ISSN 0105-8517

DI-Domains as Information Systems*

Guo Qiang Zhang

DAIMI PB – 282
August 1989

AARHUS UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
Ny Munkegade 116 – DK 8000 Aarhus C – DENMARK
Telephone: + 45 86 12 71 88 Telex: 64767 aausci dk

* An extended abstract of the paper appeared in the proceedings of ICALP 89.

Abstract

This paper introduces stable information systems. Stable information systems determine dI-domains and stable approximable mappings give stable functions, and vice versa. The notion of rigid embedding is captured by a subsystem relation. Under this relation, stable information systems form a cpo. Constructions like lifting, sum, product and function space are proposed which induce continuous functions on the cpo. In this way recursive stable information systems can be defined using fixed point theory.

1. Introduction

Stable information systems are a representation of dI-domains which are particular kinds of Scott domains discovered by Berry (Berry, 1978) from the study of the full-abstraction problem for typed λ-calculi. Stable information systems are logical structures with special kinds of entailment, and can be seen as a framework parallel to the stable event structures of Winskel (Winskel, 1988). Stable event structures cast light on the computational intuition of dI-domains while stable information systems cast light on the logical aspects of dI-domains.

The logical approach to domain theory uses some topological idea about computation. A topological space X can be taken as a 'data type', with the open sets as 'properties' and functions between topological spaces as 'computations' (Smyth, 1983). Such a topological idea is supported mathematically by a formal notion of Stone duality (see Johnstone, 1982). Complete Heyting algebras are complete lattices satisfying the infinite distributive law

$$a \wedge \bigvee S = \bigvee \{ a \wedge s \mid s \in S \}.$$

The category of *frames* has objects complete Heyting algebras, and morphisms functions which preserve finite meets and arbitrary joins. As a special kind of frames one has the set of open sets $\Omega(D)$ of a topological space D ordered by inclusion; in this case the frame morphisms are precisely those functions

$$f^{-1} : \mathbf{\Omega}(E) \to \mathbf{\Omega}(D)$$

for which $f : D \to E$ is continuous. The category of *locales* is the opposite of the category of frames where the morphism direction is reversed. *Stone dualities* are contravariant equivalences between certain categories of topological spaces and corresponding categories of locales (Johnstone, 1982). They have been proposed as providing the right framework for understanding the relationship between denotational semantics and program logic. Information systems can be seen as description of locales where the relevant topological spaces consist of the Scott open sets of domains. The *duality* between the category of information systems and the category of Scott topologies of domains is just the *equivalence* between the category of information systems and the category of domains.

Intuitively, an information system is a structure describing the logical relations among propositions that can be made about computations. It consists of a set of propositions, a consistency predicate and an entailment relation specified as follows (For convenience of getting a cpo of information systems we use a definition slightly different from the original one given in (Scott, 1982), without using a distinguished Δ standing for the proposition that is always true. The definition given below is the same as the one used in (Larsen and Winskel, 1984)).

Definition 1.1 An *information system* is a triple

$$\underline{A} = (\, A,\; Con,\; \vdash \,)$$

where

- A is a set of propositions
- $Con \subseteq \mathrm{Fin}(A)$, the consistent sets
- $\vdash \subseteq Con \times A$, the entailment relation

which satisfy

1. $X \subseteq Y \;\&\; Y \in Con \Longrightarrow X \in Con$
2. $a \in A \Longrightarrow \{a\} \in Con$
3. $X \vdash a \Longrightarrow X \cup \{a\} \in Con$
4. $a \in X \;\&\; X \in Con \Longrightarrow X \vdash a$
5. $(\forall b \in Y.\, X \vdash b \;\&\; Y \vdash c) \Longrightarrow X \vdash c$

Notation. We write $\mathrm{Fin}(A)$ for the set of finite subsets of A. Write $X \vdash Y$ to mean $\forall b \in Y.\ X \vdash b$; $X \dashv\vdash X'$ to mean $X \vdash X'$ and $X' \vdash X$; $X \subseteq^{fin} y$ to mean X is a finite subset of y.

Propositions are basic facts that can be affirmed about computations. They can be seen as units of information. *Con* contains all finite subsets of propositions that are non-contradictory, in a sense related to the computation under consideration. $X \vdash a$ can be roughly interpreted as: If the propositions in X are true of a computation, then a is also true of the computation.

An information system determines a family of subsets of propositions called its elements. Intuitively, an element consists of a set of propositions that can be truly made about a possible computation. Thus it is expected that the propositions should be in consistency with each other and, if a finite set of propositions is valid for a computation all the logical consequences should also be valid for it.

Definition 1.2 The *elements* $Pt(\underline{A})$, of an information system $\underline{A} = (A, Con, \vdash)$ consists of subsets x of propositions which are

1. finitely consistent: $X \subseteq^{fin} x \Longrightarrow X \in Con$,
2. closed under entailment: $X \subseteq x \;\&\; X \vdash a \Longrightarrow a \in x$.

For an information system $\underline{A}$, $(Pt\,\underline{A}, \subseteq)$ is a Scott domain (Scott, 1982). More generally, information systems form a category with the approximable mappings as morphisms, which is equivalent [Mac71] to the category of Scott domains. Constructions such as product, sum and function space have been proposed on information systems (Scott, 1982; Larsen and Winskel, 1984), corresponding to those on domains. Using information systems one can solve recursive equations concretely (Larsen and Winskel, 1984) with the resulting isomorphism being an equality.

Scott domains form a foundational framework for the denotational semantics of programming languages. There is another 'non-standard' framework of domains called dI-domains which were discovered by Berry (Berry, 1978) from the study of the full-abstraction problem for typed λ-calculi. They are special kinds of Scott domains which have a more operational nature. The functions between dI-domains are stable functions under an order which takes into account the manner in which they compute.

A dI-domain is a consistently complete cpo D which satisfies

- axiom d: $\forall x,\ y,\ z \in D.\ y \uparrow z \Longrightarrow x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$
- axiom I: $\forall d \in D^0.\ \mid \{ x \mid x \sqsubseteq d \} \mid < \infty$

where D^0 is the set of finite elements of D. Axiom d expresses the distributive property and axiom I says that D is *finitary.* A function f from a dI-domain D to a dI-domain E is stable if it is Scott-continuous and preserves meets of pairs of compatible elements, i.e.,

$$\forall x,\ y \in D.\ \ x \uparrow y \Longrightarrow f(x \sqcap y) = f(x) \sqcap f(y).$$

Let $f,\ g$ be in $[D \rightarrow_s E]$, the set of stable functions from D to E. f stably less than g, written $f \sqsubseteq_s g$, if

$$\forall x,\ y \in D.\ \ x \sqsubseteq y \Longrightarrow f(x) = f(y) \sqcap g(x).$$

DI-domains with stable functions form a cartesian closed category **DI** (Berry, 1978). The products are the cartesian product ordered coordinatewise and the function space consists of stable functions ordered under the stable order. These properties make dI-domains a nice alternative framework in which to do denotational semantics.

DI-domains can be represented as *stable event structures*, which are models for processes of concurrent computation. An *event structure* is a description of a set of events in terms of *consistency* and *enabling* relations. The consistency relation indicates whether some events can occur together or not, and the enabling relation specifies the condition when a particular event may occur with regards to the occurrence of other events.

