi-Scott domain. For example, let $\mathcal{N} = \langle \mathbb{N}, \leq \rangle$ be the poset of natural numbers with their usual orders. Clearly, $\mathcal{N}$ is not a dcpo, and therefore is not a Scott domain. But $\mathcal{N}^{op}$ is a Scott domain. So $\mathcal{N}^{op}$ is a Scott domain which is not bi-Scott because their opposite poset ($\mathcal{N}$) is not a Scott domain. Another example of a Scott domain which is not a bi-Scott domain is the poset illustrated in the figure 1, in this case the reverse poset illustrated in the figure 2 is not algebraic, and therefore it is not a Scott domain, because the set $\downarrow \top \cap \mathbf{D}^0$ is not a directed set.

Since bi-Scott domains have a stronger structure than Scott domains, their morphisms must consider this additional structure.

[1] Usually Scott domains are defined as consistently complete ω-algebraic cpo. But since we will work with dcpo's, we need this definitions.

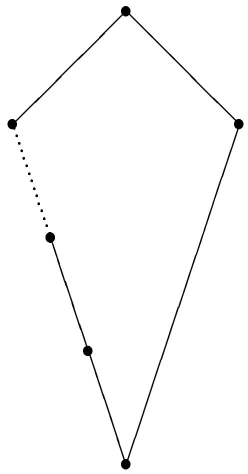

Figure 1: A Scott domain which is not a bi-Scott domain

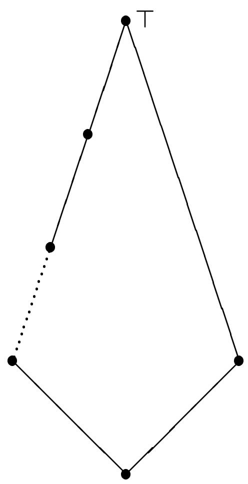

Figure 2: An algebraic cpo which is not a Scott domain

Definition 2.3
Let $\mathbf{D} = \langle D, \leq_D \rangle$ and $\mathbf{E} = \langle E, \leq_E \rangle$ be bi-Scott domains. A continuous function $f : D \longrightarrow E$ is **bi-continuous** if it is continuous with respect to $\mathbf{D}^{op}$ and $\mathbf{E}^{op}$.

Notice that bi-continuous functions can be seen as a generalization of stable functions used in DI-domains [31], in the sense that preserve supremum and infimum.

Definition 2.4
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. We say that a subset $O \subseteq D$ is a **Scott open set** of $\mathbf{D}$ if

1. for each $x, y \in D$ such that $x \leq y$ and $x \in O$ then $y \in O$
2. for each direct set $\Delta \subseteq D$ such that $\bigsqcup \Delta \in O$ there exists $y \in \Delta$ such that $y \in O$.

The set of open sets of $\mathbf{D}$ is denoted by $\Omega_S(\mathbf{D})$ and it is called the **Scott topology** of $\mathbf{D}$.

Proposition 2.1
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. Then $\Omega_S(\mathbf{D})$ is a topology and $\uparrow \mathbf{D}^0 = \{\uparrow x \ / \ x \in \mathbf{D}^0\}$, where $\uparrow x = \{y \in D \ / \ x \leq y\}$, is a base of $\Omega_S(\mathbf{D})$.

Proof: See [28]. ∎

3 Constructors on bi-Scott 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, disjunted and collapsed sum (sometimes is also considered the powerdomain constructors).

3.1 The interval constructor

In [24] Dana Scott introduced a continuous domain of closed real interval ordered by reverse inclusion in order to present an appropriate domain to represent the real data type. This continuous domain was used by Acióly in [3] to give a computational foundation of the interval mathematics of Moore [22]. Notice that both the real intervals and the order on the set of real intervals depends upon the usual order on the real numbers. In [9] this constructions was generalized as a constructor on the category **POSET**. This results are given in the following.

Definition 3.1
[9] Let $\mathbf{D} = \langle D, \leq \rangle$ be a poset. The poset $\mathbb{I}(\mathbf{D}) = \langle \mathbb{I}(D), \sqsubseteq \rangle$, where

- $\mathbb{I}(D) = \{[a, b] \ / \ a, b \in D \text{ and } a \leq b\}$
- $[a, b] \sqsubseteq [c, d]$ if $a \leq c$ and $d \leq b$,

is called the **poset of intervals of D**.

There are two natural functions from $\mathbb{I}(D)$ to D, which are the left and right projections $l : \mathbb{I}(D) \longrightarrow D$ and $r : \mathbb{I}(D) \longrightarrow D$ respectively, defined by

$$l([a,b]) = a \text{ and } r([a,b]) = b$$

Clearly the function l is monotonic and therefore it is a morphism from the poset $\mathbb{I}(\mathbf{D})$ to the poset $\mathbf{D}$, but the function r is not monotonic. But this could be overcome by reverting the order on $\mathbf{D}$.

Theorem 3.1
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. Then $\mathbb{I}(\mathbf{D})$ is a bi-Scott domain and the functions $l : \mathbb{I}(D) \longrightarrow D$ and $r : \mathbb{I}(D) \longrightarrow D^{op}$ are bi-continuous.

Proof: In [9] was showed that $\mathbb{I}(\mathbf{D})$ is a bi-Scott domain and that the functions l and r are continuous. Then, we need only to show that these functions are continuous w.r.t. the opposite posets, i.e. that $l : \mathbb{I}(D)^{op} \longrightarrow D^{op}$ and $r : \mathbb{I}(D)^{op} \longrightarrow D$ are continuous.

