DI-Domains as Prime Information Systems[1]

Guo Qiang Zhang

Department of Computer Science

The University of Georgia

Athens, GA 30602

[1]An extended abstract of the paper appeared in the proceedings of ICALP 89.

Abstract

This paper introduces prime information systems. Prime 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, prime 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 prime information systems can be defined using fixed point theory.

1. Introduction

The classical framework for the denotational semantics of programming languages is the category of Scott domains with continuous functions. A Scott domain is a complete partial order whose elements are thought of as some information about computation. To get a reasonably good structure Scott domains are required to satisfy a property called consistent completeness (see section 2). In (Scott 1982) a representation of Scott domains is introduced, using some kind of structure called information systems. This representation has two prominent advantages. First, information systems only use a very limited part of (elementary) set theory that are familiar to many people, and therefore, is less 'abstract' than the complete partial orders. Secondly, information systems provide a logical approach to domain theory which turned out to be important to the understanding of the relation between denotational semantics and program logics.

There are many other frameworks for the denotational semantics of programming languages, among which there is the less standard but important category of dI-domains and stable functions. DI-domains were discovered by Berry 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. 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 (Winskel 1988), 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 at the same time or not, and the enabling relation specifies the condition when a particular event may occur with regards to the occurrence of other events.

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.

Therefore, although event structures provide a concrete, less 'abstract' representation of dI-domains, a representation which captures the logical aspects of dI-domains will be very useful.

This paper introduces prime information systems, to represent dI-domains by logical structures similar to information systems. Section 2 gives the definition of prime information systems and show that they determine dI-domains. Section 3 introduces a category of prime information systems. This category is shown to be equivalent to the category of dI-domains. In Section 4, a complete partial order of prime information system is given so that recursive prime information systems can be defined. Constructions in the category of prime information systems are introduced in Section 5. Various common constructions turn out to specify continuous functions in the complete partial order given in Section 4. Discussions are given at the conclusion section.

2. Prime Information Systems

In this section we introduce prime information systems and show that they determine dI-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 2.1 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 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.

There is a more 'proof theoretical' way to look at information systems (Zhang, 1989). Considering sequents of the form $X \ \rightarrow\ Y$ (Note that a more common notation for Gentzen's sequents is $X \vdash Y$, but we have used $X \vdash Y$ to mean something different), where X, Y are finite sets of propositions with X non-empty. An information system is just a special kind of such sequents satisfying the following well-known rules of proof theory:

$$(Identity) \qquad a \ \rightarrow\ a$$

$$(Weakening) \qquad \frac{X' \supseteq X \qquad X \ \rightarrow\ Y \qquad Y \subseteq Y'}{X' \ \rightarrow\ Y'}$$

$$(Cut) \qquad \frac{X \ \rightarrow\ Y, a \qquad a, X' \ \rightarrow\ Y'}{X, X' \ \rightarrow\ Y, Y'}$$

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 2.2 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 (in the sense of Mac Lane 1971) 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.

Definition 2.3 An information system $\underline{A} = (A, Con, \vdash)$ is called *prime* if it satisfies two extra axioms:

6. $X \vdash a \Longrightarrow \exists b \in X.\ \{b\} \vdash a$
7. $\{b \mid X \vdash b\}$ is finite for all finite proposition set X

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 prime information systems are similar to prime event structures (Winskel, 1988). The only difference is that the entailment relation here determines a pre-order while for prime event structures the causal dependency relation is a partial order. Prime event structures in the sense of Winskel 1988 can be seen both as information systems and event structures. By taking a logical approach (tokens as propositions and configurations as theories) we committed ourselves not to require that the entailment between propositions be a partial order. The technical differences between prime information systems and prime event structures are superficial at this stage, though. However, later when we come to the constructions in the category of prime information systems, especially the construction of stable function space, the reason for taking the logical stand (and hence the name prime information systems instead of prime event structures) becomes more clear. There, molecules are used as tokens. Molecules are finite collections of some pairs which specify the minimal information. They can be regarded as a conjunction of specifications about minimal information. But regarding the molecules as events seemed to lack a computational justification.