A *configuration* of an event structure is a set of events which is consistent and each of its event is enabled by a set of events of the configuration occurred previously. Therefore, a configuration is a set of events which have occurred by certain stage in a process. More formally,

Definition 1.3 An *event structure* is a triple $\underline{E} = (E, \mathit{Con}, \vdash)$ where

- E is a countable set of events,
- Con is a non-empty subset of $\mathrm{Fin}(E)$, the finite subsets of E called the consistency predicate, which satisfies

$$X \subseteq Y \ \&\ Y \in \mathit{Con} \Longrightarrow X \in \mathit{Con},$$

- $\vdash \subseteq \mathit{Con} \times E$ is the enabling relation which satisfies

$$(X \vdash e \ \&\ X \subseteq Y \ \&\ Y \in \mathit{Con}) \Longrightarrow Y \vdash e.$$

When $X \vdash e$, we say e is enabled by X. Although an event structure looks similar to an information system, it is based on a different intuition and they are regarded in totally different ways. Typically, for an information system if $X \vdash a$ and $X \vdash a'$ then a and a' must be consistent propositions while for an event structure, we cannot say anything about the consistency of two events e, e' enabled by the same set of events. This reflects the fact that for information systems $\vdash$ stands for logical entailment between propositions whereas in the case of event structures it expresses when an event is enabled due to the previous occurrences of other events. Information systems capture the logical relations between facts about a computation while event structures capture their temporal relationship. Accordingly there is a different definition of configurations.

Definition 1.4 The *configurations* $\mathcal{F}(\underline{E})$, of an event structure $\underline{E} = (E, Con, \vdash)$ consists of subsets $x \subseteq E$ which are

- consistent: $\forall X \subseteq^{fin} x.\ X \in Con$,
- secured: $\forall e \in x\ \exists e_0, e_1, \cdots, e_n \in x.\ e_n = e\ \&$

$$\forall i \leq n.\ \{ e_k \mid 0 \leq k \leq i-1 \} \vdash e_i.$$

There is a special class of event structures for which each configuration determines a partial order of causal dependency on the events; intuitively, an event e_1 causally depends on an event e_0 if the occurrence of the event e_0 is *necessary* in order for e_1 to occur. Event structures of this kind are called *stable.*

Definition 1.5 An event structure $\underline{E}$ is stable if it satisfies the following axiom

$$(X \vdash e\ \&\ Y \vdash e\ \&\ X \cup Y \cup \{e\} \in Con) \Longrightarrow X \cap Y \vdash e.$$

When $\underline{E}$ is stable, $(\mathcal{F}(\underline{E}), \subseteq)$ is a dI-domain. Stable event structures with stable function on the set of configurations form a category which is equivalent to the category of dI-domains (Winskel, 1988).

2. Stable Information Systems

In this section we introduce stable information systems, which determine dI-domains. We also introduce approximable mappings as morphisms between stable information systems which give stable functions.

Definition 2.1 An information system $\underline{A} = (A,\ Con,\ \vdash)$ is called *stable* if it satisfies two extra axioms:

6. $X \vdash a \implies \exists b \in X.\ \{\, b \,\} \vdash a$

7. $\forall a \in A.\ \{\, b \mid \{\, a \,\} \vdash b \,\}$ is finite

Axiom 6 indicates that the entailment $\vdash$ is determined by a pre-order on A by letting $a \leq b$ iff $\{\, b \,\} \vdash a$. Thus stable information systems are similar to prime event structures (Nielsen, Plotkin and Winskel, 1981). Note, however, the entailment relation here determines a pre-order while for prime event structures the causal dependency relation is a partial order. Axiom 6 requires, in particular, that when $X \vdash a$ in a stable information system, X must be non-empty.

Axiom 7 corresponds to the axiom of finite cause for event structures. There is a strong computational intuition behind the axiom there[Wi86]. Another justification for axiom 7 is to get a cartesian closed category with A countable. Consider the stable functions from $(\omega \cup \{\perp\},\ \sqsubseteq)$ to itself, where $i \sqsubseteq j$ iff i is bigger than j. It can be shown that there are uncountably many finite elements in this function space (Zhang, 1989b). Therefore, dropping axiom 7 means we have to go beyond the countable, which is intuitively unwelcome as far as computation is concerned. Note that a choice is made here: we could have used an axiom which requires that there are only finitely many equivalent classes (under $\dashv\vdash$) in $\{\, b \mid \{\, a \,\} \vdash b \,\}$ rather than the whole set $\{\, b \mid \{\, a \,\} \vdash b \,\}$ be finite.

Theorem 2.1 Let $\underline{A} = (A,\ Con,\ \vdash)$ be a stable information system. $(Pt(\underline{A}),\ \subseteq)$ is a dI-domain.

It is convenient to work with another characterisation of dI-domains. Recall that a ***complete prime*** of a consistently complete partial order D is an element $p \in D$ such that

$$p \sqsubseteq \bigsqcup X \Longrightarrow \exists x \in X.\, p \sqsubseteq x$$

for all compatible X. D is ***prime algebraic*** if

$$x = \bigsqcup \{\, p \mid p \sqsubseteq x \;\&\; p \text{ is a complete prime} \,\}$$

for all $x \in D$. Suppose D is a consistently complete partial order which satisfies axiom I for dI-domains. It is a fact that D is prime algebraic iff it is a dI-domain (see Winskel, 1988 for a proof).

Proof of Theorem 2.1: Axiom 7 implies that $(\,Pt\,(\underline{A}),\ \subseteq)$ is finitary. For axiom d it is enough to show that $\{\, b \in A \mid \{\, a \,\} \vdash_{\underline{A}} b \,\}$ with $a \in A$ are the complete primes of $(\,Pt\,(\underline{A}),\ \subseteq)$, and $(\,Pt\,(\underline{A}),\ \subseteq)$ is prime algebraic. But these follow from axiom 6. ∎

Definition 2.2 Let $f : D \to E$ be a stable function, where D, E are dI-domains. Define μf to be a set of pairs such that $(a,\, p) \in \mu f$ if

$$f(a) \sqsupseteq p \;\&\; \forall a' \sqsubseteq a.\ f(a') \sqsupseteq p \Longrightarrow a = a',$$

where $a \in D^0$, the set of finite elements of D and $p \in E^p$, the set of complete primes of E.

The full abstraction problem for typed lambda-calculi lead Berry to consider the problem of capturing a notion of 'sequential functions'. As one of the possible candidates for 'sequential' functions Berry introduced stable functions so that non-sequential functions like 'parallel-or' are excluded. Stable functions have a property that their values are totaly determined by those at some minimal points. One can then understand a pair $(a, p) \in \mu f$ as saying a is a minimal point for f to assume value p.

The following two lemmas are useful. From the first lemma we know that the set μf fully determines a stable function f.

Lemma 2.1 Suppose $f : D \to E$ is a stable function. Then for any $x \in D$,

$$f(x) = \bigsqcup\{ p \mid \exists a \sqsubseteq x.\ (a,\ p) \in \mu f \}.$$

Proof Let $f : D \to E$ be a stable function. We have $f(a) \sqsupseteq p$ for any $(a, p) \in \mu f$. Therefore if $x \sqsupseteq a$ and $(a, p) \in \mu f$, then $f(x) \sqsupseteq p$. Hence

$$f(x) \sqsupseteq \bigsqcup\{ p \mid \exists a \sqsubseteq x.\ (a,\ p) \in \mu f \}.$$

On the other hand, it is easy to see that for any complete prime q in E such that $q \sqsubseteq f(x)$, there is an element $b \sqsubseteq x$ for which $(b, q) \in \mu f$. This means

$$q \sqsubseteq \bigsqcup\{ p \mid \exists a \sqsubseteq x.\ (a,\ p) \in \mu f \}.$$

But E is a dI-domain, hence prime algebraic. Therefore

$$f(x) = \bigsqcup\{ q \mid q \in E^p \ \&\ q \sqsubseteq f(x) \} \ \sqsubseteq\ \bigsqcup\{ p \mid \exists a \sqsubseteq x.\ (a,\ p) \in \mu f \}.$$

∎

The second lemma implies that compatible stable functions have the same minimal point related to a given value that they can both assume.