Let $\Delta \in \mathbb{I}(D)^{op}$ be a directed set in $\mathbb{I}(\mathbf{D})^{op}$. Since $\mathbb{I}(\mathbf{D})^{op}$ is a dcpo, then $\bigsqcup_{op} \Delta$ there exists. We will show that $\bigsqcup_{op} \Delta = [\bigsqcup_{op} l(\Delta), \bigsqcup r(\Delta)]$. Suppose that $\bigsqcup_{op} l(\Delta) = x$ and $\bigsqcup r(\Delta) = y$. Let $[a,b] \in \Delta$, then, trivially, $a \leq_{op} x$ and $b \leq y$, and therefore $[a,b] \sqsubseteq_{op} [x,y]$, i.e. $[x,y]$ is an upper bound (in $\mathbb{I}(\mathbf{D})^{op}$) of Δ. If $[u,v]$ is another upper bound of Δ, then for each $[a,b] \in \Delta$, $[a,b] \sqsubseteq_{op} [u,v]$, and therefore $a \leq_{op} u$ and $b \leq v$. So, $\bigsqcup_{op} l(\Delta) \leq_{op} u$ and $\bigsqcup r(\Delta) \leq v$, i.e. $x \leq_{op} u$ and $y \leq v$. Thus, $[x,y] \sqsubseteq [u,v]$ and therefore $\bigsqcup_{op} \Delta = [\bigsqcup_{op} l(\Delta), \bigsqcup r(\Delta)]$. Therefore,

$$\begin{aligned} l(\textstyle\bigsqcup_{op} \Delta) &= l([\textstyle\bigsqcup_{op} l(\Delta), \textstyle\bigsqcup r(\Delta)]) \\ &= \textstyle\bigsqcup_{op} l(\Delta) \end{aligned}$$

$$\begin{aligned} r(\textstyle\bigsqcup_{op} \Delta) &= r([\textstyle\bigsqcup_{op} l(\Delta), \textstyle\bigsqcup r(\Delta)]) \\ &= \textstyle\bigsqcup r(\Delta). \end{aligned}$$

Therefore, $l : \mathbb{I}(D)^{op} \longrightarrow D^{op}$ and $r : \mathbb{I}(D)^{op} \longrightarrow D$ are continuous. ■

Proposition 3.1
Let $\mathbf{D} = \langle D, \sqsubseteq \rangle$ be a bi-Scott domain. Then $\mathbb{I}(\mathbf{D})^0 = \mathbb{I}(\mathbf{D}^0)$ and $\mathbb{I}(\mathbf{D}^{op})^0 = \mathbb{I}((\mathbf{D}^{op})^0)$.

Proof: See [9]. ■

3.2 The lifting constructor

Definition 3.2
Let $\mathbf{D} = \langle D, \leq \rangle$ be a poset. The poset $\mathbf{D}_\uparrow = \langle D_\uparrow, \leq_\uparrow \rangle$, where

- $D_\uparrow = \{\perp\} \cup (D \times \{0\})$
- $(a,0) \leq_\uparrow (b,0)$ if $a \leq b$
- $\perp \leq_\uparrow x$ for each $x \in D_\uparrow$

is called the **lifting of D**.

Proposition 3.2
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. Then $\mathbf{D}_{\uparrow}$ is a bi-Scott domain.

Proof: If $\mathbf{D}$ is a bi-Scott domain, then $\mathbf{D}$ is a Scott domain and therefore $\mathbf{D}_{\uparrow}$ is also a Scott domain (see a proof in [27, 6]). It remain to prove that $(\mathbf{D}_{\uparrow})^{op}$ is a Scott domain. Before, notice that,

$$\begin{array}{lll} (x,0) \leq_{\uparrow_{op}} (y,0) & \Leftrightarrow & (y,0) \leq_{\uparrow} (x,0) \\ & \Leftrightarrow & y \leq x \\ & \Leftrightarrow & x \leq_{op} y \\ \perp \leq_{\uparrow_{op}} x & \Leftrightarrow & x \leq_{\uparrow} \perp \\ & \Leftrightarrow & x = \perp \end{array}$$

So, $\perp$ is the greater element of $(\mathbf{D}_{\uparrow})^{op}$.

Let $\Delta \subseteq (D_{\uparrow})^{op}$ be a directed set. If $\perp \in \Delta$, then because $\perp$ is a greater element, $\bigsqcup \Delta = \perp$. If $\perp \not\in \Delta$ then, trivially, $\Delta_0 = \{x \in D \ / \ (x,0) \in \Delta\}$ is a directed set in $\mathbf{D}^{op}$. Because, $\mathbf{D}_{op}$, is a Scott domain, $\bigsqcup \Delta_0$ exists. So, trivially, $(\bigsqcup \Delta_0, 0) = \bigsqcup \Delta$. Therefore $(\mathbf{D}_{\uparrow})^{op}$ is a dcpo.

If $X \subseteq (D_{\uparrow})^{op}$ has an upper bound different of $\perp$, i.e. $(u,0) \in (D_{\uparrow})^{op}$, then u is an upper bound of $X_0 = \{x \in D \ / \ (x,0) \in X\}$ in $\mathbf{D}^{op}$. So, because $\mathbf{D}^{op}$ is a Scott domain, X_0 has a supremum. Therefore, trivially, $(\bigsqcup X_0, 0)$ is a supremum of X. If $X \subseteq (D_{\uparrow})^{op}$ has only $\perp$ as an upper bound, then $\bigsqcup X = \perp$. So, $(\mathbf{D}_{\uparrow})^{op}$ is consistently complete.

Let $x \in (D^{op})^0$ and $\Delta \subseteq (D_{\uparrow})^{op}$ be a directed set in $(\mathbf{D}_{\uparrow})^{op}$. Let $\Delta_0 = \{y \in D \ / \ (y,0) \in \Delta\}$ then $(\bigsqcup \Delta_0, 0) = \bigsqcup \Delta$. Thus, if $(x,0) \leq_{\uparrow^{op}} \bigsqcup \Delta$ then $x \leq_{op} \bigsqcup \Delta_0$. So, because x is a compact, there exists $z \in \Delta_0$ such that $x \leq_{op} z$ and therefore $(z,0) \in \Delta$ and $(x,0) \leq_{\uparrow^{op}} (z,0)$. So, $(x,0)$ is also compact in $((\mathbf{D}_{\uparrow})^{op})^0$. It is easy to see now that $((\mathbf{D}_{\uparrow})^{op})^0 = (\mathbf{D}^{op})^0 \cup \{\perp\}$.