Axiom 6 requires, in particular, that when $X \vdash a$ in a prime information system, X must be non-empty.

Axiom 7 for prime information systems corresponds to the axiom of finite cause for event structures. There is a strong computational intuition behind the axiom there (Winskel, 1988). 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 $\dashv\vdash$ −equivalent classes in $\{ b \mid X \vdash b \}$ rather than the whole set $\{ b \mid X \vdash b \}$ be finite.

Axiom 7 is equivalent to, under the assumption of Axiom 6, the following

$$\text{Axiom 7' } \forall a \in A. \{ b \mid \{ a \} \vdash b \} \text{ is finite}$$

which is easier to use. Obviously Axiom 7 implies Axiom 7'. In reverse, assume we have Axiom 7'. By Axiom 6 whenever $X \vdash b$, there is some $a \in X$ such that $\{ a \} \vdash b$. But for each $a \in X$, $\{ b \mid \{ a \} \vdash b \}$ is finite. Therefore $\{ b \mid X \vdash b \}$ is also finite, since X is.

For the convenience of the reader we recall some basic definitions of domain theory. A *partial order* is a set D with a binary relation on D which is reflexive ($x \sqsubseteq x$), transitive ($x \sqsubseteq y \,\&\, y \sqsubseteq z \implies x \sqsubseteq z$), and antisymmetric ($x \sqsubseteq y \,\&\, y \sqsubseteq x \implies x = y$). Without antisymmetry D is called a pre-order. When $x \sqsubseteq y$, we say x is below y or y dominates x or x is less (smaller) or equal to y. Let $(D, \sqsubseteq)$ be a partial order and $X \subseteq D$ a subset. Say y is an upper bound of X if $\forall x \in X$, $x \sqsubseteq y$. Similarly, y is a lower bound of X if $\forall x \in X$, $y \sqsubseteq x$. The least upper bound (sup, supremum, lub, join) of X is an upper bound of X which is dominated by any other upper bound of X. The lub of X is written as $\bigsqcup X$, or $\bigsqcup\limits_{i \in \omega} x_i$ if $X = \{ x_i \mid i \in \omega \}$ where ω is the set $\{ 0, 1, \cdots \}$. It is written as $a \sqcup b$ when X is $\{ a, b \}$. The greatest lower bound (inf, infimum, glb, meet) of X is a lower bound of X which dominates every other lower bound of X. The glb of X is written as $\bigsqcap X$. When $X = \{ x_i \mid i \in \omega \}$ we write $\bigsqcap_{i \in \omega} x_i$ for $\bigsqcap X$. We write $a \sqcap b$ when X is $\{ a, b \}$. X is *compatible*, written $X \uparrow$, if X has an upper bound. When $\{ a, b \}$ is compatible we write $a \uparrow b$.

A *directed* set of a partial order $(D, \sqsubseteq)$ is a non-null subset $S \subseteq D$ such that $\forall s, t \in S \exists u \in S. s \sqsubseteq u \,\&\, t \sqsubseteq u$. A *complete partial order* (cpo) is a partial order $(D, \sqsubseteq)$ which has a least element $\perp$ and all least upper bounds for directed subsets. An *isolated (or finite)* element of a cpo $(D, \sqsubseteq)$ is an element $x \in D$ such that for any directed subset $S \subseteq D$ when $x \sqsubseteq \bigsqcup S$

there is an $s \in S$ such that $x \sqsubseteq s$. We write D^0 for the set of finite elements of D. A cpo $(D, \sqsubseteq)$ is *algebraic* if for all $x \in D$, $\{ e \sqsubseteq x \mid e \in D^0 \}$ is directed and $x = \bigsqcup\{e \sqsubseteq x \mid e \in D^0\}$. *Domains* are algebraic cpos. However we should not take this name too seriously; there have been different definitions of domains in the literature. When $(D, \sqsubseteq)$ is algebraic and D^0 is countable, D is said to be ω-algebraic. A cpo is a *Scott domain* if it is ω-algebraic and *consistently complete*, i.e., every compatible subset X of D has a least upper bound $\bigsqcup X$.