Lemma 2.2 Let $f, g : D \to E$ be stable functions. If $f \sqsubseteq_s g$ then

$$a \uparrow a' \;\&\; (a,\ p) \in \mu f \;\&\; (a',\ p) \in \mu g \Longrightarrow a = a'.$$

Proof Suppose $f, g : D \to E$ are stable functions, and suppose $a \uparrow a'$, $(a,\ p) \in \mu f$, and $(a',\ p) \in \mu g$. Together with $f \sqsubseteq_s g$ it follows that

$$g(a) \sqsupseteq f(a) \sqsupseteq p.$$

Therefore

$$g(a \sqcap a') = g(a) \sqcap g(a') \sqsupseteq p.$$

But $(a',\ p) \in \mu g$. We must have $a \sqcap a' = a'$, and hence $a \sqsupseteq a'$. Now

$$f(a \sqcap a') = g(a \sqcap a') \sqcap f(a) \sqsupseteq p,$$

which implies $a \sqcap a' = a$ since $(a,\ p) \in \mu f$. Thus we also have $a' \sqsupseteq a$. Hence $a = a'$. ∎

Stable information systems can be equipped with approximable mappings similar to information systems. But that does not lead to a characterisation of stable order. To get some guidance, we present the following fact about stable functions, which is a generalisation of Proposition 8.2.3 in (Zhang, 1989b).

Theorem 2.2 Let $f, g \in [D \to_s E]$. $f \sqsubseteq_s g$ iff $\mu f \subseteq \mu g$. For $\{\, (a_i,\ b_i) \mid i \in I \,\} \subseteq D^0 \times E^p$,

$$\{\, (a_i,\ b_i) \mid i \in I \,\} = \mu f$$

for some $f \in [D \to_s E]$ iff

- $\forall J \subseteq^{fin} I. \{ a_i \mid i \in J \}\uparrow \Longrightarrow \{ b_i \mid i \in J \}\uparrow$,
- $a_i \uparrow a_j \,\&\, (b_i = b_j) \implies (a_i = a_j)$,
- $\forall b \in E^p.\, b_i \sqsupseteq b \implies \exists j.\, b_j = b \,\&\, a_i \sqsupseteq a_j$.

Proof Suppose $f, g : D \to E$ are stable functions such that $f \sqsubseteq_s g$. For any $(a, p) \in \mu f$, $p \sqsubseteq f(a) \sqsubseteq g(a)$. Let

$$b = \bigsqcap \{ x \mid x \sqsubseteq a \,\&\, g(x) \sqsupseteq p \}.$$

Clearly $(b, p) \in \mu g$ and $b \sqsubseteq a$. By Lemma 2.2, $a = b$. Hence $(a, p) \in \mu g$ and $\mu f \subseteq \mu g$.

Suppose, on the other hand, that $\mu f \subseteq \mu g$. It follows from Lemma 2.1 that $\forall x \in D.\, f(x) \sqsubseteq g(x)$. To prove $f \sqsubseteq_s g$ we have to show that for $x \sqsubseteq y$ in D,

$$f(x) = f(y) \sqcap g(x).$$

To this end let $p \in E^p$ and $p \sqsubseteq f(y) \sqcap g(x)$. Clearly there exist $a \sqsubseteq x$ and $b \sqsubseteq y$ such that $(b, p) \in \mu f$ and $(a, p) \in \mu g$. However $\mu f \subseteq \mu g$; we have $(b, p) \in \mu g$. By Lemma 2.2, $a = b$. This implies that $f(x) \sqsupseteq f(b) \sqsupseteq p$. By the prime algebraicness of E we get

$$f(x) \sqsupseteq f(y) \sqcap g(x),$$

enough for the equation

$$f(x) = f(y) \sqcap g(x)$$

to hold.

Now we prove the second part of Theorem 2.2.

Suppose $\{(a_i, b_i) \mid i \in I\} = \mu f$ for some stable function f. It is rountine to check that the three properties mentioned in Theorem 2.2 hold.

Let $\{(a_i, b_i) \mid i \in I\} \subseteq D^0 \times E^p$ be a set with the three properties. We show that the stable function f for which (Lemma 2.1 concludes that such a function is unique)

$$\{(a_i, b_i) \mid i \in I\} = \mu f$$

can be obtained as pointwise lubs $\bigsqcup_{i \in I} [a_i, b_i]$ where

$$[a, b](x) = \begin{cases} b & \text{if } x \sqsupseteq a\ , \\ \perp & \text{otherwise.} \end{cases}$$

Obviously $\bigsqcup_{i \in I} [a_i, b_i]$ is continuous. To check stability let $x, y \in D$ and $x \uparrow y$. Suppose

$$p \sqsubseteq \bigsqcup_{i \in I} [a_i, b_i](x) \sqcap \bigsqcup_{i \in I} [a_i, b_i](y)$$

where $p \in E$ is a complete prime. We have, for some i, j, $p \sqsubseteq b_i$, $a_i \sqsubseteq x$ and $p \sqsubseteq b_j$, $a_i \sqsubseteq y$. By the third property, there exist s, t such that $b_s = p$, $a_s \sqsubseteq a_i$ and $b_t = p$, $a_t \sqsubseteq a_j$. Therefore $a_s = a_t$ as $a_s \uparrow a_t$ and $b_s = b_t$. We now have $a_s = a_t \sqsubseteq x \sqcap y$ and

$$p \sqsubseteq \bigsqcup_{i \in I} [a_i, b_i](x \sqcap y).$$

Since E is prime algebraic,

$$\bigsqcup_{i \in I} [a_i, b_i](x \sqcap y) \sqsupseteq \bigsqcup_{i \in I} [a_i, b_i](x) \sqcap \bigsqcup_{i \in I} [a_i, b_i](y).$$

This implies that $\bigsqcup_{i\in I}[\,a_i, b_i\,]$ is stable. It remains to show that

$$\{\ (a_i,\ b_i) \mid i \in I\ \} = \mu f$$

where we abbreviate $\bigsqcup_{i\in I}[\,a_i, b_i\,]$ as f. We have

$$\begin{aligned} f(a_j) &= \bigsqcup\{\ b_i \mid a_i \sqsubseteq a_j\ \} \\ &\sqsupseteq b_j. \end{aligned}$$

Let $y \sqsubseteq a_j$ and $f(y) \sqsupseteq b_j$, i.e. $\bigsqcup\{\ b_i \mid a_i \sqsubseteq y\ \} \sqsupseteq b_j$. Since b_j is a complete prime, $b_i \sqsupseteq b_j$ for some i with $a_i \sqsubseteq y$. The third condition mentioned in Theorem 2.2 implies the existence of some k such that $b_k = b_j$ and $a_k \sqsubseteq a_i$. But $a_k = a_j$ since $a_k \uparrow a_j$. Hence $y = a_j$. This means $(a_j,\ b_j) \in \mu f$.

For any $(a,\ p) \in \mu f$, we have $f(a) \sqsupseteq p$. Therefore

$$\bigsqcup\{\ b_i \mid a_i \sqsubseteq a\ \} \sqsupseteq p.$$

Since p is a complete prime, there is some b_i such that $b_i \sqsupseteq p$. By the third condition mentioned in Theorem 2.2 again, $b_j = p$ form some j such that $a_j \sqsubseteq a_i$. By the result from the previous paragraph we have $(a_j,\ b_j) \in \mu f$. Therefore $a_j = a$ by Lemma 2.2 (taking $f = g$). ∎

When $\{\ (a_i, b_i)\ \mid\ i \in I\ \} \subseteq D^0 \times E^p$ satisfies the three conditions above, we call

$$\{\ (\,a_i, b_i\,) \mid i \in I\ \}$$

stable joinable. Functions of the form $\bigsqcup_{i\in I}[\,a_i, b_i\,]$ with I finite and $\{\ (\,a_i, b_i\,) \mid i \in I\ \}$ stable joinable are called *step functions.*