Trivially, $\perp = \bigsqcup(\downarrow \perp \cap ((\mathbf{D}_{\uparrow})^{op})^0$. On the other hand, if $(x,0) \in (D_{\uparrow})^{op}$ then

$$\begin{array}{ll} \bigsqcup(\downarrow (x,0) \cap ((\mathbf{D}_{\uparrow})^{op})^0)) & = \bigsqcup((\downarrow x \cap (\mathbf{D}^{op})^0) \times \{0\}) \\ & = (\bigsqcup(\downarrow x \cap (\mathbf{D}^{op})^0), 0) \\ & = (x,0) \end{array}$$

So, $(\mathbf{D}_{\uparrow})^{op}$ is an algebraic domain, and therefore it is a Scott domain.

■

3.3 The cartesian product constructor

Definition 3.3
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. The **cartesian product** of $\mathbf{D}_1$ with $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \times \mathbf{D}_2$, is the poset $\mathbf{D}_1 \times \mathbf{D}_2 = \langle D_1 \times D_2, \leq \rangle$ where

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

Proposition 3.3
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. Then the poset $\mathbf{D}_1 \times \mathbf{D}_2$ is also a bi-Scott domain.

Proof: See [11]. ■

3.4 The sum constructor

Definition 3.4
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. The **sum** of $\mathbf{D}_1$ with $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \oplus \mathbf{D}_2$, is the poset $\mathbf{D}_1 \oplus \mathbf{D}_2 = \langle D_1 \biguplus D_2, \leq \rangle$ where

$$D_1 \biguplus D_2 = \{(x, 0) \ / \ x \in D_1\} \cup \{(x, 1) \ / \ x \in D_2\}$$

$$(x, n) \leq (y, m) \text{ if } n = m \text{ and } x \leq_n y$$

Proposition 3.4
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. Then the poset $\mathbf{D}_1 \oplus \mathbf{D}_2$ is also a bi-Scott domain.

Proof: If $\mathbf{D}_1$ and $\mathbf{D}_2$ are bi-Scott domains, then trivially $\mathbf{D}_1 \oplus \mathbf{D}_2$, $\mathbf{D}_1^{op}$, $\mathbf{D}_2^{op}$ and $\mathbf{D}_1^{op} \oplus \mathbf{D}_2^{op} = \langle D_1^{op} \biguplus D_2^{op}, \leq^{op} \rangle$ are also Scott domains. We will show that $\mathbf{D}_1^{op} \oplus \mathbf{D}_2^{op} = (\mathbf{D}_1 \oplus \mathbf{D}_2)^{op}$.

$$(D_1 \biguplus D_2)^{op} = D_1 \biguplus D_2 = D_1^{op} \biguplus D_2^{op}$$

$$\begin{array}{rcl} (x, n) \leq_{op} (y, m) & \Leftrightarrow & (y, m) \leq (x, n) \\ & \Leftrightarrow & n = m \text{ and } y \leq_n x \\ & \Leftrightarrow & n = m \text{ and } x \leq_{n^{op}} y \\ & \Leftrightarrow & (x, n) \leq^{op} (y, n) \end{array}$$

■

3.5 The bi-continuous function space constructor

Definition 3.5
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. The **bi-continuous function space** from $\mathbf{D}_1$ into $\mathbf{D}_2$, denoted by $\mathbf{D}_1 \to \mathbf{D}_2$, is the poset $\mathbf{D}_1 \to \mathbf{D}_2 = \langle D_1 \to D_2, \leq \rangle$ where

1. $D_1 \to D_2 = \{f : D_1 \longrightarrow D_2 \ / f \text{ is a bi-continuous function}\}$
2. $f \leq g$ if for each $x \in D_1$, $f(x) \leq_2 g(x)$.

Proposition 3.5
Let $\mathbf{D}_1 = \langle D_1, \leq_1 \rangle$ and $\mathbf{D}_2 = \langle D_2, \leq_2 \rangle$ be bi-Scott domains. Then the poset $\mathbf{D}_1 \to \mathbf{D}_2$ is also a bi-Scott domain.

Proof: See [11]. ■

3.6 The Powerdomain constructors

Definition 3.6
Let $\mathbf{D} = \langle D, \leq \rangle$ be a poset. $\mathbf{D}_S = \langle D_S, \preceq_S \rangle$ and $\mathbf{D}_H = \langle D_H, \preceq_H \rangle$, where

- $D_S = D_H = \mathcal{P}_{\emptyset}^{fin}(\mathbf{D}^0 \cup (\mathbf{D}^{op})^0) = \{X \subseteq \mathbf{D}^0 \cup (\mathbf{D}^{op})^0 \ / \ X \text{ is a finite non empty set}\}$
- $X \preceq_S Y$ if $\forall y \in Y$, $\exists x \in X$ such that $x \leq y$
- $X \preceq_H Y$ if $\forall x \in X$, $\exists y \in Y$ such that $x \leq y$

are called the **Smyth and Hoare pre-orders of $\mathbf{D}$**[2], respectively.

Proposition 3.6
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. Let $(\mathbf{D}_H)^{op} = \langle (D_H)^{op}, \preceq_{Hop} \rangle$ and $(\mathbf{D}_S)^{op} = \langle (D_S)^{op}, \preceq_{Sop} \rangle$ defined by $(D_H)^{op} = (D_S)^{op} = \mathcal{P}_{\emptyset}^{fin}(\mathbf{D}^0 \cup (\mathbf{D}^{op})^0)$ and

$$X \preceq_{Hop} Y \Leftrightarrow Y \preceq_H X \text{ and } X \preceq_{Sop} Y \Leftrightarrow Y \preceq_S .$$

Then $(\mathbf{D}_H)^{op} = (\mathbf{D}^{op})_S$ and $(\mathbf{D}_S)^{op} = (\mathbf{D}^{op})_H$.

Proof: If $\mathbf{D} = \langle D, \leq \rangle$ is a bi-Scott domain, then $\mathbf{D}$ and $\mathbf{D}^{op}$ are Scott domains, and therefore $\mathbf{D}_H = \langle D_H, \preceq_H \rangle$, $(\mathbf{D}^{op})_H = \langle (D^{op})_H, \preceq_{opH} \rangle$, $\mathbf{D}_S = \langle D_S, \preceq_S \rangle$ and $(\mathbf{D}^{op})_S = \langle (D^{op})_S, \preceq_{opS} \rangle$ are Hoare and Smyth pre-orders.