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

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

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

It is convenient to work with another characterization 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 $x \in D$. We have the following lemma, due to Winskel. However its proof does not seem to be widely accessible. Here we present a different but straightforward proof, without using the notion of prime intervals.

Lemma 2.1 Suppose D is a Scott domain satisfying axiom I. Then D is prime algebraic iff it is a dI-domain.

First there are several observations that will be used.

Observation 1 A domain D satisfies axiom d iff for all $x, y, z \in D$

$$\{ x, y, z \} \uparrow \Longrightarrow x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z).$$

This is because for any compatible pair y, z in D, $\{x \sqcap (y \sqcup z),\ y,\ z\}$ is a compatible set. Therefore,

$$[x \sqcap (y \sqcup z)] \sqcap (y \sqcup z) = \Big[[x \sqcap (y \sqcup z)] \sqcap y\Big] \sqcup \Big[[x \sqcap (y \sqcup z)] \sqcap z\Big],$$

or

$$x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z).$$

Observation 2 If a domain D satisfies axiom d then by repeatedly applying axiom d we have

$$x \sqcap \bigsqcup X = \bigsqcup \{ x \sqcap a \mid a \in X \}$$

for any finite compatible subset $X \subseteq D$.

Observation 3 Suppose X is a compatible subset of a Scott domain D and $X = X_1 \cup X_2$. We have

$$\bigsqcup X = (\bigsqcup X_1) \sqcup (\bigsqcup X_2).$$

Observation 4 The complete primes of a dI-domain D are those isolated elements which have a unique element immediately below them. We say x is immediately below y, or x is *covered* by y, if $x \sqsubseteq y$ and for every z such that $x \sqsubseteq z \sqsubseteq y$, either $x = z$ or $y = z$.

Observation 4 is not trivial and we give a proof. Let p be a complete prime and let X be the finite (by axiom I) set of isolated elements immediately below it. X must be non-empty since $\perp$ is not a complete prime. But X cannot have more than one element, for otherwise we have $p \sqsubseteq \bigsqcup X$ but not $p \sqsubseteq x$ for any $x \in X$. Therefore there is a unique element immediately below p.

Suppose, on the other hand, that there is a unique element w immediately below an isolated element $a \in D$. Let X be a compatible set such that

$$a \sqsubseteq \bigsqcup X.$$

We can assume that X is finite for a is isolated. By Observation 2,

$$a = a \sqcap (\bigsqcup X) = \bigsqcup\{\, a \sqcap x \mid x \in X \,\}.$$

Clearly $a \sqcap x \sqsubseteq a$. If $a \sqcap x \neq a$ then $a \sqcap x \sqsubseteq w$, since w is the unique element immediately below a. Therefore for $a \sqsubseteq \bigsqcup X$ to hold, we must have $a \sqcap x = a$ for some $x \in X$. That means, $a \sqsubseteq x$ for some $x \in X$. Hence a is a complete prime.

Proof of Lemma 2.1:

Assume D is a prime algebraic Scott domain satisfying axiom I. Let $x \uparrow y$ and

$$x = \bigsqcup \{ p \sqsubseteq x \mid p \in D^p \},$$

$$y = \bigsqcup \{ q \sqsubseteq y \mid q \in D^p \}.$$

Then using that fact that p and q's are complete primes, we get

$$x \sqcup y = \bigsqcup (\{ p \sqsubseteq x \mid p \in D^p \} \cup \{ q \sqsubseteq y \mid q \in D^p \}),$$

$$x \sqcap y = \bigsqcup (\{ p \sqsubseteq x \mid p \in D^p \} \cap \{ q \sqsubseteq y \mid q \in D^p \}).$$

Axiom d then follows from Observation 1, using the prime algebraicness of D.