Now we come naturally to

Definition 2.3 Let $\underline{A} = (A,\ Con_{\underline{A}},\ \vdash_{\underline{A}})$, $\underline{B} = (B,\ Con_{\underline{B}},\ \vdash_{\underline{B}})$ be stable information systems. A stable approximable mapping $R : \underline{A} \rightarrow \underline{B}$ is a relation $R \subseteq Con_{\underline{A}} \times B$ which satisfies, with I finite,

1. $(X \cup X' \in Con_{\underline{A}}\ \&\ X\,R\,c\ \&\ X'\,R\,c'\ \&\ \{c\} \dashv\vdash_{\underline{B}} \{c'\}) \Longrightarrow X \dashv\vdash_{\underline{A}} X'$
2. $(\forall i \in I\,.\ X_i\,R\,b_i)\ \&\ \bigcup X_i \in Con_{\underline{A}} \Longrightarrow \{\, b_i \mid i \in I \,\} \in Con_{\underline{B}}$
3. $X\,R\,b\ \&\ \{b\} \vdash_{\underline{B}} c \Longrightarrow \exists X'\,.\ X \vdash_{\underline{A}} X'\ \&\ X'\,R\,c$
4. $X\,R\,b\ \&\ X \dashv\vdash_{\underline{A}} X' \Longrightarrow X'\,R\,b$

These conditions are similar to those used in Theorem 2.2.

The first condition expresses the *minimal* property. $X\,R\,b$ can be read as: X in $\underline{A}$ entails b in $\underline{B}$, and, moreover, X is a weakest one in $\underline{A}$. For example, for any stable approximable mapping R, we cannot have $\emptyset\,R\,b$ and $X\,R\,b$ for some non-empty X at the same time. Assume there were such a b. Then we must have $\emptyset \vdash X$, contradicting axiom 6 for stable information systems.

The second condition means *consistency*. The third condition insists on *completeness*, in the sense that when X is a weakest proposition for b, all the propositions weaker than b must also have their weakest propositions specified. The last condition requires R to be *maximal*, which brings some technical advantages.

It is clear that because of the minimal property, axioms like

$$X \vdash_B Y\ \&\ Y\,R\,b\ \&\ \{b\} \vdash_A c \Longrightarrow X\,R\,c$$

from approximable mappings of information systems should be abandoned for stable information systems.

Proposition 2.1 Suppose $R : \underline{A} \to \underline{B}$ is a stable approximable mapping. Then the function $Pt\,(R) : Pt(\underline{A}) \to Pt(\underline{B})$ specified by

$$Pt\,(R)(x) = \{\, b \mid \exists X \subseteq x.\, X\, R\, b \,\}$$

is stable.

Proof First we check $Pt\,(\,R\,)$ is well defined.

For any element $x \in Pt\underline{A}$, $Pt\,(\,R\,)(x)$ is finitely consistent. Suppose

$$\{\, b_0, b_1, \cdots, b_n \,\} \subseteq Pt\,(\,R\,)(x).$$

There exist $X_0, X_1, \cdots, X_n$, such that

$$\forall i\,.\,(\, X_i \subseteq x \;\&\; X_i\, R\, b_i \,).$$

By axiom 2 of Definition 2.3 we have

$$\{\, b_0, b_1, \cdots, b_n \,\} \in \mathit{Con}.$$

$Pt\,(\,R\,)(x)$ is also closed under entailment. Assume $Y \subseteq Pt\,(\,R\,)(x)$ and $Y \vdash c$. We know that $\exists b \in Y\,.\,\{\,b\,\} \vdash c$. Since $b \in Pt\,(\,R\,)(x)$, there exists an $X \subseteq x$ such that $X\, R\, b$. By axiom 3 of Definition 2.3, $X'\, R\, c$ for some non-empty X' such that $X \vdash X'$, which implies $c \in Pt\,(\,R\,)(x)$ as $X' \subseteq x$; or $X = \emptyset$ and $\emptyset\, R\, c$ which also implies $c \in Pt\,(\,R\,)(x)$. So $Pt\,(\,R\,)(x)$ is an element of $Pt\underline{B}$.

It is routine to check that $Pt\,(\,R\,)$ is continuous. To check stability assume $x \uparrow y$ with $x, y \in Pt\underline{A}$. Assume also that $d \in Pt\,R(x) \cap Pt\,R(y)$.

Thus $\exists X \subseteq x, X' \subseteq y$ such that $X\,R\,d$ and $X'\,R\,d$. Since x and y are compatible, $X \cup X'$ must be consistent. Therefore, by axiom 1 of Definition 2.3, $X \dashv\vdash X'$, implying $d \in Pt\,R(x \cap y)$. We have proved that $Pt\,R(x) \cap Pt\,R(y) \subseteq Pt\,R(x \cap y)$. The other direction of the inclusion follows from the monotonicity of $Pt\,R$. ∎

To get strict functions we can simply restrict the relation $X\,R\,a$ by requiring X to be non-empty.

The following proposition says that set inclusion on the stable approximable mappings determines the stable order.

Proposition 2.2 Let $\underline{A}$ and $\underline{B}$ be stable information systems, and $R,\ S : \underline{A} \rightarrow \underline{B}$ stable approximable mappings. $R \subseteq S$ iff $Pt(R) \sqsubseteq_s Pt(S)$, where $\sqsubseteq_s$ is the stable order.

Proof Only if: Assume $R \subseteq Q$ and let $x,\ y \in Pt(\underline{A})$, $x \subseteq y$ and $b \in f_R(y) \cap f_S(x)$. There must be $X \subseteq x$ and $Y \subseteq y$ such that $X\,R\,b$ and $Y\,S\,b$. By a similar argument used in the previous proposition we know that

$$Pt\,R(x) \supseteq Pt\,R(y) \cap Pt\,S(x).$$

The other direction of the inclusion follows from monotonicity.

If: Suppose $Pt(R) \sqsubseteq_s Pt(S)$. For any X and a, if $X\,R\,a$ then $\overline{X} \in Pt(\underline{A})$, where $\overline{X} =^{\text{def}} \{\, b \mid X \vdash b \,\}$. We have

$$Pt\,R(\overline{X}) \subseteq Pt\,S(\overline{X}).$$

$a \in Pt\,S(\overline{X})$ since $a \in Pt\,R(\overline{X})$. Therefore $Y\,S\,a$ for some $Y \subseteq \overline{X}$. Clearly

$\overline{Y} \subseteq \overline{X}$. By the stable order we get

$$Pt\,R(\overline{Y}) = Pt\,R(\overline{X}) \;\cap\; Pt\,S(\overline{Y}).$$

It is easy to deduce $a \in Pt\,R(\overline{X}) \cap Pt\,S(\overline{Y})$ since $X\,R\,a$ and $Y\,S\,a$. This implies $a \in Pt\,R(\overline{Y})$. For some $Z \subseteq \overline{Y}$, therefore, $Z\,R\,a$. By the first axiom of Definition 2.3 we have $Z \dashv\vdash X$, which implies $X \dashv\vdash Y$. And by the fourth axiom we have $X\,S\,a$. Hence $R \subseteq S$. ∎

It is intended that stable approximable mappings are morphisms on stable information systems so that one gets a category. But this is far from clear at this stage: the identity Id should be given by $X\,Id\,b$ iff $X \dashv\vdash \{b\}$. How can we compose Id, the identity stable approximable mapping with other R?