Clearly,
$(D_H)^{op} = D_H = D_S = (D^{op})_S = \mathcal{P}_{\emptyset}^{fin}(\mathbf{D}^0 \cup (\mathbf{D}^{op})^0)$ and

$$\begin{aligned} X \preceq_{Hop} Y &\Leftrightarrow Y \preceq_H X \\ &\Leftrightarrow \forall y \in Y \exists x \in X.\ y \leq x \\ &\Leftrightarrow \forall y \in Y \exists x \in X.\ x \leq^{op} y \\ &\Leftrightarrow X \preceq_{opS} Y \end{aligned}$$

So $(\mathbf{D}_H)^{op} = (\mathbf{D}^{op})_S$.
Analogously, $(D_S)^{op} = D_S = D_H = (D^{op})_H$ and

$$\begin{aligned} X \preceq_{Sop} Y &\Leftrightarrow Y \preceq_S X \\ &\Leftrightarrow \forall x \in X \exists y \in Y.\ y \leq x \\ &\Leftrightarrow \forall x \in X \exists y \in Y.\ x \leq^{op} y \\ &\Leftrightarrow X \preceq_{opH} Y \end{aligned}$$

So $(\mathbf{D}_S)^{op} = (\mathbf{D}^{op})_H$. ■

Definition 3.7
Let $\mathbf{D} = \langle D, \leq \rangle$ be a Scott domain. The ideal completion of $\mathbf{D}_S$ and $\mathbf{D}_H$, that is $\widehat{\mathbf{D}_S} = \langle \widehat{D_S}, \subseteq \rangle$ and $\widehat{\mathbf{D}_H} = \langle \widehat{D_H}, \subseteq \rangle$, respectively, are called the Smyth and the Hoare Powerdomain, respectively.

Proposition 3.7
Let $\mathbf{D} = \langle D, \leq \rangle$ be a Scott domain. Then $\widehat{\mathbf{D}_S}$ and $\widehat{\mathbf{D}_H}$ are Scott domains.

Proof: See [27, 6]. ■

Proposition 3.8
Let $\mathbf{D} = \langle D, \leq \rangle$ be a bi-Scott domain. Then $\widehat{\mathbf{D}_S}$ and $\widehat{\mathbf{D}_H}$ are bi-Scott domains.

Proof: If $\mathbf{D} = \langle D, \leq \rangle$ is a bi-Scott domain, then $\mathbf{D}$ and $\mathbf{D}^{op}$ are Scott domains. So, by proposition 3.7, $\widehat{\mathbf{D}_H}$, $\widehat{(\mathbf{D}^{op})_H}$, $\widehat{\mathbf{D}_S}$ and $\widehat{(\mathbf{D}^{op})_S}$ are Scott domains.

It remain to prove that $\widehat{\mathbf{D}_H}^{op} = \langle \widehat{D_H}^{op}, \subseteq^{op} \rangle$ and $\widehat{\mathbf{D}^S}^{op} = \langle \widehat{D_S}^{op}, \subseteq^{op} \rangle$ are also Scott domains. We will show that $\widehat{\mathbf{D}_H}^{op}$ and $\widehat{(\mathbf{D}^{op})_S}$ are order isomorphics.

[2]Originally [23, 31, 18, 27], the D_S and D_H are defined by $\mathcal{P}_{\emptyset}^{fin}(\mathbf{D}^0)$, but because in our construction we will consider the reverse poset, we also included $(\mathbf{D}^{op})^0$.

The isomorphism is provided by the function $h : \widehat{(D^{op})_S} \longrightarrow \widehat{D_H}^{op}$ defined by the main ideals as follows:

$$h(\downarrow_{\widehat{opS}} X) = \downarrow_{\widehat{H}op} X.$$

Since any ideal is the union of main ideals, this definition can be trivially extended for $h : \widehat{(D^{op})_S} \longrightarrow \widehat{D_H}^{op}$ as follows:

$$h(\bigcup_{i \in I} \downarrow_{\widehat{opS}} X_i) = \bigcup_{i \in I} \downarrow_{\widehat{H}op} X_i.$$

Clearly, this function is bijective (its inverse, h^{-1}, is defined by $h^{-1}(\bigcup_{i \in I} \downarrow_{\widehat{H}op} X_i) = \bigcup_{i \in I} \downarrow_{\widehat{opS}} X_i$.
Notice that:

$$\begin{aligned}
\downarrow_{\widehat{opS}} X \subseteq \downarrow_{\widehat{opS}} Y \quad & \Leftrightarrow X \preceq_{opS} Y \\
& \Leftrightarrow X \preceq_{Hop} Y \\
& \Leftrightarrow Y \preceq_H X \\
& \Leftrightarrow \downarrow_{\widehat{H}} Y \subseteq \downarrow_{\widehat{H}} X \\
& \Leftrightarrow \downarrow_{\widehat{H}op} X \subseteq^{op} \downarrow_{\widehat{H}op} Y \\
& \Leftrightarrow h(\downarrow_{\widehat{opS}} X) \subseteq^{op} h(\downarrow_{\widehat{opS}} Y)
\end{aligned}$$

Thus, h is monotonic.

The proof of continuity of h is clear from the above results. So, $\widehat{\mathbf{D}_H}^{op}$ is a Scott domain, and therefore $\widehat{\mathbf{D}_H}$ is a bi-Scott domain.

Analogously, we will show that $\widehat{(\mathbf{D}^{op})_H}$ and $\widehat{\mathbf{D}_S}^{op}$ are order isomorphics. The isomorphism is given by the function $s : \widehat{(D^{op})_H} \longrightarrow \widehat{D_S}^{op}$ defined by

$$s(\downarrow_{\widehat{opH}})X = \downarrow_{\widehat{S}op} X.$$

Similarly to h, this definition can be trivially extended for $s : \widehat{(D^{op})_H} \longrightarrow \widehat{D_S}^{op}$ and s is bijective.
Notice that:

$$\begin{aligned}
\downarrow_{opH} X \subseteq \downarrow_{opH} Y \quad & \Leftrightarrow X \preceq_{opH} Y \\
& \Leftrightarrow X \preceq_{Sop} Y \\
& \Leftrightarrow Y \preceq_S X \\
& \Leftrightarrow \downarrow_{\widehat{S}} Y \subseteq \downarrow_{\widehat{S}} X \\
& \Leftrightarrow \downarrow_{\widehat{S}op} X \subseteq^{op} \downarrow_{\widehat{S}op} Y \\
& \Leftrightarrow s(\downarrow_{\widehat{opH}} X) \subseteq^{op} s(\downarrow_{\widehat{opH}} Y)
\end{aligned}$$

Thus, s is monotonic and from the above result the continuity of s is clear. So, $\widehat{\mathbf{D}_S}^{op}$ is a Scott domain, and therefore $\widehat{\mathbf{D}_S}$ is a bi-Scott domain. ■

By notational simplicity and to deal with the powerdomains as constructors, we will use in the next sections the notation $\mathbf{D}^S$ and $\mathbf{D}^H$ instead of $\widehat{\mathbf{D}_S}$ and $\widehat{\mathbf{D}_H}$, respectively.