Assume D is a dI-domain. We show that it is prime algebraic. For that it is enough to show that

$$\forall d \in D^0 .\, d = \bigsqcup \{ p \sqsubseteq d \mid p \in D^p \},$$

using the algebraicness of D. If $d \in D^0$ is already a complete prime, there is nothing to prove. If $d \in D^0$ is not a complete prime then

$$d = \bigsqcup \{ a \sqsubseteq d \mid a \text{ is covered by } d \},$$

by Observation 4. For each a covered by d, it is either a complete prime or an isolated element but not a complete prime. For the later case a is again the lub of those elements covered by a. Reasoning in this way repeatedly, we get down to d_0's each having a unique element below it. Those d_0's, by Observation 4, are complete primes. Going upwards from those d_0's we see that the original d is the lub of the complete primes below it. ∎

Now it is easy to prove Theorem 2.1.

Proof of Theorem 2.1: Axiom 7 implies that $(\mathit{Pt}(\underline{A}), \subseteq)$ 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 $(\mathit{Pt}(\underline{A}), \subseteq)$, and $(\mathit{Pt}(\underline{A}), \subseteq)$ is prime algebraic. But these follow from axiom 6. ∎

3. A Category of Prime Information Systems

The subtle structure of stable functions makes it rather difficult to get the right definition of morphisms to form a category of prime information systems. A slightly modified version of the approximable mappings used in the category of information systems does not lead to a characterization of stable order. To get some guidance we study the structure of stable functions first. In particular, we generalize the notion of traces as used in the coherent spaces to dI-domains. After that, it becomes clear what should be the morphisms we are looking for.

For convenience we use p, q for complete primes and a, b for isolated elements.

Definition 3.1 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 totally 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 3.1 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, by axiom I we know 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

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

∎

Note that Definition 3.1 can be extended to continuous functions (not necessarily stable) on dI-domains. The proof of Lemma 3.1 does not need the stability of 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 3.2 Let $f, g : D \to E$ be stable functions. We have

1. If $f \sqsubseteq_s g$ then $\mu f \subseteq \mu g$;
2. $(a, \; p), \; (a', \; p) \in \mu f \;\&\; a \uparrow a' \Longrightarrow a = a'$.

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

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

Apparently $b \sqsubseteq a$, and, by axiom I, $(b,\ p) \in \mu g$. $f \sqsubseteq_s g$ implies, by definition,

$$f(b) = f(a) \sqcap g(b).$$

Hence $f(b) \sqsupseteq p$, as $f(a) \sqsupseteq p$, $g(b) \sqsupseteq p$ (using axiom I and stability). We must have $a = b$, for $(a,\ p) \in \mu f$. This means $(a,\ p) \in \mu g$, too.

Suppose f is stable, $a \uparrow a'$, and $(a,\ p)$, $(a',\ p)$ are members of μf. It follows that $f(a) \sqsupseteq p$, $f(a') \sqsupseteq p$. As $a \uparrow a'$, we have

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

Thus we also have $a \sqcap a' = a$ and $a \sqcap a' = a'$. Hence $a = a'$. ∎

Now we have

Theorem 3.1 Let $f, g \in [D \to_s E]$. $f \sqsubseteq_s g$ iff $\mu f \subseteq \mu g$. For $\{\ (a_i,\, p_i\,) \mid i \in I\ \} \subseteq D^0 \times E^p$,

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

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

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

Proof Lemma 3.2 shows that $f \sqsubseteq_s g$ implies $\mu f \subseteq \mu g$. Suppose, on the other hand, that $\mu f \subseteq \mu g$. It follows from Lemma 3.1 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 3.2, $a = b$. 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 3.1.

Suppose $\{\,(a_i,\ p_i) \mid i \in I\,\} = \mu f$ for some stable function f. It is routine to check that the three properties mentioned in Theorem 3.1 hold.