Let $R : \underline{A} \to \underline{B}$ be a stable approximable mapping. Define $\tilde{R}$ to be a relation on $Con_{\underline{A}} \times Con_{\underline{B}}$ such that $X\,R\,Y$ iff there exist X_i, b_i such that $X_i\,R\,b_i$, $1 \le i \le n$, $X = \bigcup_{1 \le i \le n} X_i \in Con_{\underline{A}}$, and $Y = \{\, b_i \mid 1 \le i \le n \,\}$. It is not difficult to check that $\tilde{R}$ has similar properties to that of stable approximable mappings, including

1. $(X \cup X' \in Con_{\underline{A}} \;\&\; X\,\tilde{R}\,Y \;\&\; X'\,\tilde{R}\,Y' \;\&\; Y \dashv\vdash_{\underline{B}} Y') \Longrightarrow X \dashv\vdash_{\underline{A}} X'$
2. $(\forall i \in I\,.\, X_i\,\tilde{R}\,Y_i) \;\&\; \bigcup X_i \in Con_{\underline{A}} \Longrightarrow \bigcup Y_i \in Con_{\underline{B}}$
3. $X\,\tilde{R}\,Y \;\&\; Y \vdash_{\underline{B}} Y' \Longrightarrow \exists X'\,.\, X \vdash_{\underline{A}} X' \;\&\; X'\,\tilde{R}\,Y'$

Now we can compose stable approximable mappings $R : \underline{A} \to \underline{B}$ and $S : \underline{B} \to \underline{C}$. Define $R \circ S$ to be a relation on $Con_{\underline{A}} \times C$ by letting

$$X\,(R \circ S)\,c \Longleftrightarrow \exists Y \in Con_{\underline{B}}.\; X\,\tilde{R}\,Y \;\&\; Y\,S\,c.$$

From the property of $\tilde{R}$ one can see that $R \circ S : \underline{A} \to \underline{C}$ is a stable approximable mapping and further, by inspecting the axioms for a category, we have

Theorem 2.3 Stable information systems with stable approximable mappings form a category **SIS**.

So far this is only one side of the story: We can get dI-domains from stable information systems. It is also possible to get stable information systems from dI-domains.

Definition 2.4 Let D be a dI-domain. Define

$$\mathit{SI}\, D = (\, A,\, \mathit{Con},\, \vdash),$$

by taking

- $A = \{\, p\!\uparrow \mid p \in D^p \,\}$
- $X \in \mathit{Con} \Longleftrightarrow \bigcap X \neq \emptyset$
- $X \vdash a \Longleftrightarrow \bigcap X \subseteq a$

with $p\!\uparrow = \{\, d \in D \mid p \sqsubseteq d \,\}$.

Following standard convention, let $\bigcap X = D$ when X is empty. Note $\bot$ is *not* a complete prime.

Proposition 2.3 If D is a dI-domain then $\mathit{SI}\,(D)$ is a stable information system.

Proof By inspecting all the axioms for stable information systems.
∎

We can also get stable approximable mappings from stable functions.

Definition 2.5 Let D, E be dI-domains, and $f : D \rightarrow E$ a stable function. Define a relation $SI(f) \subseteq Con_{SI(D)} \times A_{SI(E)}$ by taking

$$X\ SI(f)\ a \iff (x, p) \in \mu f$$

provided $\bigcap X = x\uparrow$ and $a = p\uparrow$.

Note $(x, p) \in \mu f$ requires, by Definition 2.2, that x is a finite element of D and p is a complete prime of E.

Proposition 2.4 Let D, E be dI-domains, and $f : D \rightarrow E$ a stable function. Then $SI(f)$ is a stable approximable mapping from $SI(D)$ to $SI(E)$.

Proof We check that axioms 1, 2, 3 and 4 of Definition 2.3 hold for $SI\, f$.

Axiom 1 follows from Lemma 2.2 by taking $f = g$.

Axiom 2 and 4 are easy.

Axiom 3 follows from the third property mentioned in Theorem 2.2 about the set μf. ∎

Note that for the stable information system $SI\, D$ determined by a dI-domain D,

$$\{a\} \dashv\vdash \{b\} \Longrightarrow a = b.$$

We conclude this section by

Theorem 2.4 *Pt* : **SIS** $\rightarrow$ **DI** and *SI* : **DI** $\rightarrow$ **SIS** are functors which determine an equivalence of **SIS** and **DI**.

Proof That Pt and SI are functors is routine.

We use one of MacLane's results in (Maclane, 1971). It is enough to show that Pt is full and faithful, and each dI-domain D is isomorphic to $Pt\,(\underline{A})$ for some stable information system $\underline{A}$. The latter is straightforward. It remains to show that Pt is full and faithful.

First we show that Pt is full. Suppose $\underline{A}$ and $\underline{B}$ are stable information systems and

$$f : Pt\,(\underline{A}) \rightarrow Pt\,(\underline{B})$$

a stable function. Define a relation $R \subseteq Con_{\underline{A}} \times B$ by letting $X\,R\,b$ if $(\overline{X},\ \overline{b}) \in \mu f$. It follows from Proposition 2.4 that this relation is an approximable mapping form $\underline{A}$ to $\underline{B}$. By Theorem 2.2, the stable function $Pt\,R$ determined by R is actually equal to f.

Suppose $R,\ S :\ \underline{A} \rightarrow \underline{B}$ are approximable mappings such that $Pt\,R = Pt\,S$. It follows from Proposition 2.2 that $R \subseteq S$ and $S \subseteq R$, and hence $R = S$. Therefore Pt is faithful. ∎

3. A Cpo of Stable Information Systems

In this section we introduce a subsystem relation on stable information systems. The subsystem relation captures the notion of rigid embedding (Kahn and Plotkin, 1978). We get a cpo with the subsystem relation, which enables us to give meaning to recursively defined stable information systems through the construction of least fixed points for continuous functions.

Definition 3.1 Let $\underline{A} = (A, Con_{\underline{A}}, \vdash_{\underline{A}})$ and $\underline{B} = (B, Con_{\underline{B}}, \vdash_{\underline{B}})$ be stable information systems. $\underline{A} \trianglelefteq \underline{B}$ if

1. $A \subseteq B$
2. $X \in Con_{\underline{A}} \iff X \subseteq A \;\&\; X \in Con_{\underline{B}}$
3. $X \vdash_{\underline{A}} a \iff X \subseteq A \;\&\; X \vdash_{\underline{B}} a$

When $\underline{A} \trianglelefteq \underline{B}$ we call $\underline{A}$ a subsystem of $\underline{B}$. Note condition 3 above implies that

$$X \vdash_{\underline{B}} a \;\&\; X \subseteq A \Longrightarrow a \in A.$$

Hence our definition of subsystem is different from (Larsen and Winskel, 1984), where the relation captures the notion of embedding between domains. We have a stronger notion of subsystems (This is not surprising at all because rigid embeddings are embeddings but not vice versa).

Let $\underline{A} = (A, Con_{\underline{A}}, \vdash_{\underline{A}})$ and $\underline{B} = (B, Con_{\underline{B}}, \vdash_{\underline{B}})$ be stable information systems. If $A = B$ and $\underline{A} \trianglelefteq \underline{B}$, then $\underline{A} = \underline{B}$.

Definition 3.2 Let D, E be dI-domains. A stable function $f : D \to E$ is a ***rigid embedding*** if there is a stable function $g : E \to D$ called a *projection* such that

- $\forall d \in D.\; gf(d) = d$
- $\forall e \in E.\; fg(e) \sqsubseteq e$
- $\forall d \in D, e \in E.\; e \sqsubseteq f(d) \Longrightarrow fg(e) = e$

Proposition 3.1 Let $\underline{A} = (A, Con_{\underline{A}}, \vdash_{\underline{A}})$ and $\underline{B}$ be stable information systems. If $\underline{A} \trianglelefteq \underline{B}$ then the inclusion map $i : Pt\underline{A} \to Pt\underline{B}$ is a rigid embedding with projection $j : Pt\underline{B} \to Pt\underline{A}$ given by $j(y) = y \cap A$ for $y \in Pt\underline{B}$.