4 Typed language and formation rules

In this section, we shall introduce a metalanguage for denotational semantics of programs, whose language of the typed expressions has the following syntax:

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

where t is a variable type, and σ and τ ranges over type expressions. Each type expression σ is interpreted by a bi-Scott domain, denoted by $\mathbf{D}_\sigma$. The basic expression type $\mathbf{1}$ is interpreted by the bi-Scott domain consisting of a unique element $\mathbf{D_1} = \langle\{\perp\}, =\rangle$. The $\times$, $\to$, $\oplus$, ^{op}, ${}_{\uparrow}$, $\mathbb{I}(\ )$, ^{H} and ^{S} are interpreted by the constructors on bi-Scott domains: cartesian product, bi-continuous function space, sum, opposite, lifting, interval, Hoare powerdomain and Smyth powerdomain, respectively. For example the expression type $\mathbb{I}(\mathbf{1}_{\uparrow})$ is interpreted by the bi-Scott domain $\mathbb{I}(\mathbf{D}_{\mathbf{1}_{\uparrow}})$. $\mathbf{rec}\ t.\sigma$ represents the solution of the domain equation

$$\mathbf{D}_t = \mathbf{D}_{\sigma(t)}$$

For example, $\mathbf{rec}t.t$ represents the bi-Scott domain $\mathbf{D_1}$ and therefore we could have avoided using $\mathbf{1}$ as a ground type. $\mathbf{rec}t.\mathbf{1} \oplus t$ represents the bi-Scott domain $\langle Nat, =\rangle$, i.e. the flat domain of natural numbers.

By using the above metalanguage we can provide denotational semantics for a large class of programming languages. Each programming language L is specified by a typed expression σ and each program in it is denoted by an element of $\mathbf{D}_\sigma$. We are not concerned how we select a typed expression for a particular programming language.

For each expression type σ we will introduce an assertion language L_σ based on a set of atomic information I_σ, such that there exists a bijection $\rho_{I_\sigma} : I_\sigma \longrightarrow \mathbf{D}^0_\sigma$. The assertions of L_σ are constructed in a similar way as σ is constructed from their subtypes.

Assertion Formation Rule

$(\mathbf{t}, \mathbf{f})$ $\quad \dfrac{}{\mathbf{t} : \sigma} \quad \dfrac{}{\mathbf{f} : \sigma}$ $\qquad$ $(Atomic)$ $\quad a : \sigma$ for each $a \in I_\sigma$

$(\wedge)$ $\quad \dfrac{\alpha, \beta : \sigma}{\alpha \wedge \beta : \sigma}$ $\qquad$ $(\vee)$ $\quad \dfrac{\alpha, \beta : \sigma}{\alpha \vee \beta : \sigma}$

$(\times)$ $\quad \dfrac{\alpha : \sigma \quad \beta : \tau}{(\alpha, \beta) : \sigma \times \tau}$ $\qquad$ $(\to)$ $\quad \dfrac{\alpha : \sigma \quad \beta : \tau}{\alpha \Rightarrow \beta : \sigma \to \tau}$

$(\oplus)$ $\quad \dfrac{\alpha : \sigma}{(\alpha, 0) : \sigma \oplus \tau} \quad \dfrac{\beta : \tau}{(\beta, 1) : \sigma \oplus \tau}$ $\qquad$ $((\ )^{op})$ $\quad \dfrac{\alpha : \sigma}{\alpha : \sigma^{op}}$

$((\ )_{\uparrow})$ $\quad \dfrac{\alpha : \sigma}{(\alpha, 0) : \sigma_{\uparrow}} \quad \dfrac{}{\perp : \sigma_{\uparrow}}$ $\qquad$ $(\mathbb{I}(\ ))$ $\quad \dfrac{\alpha, \beta : \sigma}{[\alpha, \beta] : \mathbb{I}(\sigma)}$

$(\Box)$ $\quad \dfrac{\alpha : \sigma}{\Box\alpha : \sigma^{S}}$ $\qquad$ $(\Diamond)$ $\quad \dfrac{\alpha : \sigma}{\Diamond\alpha : \sigma^{H}}$

(rec) $\quad \dfrac{\alpha : \sigma[rec\ X.\sigma/X]}{\alpha : rec\ X.\sigma}$

5 Proof system

In order to give axioms to the logic program, we will introduce a relation for every type σ. By $\alpha \vdash_\sigma \beta$, we mean that each object (program) satisfying α also satisfies β, i.e., each model of α is a model of β. Thus, from a logical stand point, when $\alpha \vdash_\sigma \beta$ we can say that α entail β in L_σ or, in other words, β is a logical consequence of α. For notational simplicity, we will eliminate subscripts if no confusion arise.