Let $\{\,(a_i,\ p_i) \mid i \in I\,\} \subseteq D^0 \times E^p$ be a set with the three properties. We show that the stable function f for which (Lemma 3.1 concludes that such a function is unique)

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

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

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

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

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

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

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

Since E is prime algebraic,

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

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

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

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

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

Let $y \sqsubseteq a_j$ and $f(y) \sqsupseteq p_j$, i.e. $\bigsqcup \{\ p_i \mid a_i \sqsubseteq y\ \} \sqsupseteq p_j$. Since p_j is a complete prime, $p_i \sqsupseteq p_j$ for some i with $a_i \sqsubseteq y$. The third condition mentioned in Theorem 3.1 implies the existence of some k such that $p_k = p_j$ and $a_k \sqsubseteq a_i$. But $a_k = a_j$ since $a_k \uparrow a_j$. Hence $y = a_j$. This means $(a_j,\ p_j) \in \mu f$.

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

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

Since p is a complete prime, there is some p_i such that $p_i \sqsupseteq p$. By the third condition mentioned in Theorem 3.1 again, $p_j = p$ for some j such that $a_j \sqsubseteq a_i$. By the result from the previous paragraph we have $(a_j,\ p_j) \in \mu f$. Therefore $a_j = a$ by Lemma 3.2 (taking $f = g$). ∎

Note that in Theorem 3.1 we have generalized the characterization of traces of stable functions from the category of coherent spaces to the category of dI-domains. Indeed, comparing our Theorem 3.1 with Theorem IV.2. in (Girard, 1987) one finds out that if the three conditions in Theorem 3.1 are satisfied, one indeed gets a trace when the related dI-domains are coherent spaces. Functions of the form $\bigsqcup_{i \in I}[\, a_i, p_i\,]$ with I finite represent all the isolated elements in the stable function space, where $\{\ (\,a_i,\ p_i\,) \mid i \in I\ \}$ satisfies the three conditions mentioned in Theorem 3.1.

Corollary The complete primes of the stable function space $[D \to_s E]$ are functions $\bigsqcup_{i \in I}[\, a_i, p_i\,]$ determined by sets of pairs

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

satisfying the three conditions in Theorem 3.1, such that I is finite and for all $i \in I$ $a_i \sqsubseteq a$, $p_i \sqsubseteq p$ for some $(a,\, p) \in \{\ (a_i,\ p_i) \mid i \in I\ \}$.

Proof By Theorem 3.1 it is clear that such functions are complete primes. Suppose, on the other hand, that we have a complete prime f in the function space. Then $\mu\, f$ (see Definition 3.1) must be finite. Also, by Observation 4 in Section 2, there is only a unique proper subset $\mu\, g$ immediately contained in the trace set $\mu\, f$ for some stable function g. This can happen only when there exists a pair $(a,\ p) \in \mu\, f$ with the property as mentioned above. ∎

Now we come naturally to

Definition 3.2 Let $\underline{A} = (\,A,\ Con_{\underline{A}},\ \vdash_{\underline{A}}\,)$, $\underline{B} = (\,B,\ Con_{\underline{B}},\ \vdash_{\underline{B}}\,)$ be prime information systems. A stable approximable mapping $R : \underline{A} \to \underline{B}$ is a relation $R \subseteq \mathrm{Fin}_{\underline{A}} \times B$ which satisfies

1. Compatibility : $(\forall i \in I \,.\, a_i \, R \, p_i) \;\&\; \bigcup a_i \in Con_{\underline{A}} \Longrightarrow \{ p_i \mid i \in I \} \in Con_{\underline{B}}$ I finite

2. Minimality : $(a \cup b \in \mathrm{Fin}_{\underline{A}} \;\&\; a \, R \, p \;\&\; b \, R \, p) \Longrightarrow a = b$

3. Completeness : $a \, R \, p \;\&\; \{ p \} \vdash_{\underline{B}} q \Longrightarrow b \, R \, q$ for some $b \subseteq a$.

Here $\mathrm{Fin}_{\underline{A}}$ stands for the set of finite elements of $P_t(\underline{A})$. These conditions are instances of those used in Theorem 3.1.