Proof We have

$$\forall x \in Pt\underline{A}.\ ji(x) = x \cap A = x,$$

$$\forall y \in Pt\underline{B}.\ ij(y) = y \cap A \subseteq y, \text{ and}$$

$$\forall y \in Pt\underline{B}.\ y \subseteq A \Rightarrow ij(y) = y.$$

Hence it is enough to show that i , j are well defined functions, which is trivial. ∎

The relation $\trianglelefteq$ is almost a complete partial order on stable information systems. Clearly there is a least stable information system, with the empty set as propositions. The limit of an ω-increasing chain is a stable information system with the proposition set, consistency and entailment relations the union of those in the chain. We have

Theorem 3.1 The relation $\trianglelefteq$ is a partial order with the least element

$$\underline{\perp} = (\emptyset, \{\emptyset\}, \emptyset).$$

If $\underline{A_0} \trianglelefteq \underline{A_1} \trianglelefteq \cdots \trianglelefteq \underline{A_i} \trianglelefteq \cdots$ is an increasing chain of stable information systems where $\underline{A_i} = (A_i, Con_i, \vdash_i)$, then their least upper bound is

$$\bigcup_i \underline{A_i} = \left(\bigcup_i A_i, \bigcup_i \vdash_i, \bigcup_i Con_i \right).$$

Proof As our notion of subsystem is stronger than the one in (Larsen and Winskel, 1984) we know that

$$\bigcup_i \underline{A_i} = \left(\bigcup_i A_i, \bigcup_i \vdash_i, \bigcup_i Con_i \right)$$

is the least upper bound of the chain as information systems. We check that axiom 6 and axiom 7 of Definition 2.1 hold for $\bigcup_i \underline{A_i}$. Suppose $X \vdash a$ in $\bigcup_i \underline{A_i}$. Since the entailment is the union of those of $\underline{A_i}$'s and X is finite, there must be some k such that $X \vdash_k a$. But $\underline{A_k}$ is stable. Therefore $\exists b \in X. \{ b \} \vdash_k a$ which implies $\{ b \} \vdash a$ in $\bigcup_i \underline{A_i}$. To see axiom 7 holds consider $\{ b \mid \{ a \} \vdash b \}$ in $\bigcup_i \underline{A_i}$. Obviously $a \in A_k$ for some k. We show that

$$\{ b \mid \{ a \} \vdash b \} \subseteq \{ b \mid \{ a \} \vdash_k b \},$$

which implies the finiteness of $\{ b \mid \{ a \} \vdash b \}$. Assume $\{ a \} \vdash_j t$ for $t \in A_j$. If $j \leq k$ then $\{ a \} \vdash_k t$ as $\underline{A_j} \trianglelefteq \underline{A_k}$. If $j \geq k$ then $\underline{A_k} \trianglelefteq \underline{A_j}$. By axiom 3 of Definition 3.1 $\{ a \} \vdash_k t$ since $a \in A_k$. Therefore axiom 7 holds for $\bigcup_i \underline{A_i}$.

That for each j $\underline{A_j} \trianglelefteq \bigcup_i \underline{A_i}$ is trivial. ∎

Write $\mathbf{CPO}_{\mathrm{sis}}$ for the 'cpo' of stable information systems under $\trianglelefteq$. $\mathbf{CPO}_{\mathrm{sis}}$ is not a cpo in the usual sense simply because they are not a set but a class. However, this 'large' cpo still suits our purpose.

The subsystem relation $\trianglelefteq$ can be easily extended to $n-$tuples coordinatewisely. More precisely we require

$$(\underline{A_1}, \underline{A_2}, \cdots \underline{A_n}) \trianglelefteq (\underline{B_1}, \underline{B_2}, \cdots \underline{B_n})$$

iff for each $1 \leq i \leq n$, $\underline{A_i} \trianglelefteq \underline{B_i}$. For convenience write $\vec{\underline{A}}$ for $(\underline{A_1}, \underline{A_2}, \cdots \underline{A_n})$.

The least upper bound of an $\omega-$chain of $n-$tuples of stable information systems is then just the $n-$tuple of stable information systems consisting of the least upper bounds on each component, i.e. if

$$\underline{\vec{A}}_1 \trianglelefteq \underline{\vec{A}}_2 \cdots \trianglelefteq \underline{\vec{A}}_i \trianglelefteq \cdots$$

then for the j-th component

$$\pi_j \left(\bigcup_i \underline{\vec{A}}_i \right) = \bigcup_i \pi_j \left(\underline{\vec{A}}_i \right).$$

An operation F form $n-$tuples of stable information systems to $m-$tuples of stable information systems is said to be continuous iff it is monotonic, i.e. $\underline{\vec{A}} \trianglelefteq \underline{\vec{B}}$ implies $F(\underline{\vec{A}}) \trianglelefteq F(\underline{\vec{B}})$ and preserves $\omega-$increasing chains of stable information systems, i.e.

$$\underline{\vec{A}}_1 \trianglelefteq \underline{\vec{A}}_2 \cdots \trianglelefteq \underline{\vec{A}}_i \trianglelefteq \cdots$$

implies

$$\bigcup_i F(\underline{\vec{A}}_i) = F(\bigcup_i \underline{\vec{A}}_i).$$

It is well known that for functions on (finite) tuples of cpos they are continuous iff by changing (any) one argument while fixing others the induced function is continuous.

Proposition 3.2 An unary operation F is continuous iff it is monotonic with respect to $\trianglelefteq$ and continuous on proposition sets, i.e. for any $\omega-$chain

$$\underline{A}_1 \trianglelefteq \underline{A}_2 \cdots \trianglelefteq \underline{A}_i \trianglelefteq \cdots,$$

each proposition of $F(\bigcup_i \underline{A}_i)$ is a proposition of $\bigcup_i F(\underline{A}_i)$.

Proof The 'only if' part is trivial.

If: Let

$$\underline{A}_1 \trianglelefteq \underline{A}_2 \cdots \trianglelefteq \underline{A}_i \trianglelefteq \cdots$$

be an $\omega-$chain of stable information systems. Since F is monotonic, we clearly have

$$\bigcup_i F(\underline{A}_i) \trianglelefteq F(\bigcup_i \underline{A}_i).$$

Thus the propositions of $F(\bigcup_i \underline{A}_i)$ are the same as proposition of $\bigcup_i F(\underline{A}_i)$. Therefore they are the same stable information systems by the remark given just before Definition 3.2. ∎

Now given any continuous function F on $\mathbf{CPO}_{\text{sis}}$, we can get the least fixed point of F, which is the limit of the increasing $\omega-$chain

$$\underline{\perp} \trianglelefteq F(\underline{\perp}) \trianglelefteq F^2(\underline{\perp}) \trianglelefteq \cdots \trianglelefteq F^n(\underline{\perp}) \trianglelefteq \cdots,$$

i.e. $\bigcup_i F^i(\underline{\perp})$. Note since we are working with a partial order, we get an equality

$$F(\bigcup_i F^i(\underline{\perp})) = \bigcup_i F^i(\underline{\perp}).$$

4. Constructions

In this section we introduce constructions of lifting $(\ )_\uparrow$, sum $+$, product $\times$ and function space $\rightarrow$ on stable information systems. These constructions have their counterparts in dI-domains as the constructions of lifting, sum, product and stable function space. They induce continuous functions on $\mathbf{CPO}_{\text{sis}}$. In this way we can produce solutions to recursive equations for stable information system written in these constructions.

Lifting, sum and product are more or less the same as those on information systems (Larsen and Winskel, 1984). There is a minor technical advantage because axiom 6 for stable information system rules out the possibility $\emptyset \vdash a$. What is totally novel is the construction of function space.

Definition 4.1 (Lifting) Let $\underline{A} = (A, \; Con, \; \vdash)$ be a stable information system. Define the *lift* of $\underline{A}$ to be $\underline{A}_{\uparrow} = (A', \; Con', \; \vdash')$ where

- $A' = (\{0\} \times A) \cup \{0\}$
- $X \in Con' \iff \{ a \mid (0, \; a) \in X \} \in Con$
- $X \vdash' a \iff [\, X \neq \emptyset \,\&\, a = 0 \;\text{ or }\; a = (0, \; b) \,\&\, \{ c \mid (0, \; c) \in X \} \vdash b \,]$

Lifting is an operation which given a stable information system produces a new one by joining a new proposition weaker than all the old ones.