Logical Axioms and Logical Rules

($\mathbf{t}$) $\quad \dfrac{}{\alpha \vdash \mathbf{t}}$

($\mathbf{f}$) $\quad \dfrac{}{\mathbf{f} \vdash \alpha}$

(Ref) $\quad \dfrac{}{\alpha \vdash \alpha}$

($trans$) $\quad \dfrac{\alpha \vdash \beta \quad \beta \vdash \gamma}{\alpha \vdash \gamma}$

(Sym) $\quad \dfrac{\alpha \vdash \beta \quad \beta \vdash \alpha}{\alpha = \beta}$

($=$) $\quad \dfrac{\alpha = \beta}{\alpha \vdash \beta} \quad \dfrac{\alpha = \beta}{\beta \vdash \alpha}$

($\wedge_1$) $\quad \alpha \wedge \beta \vdash \alpha \quad \alpha \wedge \beta \vdash \beta$

($\wedge_2$) $\quad \dfrac{\alpha \vdash \beta \quad \alpha \vdash \gamma}{\alpha \vdash \beta \wedge \gamma}$

($\vee_1$) $\quad \alpha \vdash \alpha \vee \beta \quad \beta \vdash \alpha \vee \beta$

($\vee_2$) $\quad \dfrac{\alpha \vdash \gamma \quad \beta \vdash \gamma}{\alpha \vee \beta \vdash \gamma}$

With this proof systems we can prove, for example, that $\alpha \wedge \beta \vdash \alpha \vee \beta$. A proof of it is as follows:

$$\dfrac{\dfrac{}{\alpha \wedge \beta \vdash \alpha}\wedge_1 \qquad \dfrac{}{\alpha \vdash \alpha \vee \beta}\vee_1}{\alpha \wedge \beta \vdash \alpha \vee \beta}\,trans$$

We must introduce several logical axioms and rules in order to construct types.

Opposite

($\mathbf{t}^{op}$) $\quad \mathbf{f}_\sigma = \mathbf{t}_{\sigma^{op}}$

($\mathbf{f}^{op}$) $\quad \mathbf{t}_\sigma = \mathbf{f}_{\sigma^{op}}$

($\vdash^{op}$) $\quad \dfrac{\alpha \vdash_\sigma \beta}{\beta \vdash_{\sigma^{op}} \alpha}$

Intervals

$(\mathbb{I} - \mathbf{t})$ $\quad \mathbf{t}_{\mathbb{I}(\sigma)} = [\mathbf{f}_\sigma, \mathbf{t}_\sigma]$

$(\mathbb{I} - \vdash)$ $\quad \dfrac{\beta \vdash_\sigma \gamma \quad \alpha \vdash_\sigma^{op} \delta \quad \gamma \vdash_\sigma \delta}{[\alpha, \beta] \vdash_{\mathbb{I}(\sigma)} [\delta, \gamma]}$

$(\mathbb{I} - =_1)$ $\quad \dfrac{\beta \vdash_\sigma \alpha \quad \beta = \gamma}{[\alpha, \beta] = [\alpha, \gamma]} \qquad \dfrac{\gamma \vdash_\sigma \alpha \quad \beta = \alpha}{[\alpha, \gamma] = [\beta, \gamma]}$

$(\mathbb{I} - \wedge)$ $\quad [\alpha, \beta] \wedge [\gamma, \delta] = [\alpha \vee \gamma, \beta \wedge \delta]$

$(\mathbb{I} - \vee)$ $\quad [\alpha, \beta] \vee [\gamma, \delta] = [\alpha \wedge \gamma, \beta \vee \delta]$

Lifting

$(\uparrow - \mathbf{f})$ $\quad (\mathbf{f}_\sigma, 0) = \mathbf{f}_{\sigma_\uparrow}$

$(\uparrow - \mathbf{t})$ $\quad \perp = \mathbf{t}_{\sigma_\uparrow}$

$(\uparrow - \vdash)$ $\quad \dfrac{\alpha \vdash_\sigma \beta}{(\alpha, 0) \vdash_\uparrow (\beta, 0)}$

$(\uparrow - \vee)$ $\quad (\alpha \vee \beta, 0) = (\alpha, 0) \vee (\beta, 0)$

$(\uparrow - \wedge)$ $\quad (\alpha \wedge \beta, 0) = (\alpha, 0) \wedge (\beta, 0)$

Cartesian product

$(\times - \mathbf{t})$ $\quad (\mathbf{t}_\sigma, \mathbf{t}_\tau) = \mathbf{t}_{\sigma \times \tau}$

$(\times - \mathbf{f})$ $\quad (\mathbf{f}_\sigma, \alpha) = \mathbf{f}_{\sigma \times \tau} \qquad (\alpha, \mathbf{f}_\tau) = \mathbf{f}_{\sigma \times \tau}$

$(\times - \vdash)$ $\quad \dfrac{\alpha \vdash_\sigma \beta \quad \gamma \vdash_\tau \theta}{(\alpha, \gamma) \vdash_{\sigma \times \tau} (\beta, \theta)}$

$(\times - \vee)$ $\quad (\alpha \vee \beta, \gamma \vee \theta) = (\alpha, \gamma) \vee (\alpha, \theta) \vee (\beta, \gamma) \vee (\beta, \theta)$

$(\times - \wedge)$ $\quad (\alpha \wedge \beta, \gamma \wedge \theta) = (\alpha, \gamma) \wedge (\beta, \theta)$

Sum

$(\oplus - \mathbf{t})$ $\quad (\mathbf{t}_\sigma, 0) = \mathbf{t}_{\sigma\oplus\tau} \qquad (\mathbf{t}_\tau, 1) = \mathbf{t}_{\sigma\oplus\tau}$

$(\oplus - \mathbf{f})$ $\quad (\mathbf{f}_\sigma, 0) = \mathbf{f}_{\sigma\oplus\tau} \qquad (\mathbf{f}_\tau, 1) = \mathbf{f}_{\sigma\oplus\tau}$

$(\oplus - \vdash)$ $\quad \dfrac{\alpha \vdash_\sigma \beta}{(\alpha, 0) \vdash_{\sigma\times\tau} (\beta, 0)} \qquad \dfrac{\alpha \vdash_\tau \beta}{(\alpha, 1) \vdash_{\sigma\times\tau} (\beta, 1)}$

$(\oplus - \vee)$ $\quad (\alpha \vee \beta, 0) = (\alpha, 0) \vee (\alpha, 0) \qquad (\alpha \vee \beta, 1) = (\alpha, 1) \vee (\alpha, 1)$

$(\oplus - \wedge)$ $\quad (\alpha \wedge \beta, 0) = (\alpha, 0) \wedge (\beta, 0) \qquad (\alpha \wedge \beta, 1) = (\alpha, 1) \wedge (\beta, 1)$

bi-Continuous function space

$(\rightarrow -\mathbf{t})$ $\quad \alpha \Rightarrow \mathbf{t}_\tau = \mathbf{t}_{\sigma\rightarrow\tau}$