The compatibility condition is quite standard, which is also used in Scott's information systems. The second condition expresses the *minimal* property. $a \, R \, p$ can be read as: a in $\underline{A}$ entails p in $\underline{B}$, and, moreover, a is a weakest one in $\underline{A}$ (or, it is not only sufficient but also necessary for a to produce p). For example, for any stable approximable mapping R, we cannot have $\emptyset \, R \, p$ and $a \, R \, p$ for some non-empty a at the same time. The third condition insists on *completeness*, in the sense that when a is a weakest proposition for p, all the propositions weaker than p must also have their weakest propositions specified.

Suppose $a \, R \, p$ and $\{ p \} \dashv\vdash \{ q \}$. By the completeness axiom there is some $b \subseteq a$, $b \, R \, q$. But again, $\{ q \} \vdash p$. Hence $c \, R \, p$ for some $c \subseteq b$. Clearly $a \cup c$ is a finite element. By minimality we must have $a = c$. Therefore $a = b$, which means $a \, R \, q$, too. This analysis shows that under the completeness axiom, minimality is equivalent to the otherwise weaker statement

$$(a \cup b \in \mathrm{Fin}_{\underline{A}} \;\&\; a \, R \, p \;\&\; b \, R \, q \;\&\; \{ p \} \dashv\vdash_{\underline{B}} \{ q \}) \Longrightarrow a = b.$$

Proposition 3.1 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

$$Pt\,(R)(x) = \{ p \mid \exists a \subseteq x \,.\, a \, R \, p \}$$

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

$$\{\,p_0,\; p_1, \cdots,\; p_n\,\} \subseteq Pt\,(\,R\,)(x).$$

There exist $a_0,\; a_1,\; \cdots,\; a_n$, such that

$$\forall i\,.\;(\;a_i \subseteq x \;\&\; a_i\,R\,p_i\;).$$

By axiom 2 of Definition 3.2 we have $\{\,p_0,\; p_1, \cdots,\; p_n\,\} \in Con_{\underline{B}}$.

$Pt\,(\,R\,)(x)$ is also closed under entailment. Assume $Y \subseteq Pt\,(\,R\,)(x)$ and $Y \vdash q$. We know that $\exists p \in Y\,.\,\{\,p\,\} \vdash q$. Since $p \in Pt\,(\,R\,)(x)$, there exists an $a \subseteq x$ such that $a\,R\,p$. By axiom 3 of Definition 3.2, $b\,R\,q$ for some $b \subseteq a$, which implies $q \in Pt\,(\,R\,)(x)$. Therefore $Pt\,(\,R\,)(x)$ 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 $p \in Pt\;R(x) \cap Pt\;R(y)$. Thus $\exists a \subseteq x\,, b \subseteq y$ such that $a\,R\,p$ and $b\,R\,p$. Since x and y are compatible, $a \cup b \in \mathrm{Fin}_{\underline{A}}$. Therefore, by axiom 1 of Definition 3.2, $a = b$, which implies $p \in Pt\;R(x \cap y)$. 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 $a\,R\,p$ by requiring a to be non-empty.

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

Proposition 3.2 Let $\underline{A}$ and $\underline{B}$ be prime information systems, and $R,\ S : \underline{A} \to \underline{B}$ stable approximable mappings. $R \subseteq S$ iff $Pt\,(\,R\,) \sqsubseteq_s Pt\,(\,S\,)$, where $\sqsubseteq_s$ is the stable order.

Proof Only if: Assume $R \subseteq S$ and let $x,\ y \in Pt(\,\underline{A})$, $x \subseteq y$ and $p \in Pt(R)(\,y\,) \cap Pt(S)(\,x\,)$. There must be $a \subseteq x$ and $b \subseteq y$ such that $a\,R\,p$ and $b\,S\,p$. 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 $Pt\,(R) \sqsubseteq_s Pt\,(S)$. If $a\,R\,p$ then $p \in (Pt\,R)(a) \subseteq (Pt\,S)(a)$. Therefore $b\,S\,p$ for some $b \subseteq a$. By the stable order we get

$$Pt\,R(b) = Pt\,R(a)\ \cap\ Pt\,S(b).$$

It is easy to deduce $p \in Pt\,R(a)\ \cap\ Pt\,S(b)$ since $a\,R\,p$ and $b\,S\,p$. This implies $p \in Pt\,R(b)$. For some $c \subseteq b$, therefore, $c\,R\,p$. By the first axiom of Definition 3.2 we have $a = c$, and therefore $a\,S\,p$. ∎