Definition 4.2 (Sum) Let

$$\underline{A} = (A, \; Con_{\underline{A}}, \; \vdash_{\underline{A}})$$

and

$$\underline{B} = (B, \; Con_{\underline{B}}, \; \vdash_{\underline{B}})$$

be stable information systems. Define their *sum*, $\underline{A} + \underline{B}$, to be $\underline{C} =$

$(C,\ Con,\ \vdash)$ where

- $C = \{0\} \times A \cup \{1\} \times B$

- $W \in Con \iff \exists X \in Con_{\underline{A}}.\ W = \{(0,\ a) \mid a \in X\}$ or

$$\exists Y \in Con_{\underline{B}}.\ W = \{(1,\ b) \mid b \in Y\}$$

- $W \vdash c \iff W = \{(0,\ a) \mid a \in X\}\ \&\ c = (0,\ r)\ \&\ X \vdash_{\underline{A}} r$ or

$$W = \{(1,\ b) \mid b \in Y\}\ \&\ c = (1,\ t)\ \&\ Y \vdash_{\underline{B}} t$$

The effect of sum is to juxtaposing disjoint copies of two stable information systems. We can obtain the separated sum $\oplus$ by letting $\underline{A} \oplus \underline{B} =^{\text{def}} \underline{A}_{\uparrow} + \underline{B}_{\uparrow}$.

Definition 4.3 (Product) Let $\underline{A} = (A,\ Con_{\underline{A}},\ \vdash_{\underline{A}})$ and $\underline{B} = (B,\ Con_{\underline{B}},\ \vdash_{\underline{B}})$ be stable information systems. Define their ***product***, $\underline{A} \times \underline{B}$, to be $\underline{C} = (C,\ Con,\ \vdash)$ where

- $C = \{0\} \times A \cup \{1\} \times B$

- $W \in Con \iff \{a \mid (0,\ a) \in W\} \in Con_{\underline{A}}\ \&\ \{b \mid (1,\ b) \in W\} \in Con_{\underline{B}}$

- $W \vdash c \iff c = (0,\ r)\ \&\ \{a \mid (0,\ a) \in W\} \vdash_{\underline{A}} r$ or

$$c = (1,\ t)\ \&\ \{b \mid (1,\ b) \in W\} \vdash_{\underline{B}} t$$

The proposition set of the product is the disjoint union of propositions of the components. A finite set of propositions is consistent if its projections to the components are. And a consistent set entails a proposition if it does so when projected into the appropriate component.

Notations. π_0 and π_1 are projections which give the first and the second argument respectively when applied to a pair. When they are applied to a set S of pairs, we write $\pi_0 S$ and $\pi_1 S$ for the set of first argument and second argument of element in S, respectively.

Definition 4.4 Let $\underline{A} = (A,\ Con_{\underline{A}},\ \vdash_{\underline{A}})$ and $\underline{B} = (B,\ Con_{\underline{B}},\ \vdash_{\underline{B}})$ be stable information systems. A subset $m \subseteq^{fin} Con_{\underline{A}} \times B$ is said to be a ***molecule*** if m is a stable approximable mapping and

$$\exists \alpha \in m.\ [\ \forall \beta \in m.\ \{\ \pi_1 \alpha\ \} \vdash_{\underline{B}} \pi_1 \beta\ \&\ \{\ \pi_0 \alpha\ \} \vdash_{\underline{A}} \pi_0 \beta\ ].$$

Molecules capture complete primes in the function space.

Definition 4.5 (Function Space) Let $\underline{A} = (A,\ Con_{\underline{A}},\ \vdash_{\underline{A}})$ and $\underline{B} = (B,\ Con_{\underline{B}},\ \vdash_{\underline{B}})$ be stable information systems. Define their ***function space***, $[\underline{A} \to \underline{B}]$, to be $\underline{C} = (C,\ Con,\ \vdash)$ where

- $C = \{\ m \mid m \text{ is a molecule of } Con_{\underline{A}} \times B\ \}$
- $$X \in Con \Longleftrightarrow \begin{cases} \forall S \subseteq \bigcup X.\ \bigcup \pi_0 S \in Con_{\underline{A}} \Rightarrow \bigcup \pi_1 S \in Con_{\underline{B}} \\ \\ \forall \alpha, \beta \in \bigcup X. [\{\pi_1 \alpha\} \dashv\vdash_{\underline{B}} \{\pi_1 \beta\}\ \&\ \pi_0 \alpha \cup \pi_0 \beta \in Con_{\underline{A}}] \\ \qquad \Rightarrow \pi_0 \alpha \dashv\vdash_{\underline{A}} \pi_0 \beta \end{cases}$$
- $X \vdash c \Longleftrightarrow \exists c' \in X.(\forall \alpha \in c \exists \beta \in c'.\ \{\ \pi_1 \alpha\ \} \dashv\vdash_{\underline{B}} \{\ \pi_1 \beta\ \}\ \&\ \pi_0 \alpha \dashv\vdash_{\underline{A}} \pi_0 \beta\ )$

Theorem 4.1 Lifting is a continuous function $(\ \ )_\uparrow : \mathbf{CPO}_{\text{sis}} \to \mathbf{CPO}_{\text{sis}}$. Sum, product and function space $+,\ \times,\ \to :\ \mathbf{CPO}^2_{\text{sis}} \to \mathbf{CPO}_{\text{sis}}$ are also continuous functions.

Proof We take the construction of function space as an example. Other cases are much simplier, hence omitted.

First we check that $\to$ preserves stable information systems. Let $\underline{A}$, $\underline{B}$ be stable information systems as in definition 4.5. It is easy to see that

$[\underline{A} \to \underline{B}]$ is an information system. Suppose $X \vdash c$ in the function space. By definition 4.5 $\exists b \in X.\ \{\, b \,\} \vdash c$. Thus axiom 6 holds. Suppose a is a molecule. We want to show that $\{\, b \mid \{\, a \,\} \vdash b \ \&\ b$ is a molecule $\}$ is finite. By definition, $\{\, a \,\} \vdash b$ implies

$$\forall \alpha \in b \exists \beta \in a.\ \{\, \pi_1 \alpha \,\} \dashv\vdash_{\underline{B}} \{\, \pi_1 \beta \,\} \ \&\ \pi_0 \alpha \dashv\vdash_{\underline{A}} \pi_0 \beta.$$

But because $\underline{A}$ and $\underline{B}$ are stable, we know that for any $\beta \in a$ the sets

$$\{\, s \in B \mid \{\, \pi_1 \beta \,\} \vdash_{\underline{B}} s \,\}$$

and

$$\{\, X \subseteq A \mid \pi_0 \beta \vdash_{\underline{A}} X \,\}$$

are finite. Hence $\{\, b \mid \{\, a \,\} \vdash b \ \&\ b$ is a molecule $\}$ is finite.