$(\rightarrow -\mathbf{f})$ $\quad \mathbf{f}_\sigma \Rightarrow \alpha = \mathbf{t}_{\sigma\rightarrow\tau}$

$(\rightarrow - \vdash\ )$ $\quad \dfrac{\alpha \vdash_\sigma \beta \quad \gamma \vdash_\tau \theta}{\beta \Rightarrow \gamma \vdash_{\sigma\rightarrow\tau} \alpha \Rightarrow \theta}$

$(\rightarrow -\wedge)$ $\quad \alpha \Rightarrow (\beta \wedge \gamma) = (\alpha \Rightarrow \beta) \wedge (\alpha \Rightarrow \gamma)$

$\rightarrow -\vee)$ $\quad (\alpha \vee \beta) \Rightarrow \gamma = (\alpha \Rightarrow \gamma) \wedge (\beta \Rightarrow \gamma)$

Hoare powerdomain

$(\Diamond - \mathbf{t})$ $\quad \Diamond\mathbf{t}_\sigma = \mathbf{t}_{\sigma^H}$

$(\Diamond - \mathbf{f})$ $\quad \Diamond\mathbf{f}_\sigma = \mathbf{f}_{\sigma^H}$

$(\Diamond - \vdash)$ $\quad \dfrac{\varphi \vdash_\sigma \psi}{\Diamond\varphi \vdash_{\sigma^H} \Diamond\psi}$

$(\Diamond - \vee)$ $\quad \Diamond(\varphi \vee \psi) = \Diamond\varphi \vee \Diamond\psi$

Smyth powerdomain

$(\Box - \mathbf{t})$ $\quad \Box\mathbf{t}_\sigma = \mathbf{t}_{\sigma^S}$

$(\Box - \mathbf{f})$ $\quad \Box\mathbf{f}_\sigma = \mathbf{f}_{\sigma^S}$

$(\Box - \vdash)$ $\quad \dfrac{\varphi \vdash_\sigma \psi}{\Box\varphi \vdash_{\sigma^S} \Box\psi}$

$(\Box - \wedge)$ $\quad \Box(\varphi \wedge \psi) = \Box\varphi \wedge \Box\psi$

That is, the Hoare and Smyth powerdomains can be seen as the modal operators $\Diamond$ and $\Box$, respectively.

6 Semantics

In order to better understand the sense by which this axiomatic presentation is equivalent to the usual denotational construction we define, for each closed type expression σ, an interpretation function

$$[\![\ \ ]\!]_\sigma : \mathcal{L}_\sigma \longrightarrow \Omega_S(\mathbf{D}_\sigma)$$

by

$[\![\mathbf{t}]\!]_\sigma = \mathbf{D}_\sigma$

$[\![\mathbf{f}]\!]_\sigma = \emptyset$

$[\![a]\!]_\sigma = \uparrow \rho_{I_\sigma}(a) = \{b \in D \ / \ \rho_{I_\sigma}(a) \leq_{\mathbf{D}_\sigma} b\}$ for each $a \in I_\sigma$

$[\![\alpha \wedge \beta]\!]_\sigma = [\![\alpha]\!]_\sigma \cap [\![\beta]\!]_\sigma$

$[\![\alpha \vee \beta]\!]_\sigma = [\![\alpha]\!]_\sigma \cup [\![\beta]\!]_\sigma$

$[\![a]\!]_{\sigma^{op}} = \{b \in D \ / \ b \leq_{\mathbf{D}_\sigma} \rho_{I_\sigma}(a)\}$ for each $a \in I_\sigma$

$[\![[\alpha,\beta]]\!]_{\mathbb{I}(\sigma)} = \{[\gamma,\theta] \ / \ \gamma \in [\![\alpha]\!]_\sigma$ and $\theta \in [\![\beta]\!]_{\sigma^{op}}$ and $\theta \vdash_\sigma \gamma\}$

$[\![(\alpha,0)]\!]_{\sigma_\uparrow} = ([\![\alpha]\!]_\sigma \times \{0\}) \cup \{\perp\}$

$[\![\perp]\!]_{\sigma_\uparrow} = D_{\sigma_\uparrow}$

$[\![(\alpha,\beta)]\!]_{\sigma\times\tau} = [\![\alpha]\!]_\sigma \times [\![\beta]\!]_\tau$

$[\![(\alpha,0)]\!]_{\sigma\oplus\tau} = [\![\alpha]\!]_\sigma$

$[\![(\beta,1)]\!]_{\sigma\oplus\tau} = [\![\beta]\!]_\tau$

$[\![\alpha \Rightarrow \beta]\!]_{\sigma\to\tau} = \{f : D_\sigma \longrightarrow D_\tau \ / \ f$ is bi-continuous and $[\![\alpha]\!]_\sigma \subseteq f^{-1}([\![\beta]\!]_\tau)\}$

$[\![\Box\alpha]\!]_{\sigma^S} = \{X \in \mathbf{D}_\sigma^S \ / \ X \subseteq [\![\alpha]\!]_\sigma\}$

$[\![\Diamond\alpha]\!]_{\sigma^H} = \{X \in \mathbf{D}_\sigma^H \ / \ X \cap [\![\alpha]\!]_\sigma \neq \emptyset\}$

$[\![\alpha]\!]_{\mathbf{rec}\ t.\sigma} = \{h_\sigma(\beta) \ / \ \beta \in [\![\alpha]\!]_{\sigma[\mathbf{rec}\ t.\sigma/t]}\}$

where $h_\sigma : \mathbf{D}_{\sigma[\mathbf{rec}\ t.\sigma/t]} \longrightarrow \mathbf{D}_{\mathbf{rec}\ t.\sigma}$ is an isomorphism.