It is intended that stable approximable mappings are morphisms on prime information systems so that one gets a category. But this is far from clear at this stage: the identity Id should be given by $a\,Id\,p$ iff $a \dashv\vdash \{\,p\,\}$. 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 $\mathrm{Fin}_{\underline{A}} \times \mathrm{Fin}_{\underline{B}}$ such that $a\,R\,b$ iff there exist $a_i,\ p_i$ such that $a_i\,R\,p_i$, $1 \le i \le n$, $a = \bigcup_{1 \le i \le n} a_i \in Con_{\underline{A}}$, and $b = \{\,p_i \mid 1 \le i \le n\,\}$. It is not difficult to check that $\tilde{R}$ has similar properties to that of stable approximable mappings:

1. $(\forall i \in I \, . \, a_i \, \tilde{R} \, b_i \,) \;\&\; \bigcup a_i \in Con_{\underline{A}} \Longrightarrow \bigcup b_i \in Con_{\underline{B}}$

2. $(\, a \cup b \in \mathrm{Fin}_{\underline{A}} \;\&\; a \, \tilde{R} \, c \;\&\; b \, \tilde{R} \, c) \Longrightarrow a = b$

3. $a \, \tilde{R} \, b \supseteq c \Longrightarrow \exists a' \subseteq a . \, a' \, \tilde{R} \, c$

Intuitively, $\tilde{R}$ extends the minimal relation from propositions in $\underline{B}$ to its finite elements.

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

$$a \, (R \circ S) \, p \Longleftrightarrow \exists b \in \mathrm{Fin}_{\underline{B}} . \; a \, \tilde{R} \, b \;\&\; b \, S \, p.$$

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

Theorem 3.2 Prime information systems with stable approximable mappings form a category **PIS**.

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

Definition 3.3 Let D be a dI-domain. Define

$$SI \, D = (\, A, \, Con, \, \vdash \,),$$

by taking

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

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

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

Proposition 3.3 If D is a dI-domain then $SI\,(D)$ is a stable information system.

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

We can also get stable approximable mappings from stable functions.

Definition 3.4 Let D, E be dI-domains, and $f : D \to E$ a stable function. Define a relation $SI\,(\,f\,) \subseteq \mathrm{Fin}_{SI\,(D)} \times A_{SI\,(E)}$ by taking

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

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

Note $(x,\, p) \in \mu f$ requires, by Definition 3.1, x is a finite element of D and p is a complete prime of E.

Proposition 3.4 Let D, E be dI-domains, and $f : D \to E$ 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 3.2 hold for $SI\,f$.

Axiom 1 follows from Lemma 3.2.

Axiom 2 and 4 are easy.

Axiom 3 follows from the third property mentioned in Theorem 3.1 about the set μf. ∎

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

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

We conclude this section by

Theorem 3.3 $\mathit{Pt} : \mathbf{PIS} \to \mathbf{DI}$ and $\mathit{SI} : \mathbf{DI} \to \mathbf{PIS}$ are functors which determine an equivalence of **PIS** 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 $\mathit{Pt}\,(\underline{A})$ for some prime 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 prime information systems and

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

a stable function. Define a relation $R \subseteq \mathrm{Fin}_{\underline{A}} \times B$ by letting $a\,R\,p$ if $(a, \{\, q \mid \{\, p \,\} \vdash_{\underline{B}} q \,\}) \in \mu f$. It follows from Proposition 3.4 that this relation is an approximable mapping form $\underline{A}$ to $\underline{B}$. By Theorem 3.1, the stable function $\mathit{Pt}\,R$ determined by R is actually equal to f.