$\to$ is monotonic in its first argument. Suppose $\underline{A} \trianglelefteq \underline{A'}$. Write

$$\underline{C} = (\, C,\ Con,\ \vdash \,) = [\underline{A} \to \underline{B}]$$

and

$$\underline{C'} = (\, C',\ Con',\ \vdash' \,) = [\underline{A'} \to \underline{B}].$$

We check 1, 2 and 3 in definition 3.1, to show that $\underline{C} \trianglelefteq \underline{C'}$. Axiom 1 is trivial. Axiom 2. Suppose $\{\, c_i \mid 1 \le i \le n \,\} \in Con$. Then clearly $\{\, c_i \mid 1 \le i \le n \,\} \subseteq C$ and $\{\, c_i \mid 1 \le i \le n \,\} \in Con'$. On the other hand, suppose $\{\, c_i \mid 1 \le i \le n \,\} \subseteq C$ and $\{\, c_i \mid 1 \le i \le n \,\} \in Con'$. We have

$$\forall S \subseteq \textstyle\bigcup_i c_i.\ \bigcup \pi_0 S \in Con_{\underline{A'}} \Longrightarrow \bigcup \pi_1 S \in Con_{\underline{B}}\ \&$$

$$\forall \alpha, \beta \in \textstyle\bigcup_i c_i.\ \{\, \pi_1 \alpha \,\} \dashv\vdash_{\underline{B}} \{\, \pi_1 \beta \,\} \&\ \pi_0 \alpha \cup \pi_0 \beta \in Con_{\underline{A'}}$$

$$\Longrightarrow \pi_0 \alpha \dashv\vdash_{\underline{A'}} \pi_0 \beta$$

However,

$$\bigcup \pi_0 S \in Con_{\underline{A}} \Longrightarrow \bigcup \pi_0 S \in Con_{\underline{A'}}$$

and

$$\pi_0\alpha \cup \pi_0\beta \subseteq A \; \& \; \pi_0\alpha \dashv\vdash_{\underline{A'}} \pi_0\beta \Longrightarrow \pi_0\alpha \dashv\vdash_{\underline{A}} \pi_0\beta,$$

as $\underline{A} \trianglelefteq \underline{A'}$. So we have $\{\, c_i \mid 1 \leq i \leq n \,\} \in Con$.

Axiom 3 follows from a similar argument used in 2. Let

$$\underline{A_0} \trianglelefteq \underline{A_1} \trianglelefteq \cdots \trianglelefteq \underline{A_i} \trianglelefteq \cdots$$

be a chain of stable information systems. Let m be a molecule of $[(\bigcup_i \underline{A_i}) \to \underline{B}]$. Then $\bigcup \pi_0 m \subseteq^{\text{fin}} \bigcup_i A_i$. Hence $\bigcup \pi_0 m \subseteq A_j$ for some j, which means m is a molecule of $[\underline{A_j} \to \underline{B}]$. Thus m is a molecule of $\bigcup_i [\underline{A_i} \to \underline{B}\,]$. By Proposition 3.2 we deduce that $\to$ is continuous in its first argument. Similarly but easier we can proof that $\to$ is continuous in its second argument hence it is continuous. ∎

Therorem 4.1 provides a tool for solving equations of stable information systems by fixed point theory, since all the constructions introduced give rise to continuous functions and hence the existence of fixed points.

The reader may wonder why the construction of function space is so different from the one on information systems; Why can't we use propositions of the form $(\,X,\,Y\,)$ or even $(\,X,\,b\,)$ for the function space?

Information systems describe the consistency and the entailment relation on propositions. The entailment is global: Once $X \vdash a$, it holds for the information system irrespective of the particular computation of the type. As the stable approximable mapping suggests, a pair $(X,\ b)$ should read as: The set of propositions X entails the proposition b, and X is a weakest such set. If we take $(X,\ b)$ as the basic unit of information for the function space, it may lack the global property. Consider the function space on the simple information system $(\{1,\ 2\},\ \mathit{Con},\ \vdash)$, where Con is generated by requiring 1, 2 to be consistent and $\vdash$ by $\{2\} \vdash 1$. If we know that x is a computation which produces 2 with the minimal information 2, written as $(\{2\},\ 2) \in x$, we know that 1 is somehow also produced, since we have $\{2\} \vdash 1$. We can then ask what is the minimal information needed for x to produce 1. There are three possibilities: $(\{2\},\ 1) \in x$, $(\{1\},\ 1) \in x$ and $(\emptyset,\ 1) \in x$. Therefore $(\{2\},\ 2)$ entails $(\{2\},\ 1)$, or $(\{1\},\ 1)$, or $(\emptyset,\ 1)$, but not all of them at the same time (it depends on the computation x). This illustrates why we cannot get a global entailment by using propositions of the form $(X,\ b)$ for the function space.

Our construction of function space works for the example in the following way. There are altogether nine molecules, four containing $(\emptyset,\ 1)$, three containing $(\{1\},\ 1)$, two containing $(\{1,\ 2\},\ 1)$ and $(\{2\},\ 1)$. For example, $\{(\emptyset,\ 1)\}$ is one of the molecule. Clearly, these nine molecules corresponding to the nine complete primes in $\mathit{Pt}\,(\underline{A}) \to_s \mathit{Pt}\,(\underline{A})$, where $\underline{A}$ is the stable information system under consideration.

There is a special class of stable information systems for which one can indeed use $(X,\ b)$ as propositions for the function space. They are the stable information systems with a trivial entailment relation: $X \vdash$

a iff $a \in X$. It can be shown that these stable information systems are closed under all the constructions proposed in section 4. In fact if we further require *Con* to be binary, in the sense that $X \in Con$ iff $\forall a,\ b \in X.\ \{a, b\} \in Con$, and $\{a\} \dashv\vdash \{b\}$ implies $a = b$, then they are just Girard's coherent spaces (Girard, 1987). The reason for this is very simple: The domains $Pt\,(\underline{A})$ for such stable information systems are binary complete (coherent) and the complete primes of $Pt\,(\underline{A})$ are of the form $\{a\}$ with $a \in A$. More detailed treatments of coherent spaces are given in (Zhang, 1989b).

5. Conclusion

We have presented here a representation of dI-domains as information systems. The representation is formulated in terms of an equivalence between the category of dI-domains and the category of stable information systems. Through this representation as well as the related constructions, a more clear picture of the structure of stable functions is exposed.

In stable information systems the propositions correspond to some kind of Scott open sets. This fact promised them to be an important link in the development of logic of dI-domains. The logic of dI-domains should include the stable information systems as its backbone, but with certain kind of logical operations like conjunction and disjunction explicitly put on the propositions. The smooth formulation of such a logic requires a characterisation of stable functions in terms of some kind of Scott open sets. These open sets turn out to be disjoint in nature, in the sense that the union of two open sets makes sense in general only if their intersection is empty. It is reasonable, therefore, to expect that the logic of dI-domains is a kind of disjunctive logic. In (Zhang 1989b), progress along this direction is reported.

Acknowledgements

I was introduced to the semantics world by Glynn Winskel and I am grateful for his guidance. I'd like to thank Martin Hyland for helpful suggestions at the early stage of the work.

This work was partially done at the Computer Laboratory of Cambridge University, where I was supported by a studentship from Trinity College. Many thanks to the Department of Computer Science of Aarhus University for their hospitality.

References

Berry, G. (1978), Stable models of typed λ–calculi, LNCS 62

Girard, Jean-Yves (1987), Linear logic, Theoretical Computer Science 50

Johnstone, P.T. (1977), A syntactic approach to Diers' localizable categories, Lecture Notes in Mathematics 753

Johnstone, P.T. (1982), *Stone spaces*, Cambridge University Press

Kahn, G., Plotkin, G. (1978), Domaines concretes, Rapport INRIA Laboria No. 336

Larsen, K., Winskel, G. (1984), Using information systems to solve recursive domain equations effectively, Lecture Notes in Computer Science 173

Maclane, S. (1971), Categories for the working mathematician, Springer-Verlag

Nielsen, M., Plotkin, G., Winskel, G. (1981), Petri nets, event structures and domains, pert 1, Theoretical Computer Science 13

Scott, D. S. (1982), Domains for denotational semantics, LNCS 140

Smyth, M.B. (1983), Power domains and predicate transformers: a topological view. Lecture Notes in Computer Science 154

Winskel, G. (1988), Event structures, Lecture Notes in Computer Science 354

Zhang, G. Q. (1989a), DI-domains as Informations systems, Lecture Notes in Computer Science 372

Zhang, G. Q. (1989b), *Logics of Domains*, PhD thesis, University of Cambridge