Definition 6.1
Let α and β be assertions in L_σ. Then β is a consequence semantic of α in the interpretation $\mathbf{D}_\sigma$, denoted by $\alpha \models_{\mathbf{D}_\sigma} \beta$, if $[\![\alpha]\!]_\sigma \subseteq [\![\beta]\!]_\sigma$

Theorem 6.1 (Soundness and Completeness)
For each $\alpha, \beta \in L_\sigma$,

$$\alpha \vdash_\sigma \beta \text{ iff } \alpha \models_{\mathbf{D}_\sigma} \beta$$

7 Conclusions

The main particularity of bi-Scott domains is that it allows us to incorporate the interval constructor. This constructor is essential to deal with parametric intervals, i.e. interval of any other types. In this paper we introduced the bi-Scott logic domain, in order to develop a logic program which agrees with the denotational semantics of programming languages based on this category of domains. So, we provide a study, from a logical standpoint, of programming languages behavior that consider parametric interval data type.

Thus, analogously to [3, 6] with respect to real interval mathematics, this work joint with [9, 10] provide some aspects of a computational and logic foundation, from a theoretical domain point of view, of a generalized interval mathematics.

References

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

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

[3] Benedito Melo Acióly. *Computational Foundation of Interval Mathematics* (in portuguese). PhD thesis, CPGCC of UFRGS, Porto Alegre, 1991.

[4] Benjamín R. Callejas Bedregal and Benedito Melo Acióly. Rational Interval: An effectively given information system for the real numbers. *Scientific Computation and Mathematical Modelling*, S.M. Markov, editor. Sofia: DATECS publishing, p. 159-170, 1993.

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

[6] Benjamín R. Callejas Bedregal. *Continuous Information Systems: A Computational and Logical Approach to Interval Mathematics* (in portuguese). PhD thesis, UFPE-DI, Recife, 1996.

[7] Benjamín R. Callejas Bedregal and Benedito Melo Acióly. *Computability on the Interval Space: A domain approach.* In proceeding of XXII Conferencia Latinoamericana de Informática, Tomo I, p. 107-116. Valparaíso-Chile: 1997.

[8] Marcilia A. Campos. *An Interval Extension for a Real Probability*(in portuguese). PhD thesis, UFPE-DI, Recife, 1997.

[9] Roberto Callejas-Bedregal and Benjamín R. Callejas Bedregal. Intervals as a Domain Constructor. In *Tendências em Matemática Aplicada e Computacional (TEMA)*, E.X.L. de Andrade et al, editors, Vol. 2. p. 43-52. São Carlos: SBMAC, 2001.

[10] Roberto Callejas-Bedregal and Benjamín R. Callejas Bedregal. *Intervals as Categorical Constructors.* In 4^{th} Workshop on Formal Methods, Rio de Janeiro, october 2001.

[11] Roberto Callejas-Bedregal and Benjamín R. Callejas Bedregal. *bi-Scott Domains as a Cartesian Closed Category.* In International Symposium on Domain Theory, Chengdu-China, October, 2001.

[12] W.J. Cody. Floating-point standards, theory and practice. In Ramon E, Moore, editor, *Reliability in Computing. The Role of Interval Methods in Scientific Computing*, Vol 19 of *Perspectives in Computing*, p. 99-108, Academic Press, Inc., 1988.

[13] W. Cui and D. Blockey. Interval Probability Theory for Evidential Support. *International Journal of Intelligent Systems* V. 5, p. 183-192, 1990.

[14] Graçaliz P. Dimuro; Antônio C.R. Costa and Dalcidio M. Claudio. A Coherent Space of Rational Intervals for a Construction of $\mathbb{IR}$. In *Journal of Reliable Computing.* Dordrecht: V.6, N.2, p. 139-178, 2000.

[15] Abbas Edalat. *Domains for Computation in Mathematics, Physics and Exact Real Arithmetic.* In: Workshop and Summer School on new Paradigms for Computation on Classical Spaces, Birmingham, 1997.

[16] Abbas Edalat and Peter Sünderhauf. A Domain-Theoretic Approach to Computability on the Real Line. In *Theoretical Computer Science*, Netherland, Vol. 210, issue 1, pages 73-98, 1999.

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

[18] Carl A. Gunter and Dana S. Scott. Semantic Domains. In: *Handbook of Theoretical Computer Science, Volume B*, edited by J. Van Leeuwen, North Holland, pages 633-674, 1990.

[19] W.M. Kahan. Interval arithmetic options in the proposed IEEE floating point arithmetic standard. In Karl L.E. Nickel, editor, *Interval Mathematics*, pages 99-128, 1980.

[20] U.W. Kulisch and W.L. Miranker. Computer arithmetic in theory and practice. Technical Report 33658, IBM Thomas L. Watson Research Center, 1979.

[21] Aarão Lyra; Adrião D. Doria Neto and Benjamín R.C. Bedregal. An Interval Approach for the Digital Images Processing. In *Advances in Logic, Artificial Intelligence and Robotics*, Jair M. Abe and João I. da Silva Filho, editors (Proceeding of LAPTEC 2002). São Paulo: Editora Pleiade/SENAC, 2002.

[22] Ramon E. Moore. *Methods and Applications for Interval Analysis.* Philadelphia: SIAM, 1979. 190 p.

[23] Gordon Plotkin. *Domains.* Department of Computer Science, University of Edinburgh, 1983.

[24] D.S. Scott. *Outline of a Mathematical Theory of Computation.* In: 4^{th} Annual Princeton Conference on Information Sciences and Systems, pp. 169-176, 1970.

[25] Maria M.M.T. Silveira and Benjamín R.C. Bedregal. A Method of Inference and Defuzzification Fuzzy Interval. In: 2001 Artificial Intelligence and Application. Marbella: IASTED Press, 2001.

[26] Maria M.M.T. Silveira. *A Proposal of Integration of Interval Mathematics to Fuzzy Theory* (in portuguese). MSc thesis, PPgSC-UFRN, Natal-RN, 2002.

[27] Viggo Stoltenberg, Ingrid Lindström and Edward R. Griffer. *Mathematical Theory of Domains.* Cambridge University Press, Cambridge, 1994.

[28] Myke Smyth. Topology. In Handbook of Logic in Computer Science. Vol. 1. Clarendon Press - Oxford, 1992.

[29] I. Turksen. Interval value fuzzy sets based on normal form. *Fuzzy Sets and Systems* V.20, pp. 191-210. 1986.

[30] Guo Qiang Zhang. *A Logic for SFP.* Draft paper, University of Cambridge, 1987.

[31] Guo Qiang Zhang. *Logics of Domains.* Ph.D. Thesis, University of Cambridge, 1989.