Suppose $R,\ S : \ \underline{A} \to \underline{B}$ are approximable mappings such that $\mathit{Pt}\,R = \mathit{Pt}\,S$. It follows from Proposition 3.2 that $R \subseteq S$ and $S \subseteq R$, and hence $R = S$. Therefore Pt is faithful. ∎

4. A Cpo of Prime Information Systems

In this section we introduce a subsystem relation on prime 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 4.1 Let $\underline{A} = (\, A,\, \mathit{Con}_{\underline{A}},\, \vdash_{\underline{A}} \,)$ and $\underline{B} = (\, B,\, \mathit{Con}_{\underline{B}},\, \vdash_{\underline{B}} \,)$ be prime information systems. $\underline{A} \trianglelefteq \underline{B}$ if

1. $A \subseteq B$

2. $X \in Con_{\underline{A}} \iff X \subseteq A \,\&\, X \in Con_{\underline{B}}$

4. $X \vdash_{\underline{A}} a \iff X \subseteq A \,\&\, X \vdash_{\underline{B}} a$

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 prime information systems. If $A = B$ and $\underline{A} \trianglelefteq \underline{B}$, then $\underline{A} = \underline{B}$.

Definition 4.2 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

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

Note that the above three conditions together say exactly that we have $fg \sqsubseteq_s \mathrm{id}$, the identity function.

Proposition 4.1 Let $\underline{A} = (\, A,\, Con_{\underline{A}},\, \vdash_{\underline{A}}\,)$ and $\underline{B}$ be prime 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 prime information systems. Clearly there is a least prime 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 4.1 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 2.3 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 4.1 $\{ a \} \vdash_k t$ 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 $\mathbf{CPO}_{\mathrm{pis}}$ for the 'cpo' of prime information systems under $\trianglelefteq$. $\mathbf{CPO}_{\mathrm{pis}}$ 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 $\underline{\vec{A}}$ for $(\underline{A_1},\ \underline{A_2}, \cdots \underline{A_n})$.

The least upper bound of an $\omega-$chain of $n-$tuples of prime information systems is then just the $n-$tuple of prime 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 from $n-$tuples of prime information systems to $m-$tuples of prime information systems is said to be continuous iff it is monotonic, i.e. $\vec{\underline{A}} \trianglelefteq \vec{\underline{B}}$ implies $F(\vec{\underline{A}}) \trianglelefteq F(\vec{\underline{B}})$ and preserves $\omega-$increasing chains of prime information systems, i.e.

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

implies

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

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 4.2 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 prime 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 prime information systems by the remark given just before Definition 4.2. ∎

Now given any continuous function F on $\mathbf{CPO}_{\mathrm{pis}}$, we can get the least fixed point of F, which is the limit of the increasing ω−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

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

5. 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 $\mathbf{CPO}_{\mathrm{pis}}$. In this way we can produce solutions to recursive equations for prime 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 prime information system rules out the possibility $\emptyset \vdash a$. What is totally novel is the construction of function space.

Definition 5.1 (Lifting) Let $\underline{A} = (A,\ Con,\ \vdash)$ be a prime 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 prime information system produces a new one by joining a new proposition weaker than all the old ones.

Definition 5.2 (Sum) Let

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

and

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

be prime information systems. Define their *sum*, $\underline{A} + \underline{B}$, to be $\underline{C} = (\, C,\ Con,\ \vdash\,)$ where

- $C = \{\, 0 \,\} \times A \cup \{\, 1 \,\} \times B$
- $W \in Con \Longleftrightarrow \exists X \in Con_{\underline{A}}.\ W = \{\, (\, 0,\ a\,) \mid a \in X \,\}$ or
  $\exists Y \in Con_{\underline{B}}.\ W = \{\, (\, 1,\ b\,) \mid b \in Y \,\}$
- $W \vdash c \Longleftrightarrow W = \{\, (\, 0,\ a\,) \mid a \in X \,\}\ \&\ c = (\, 0,\ r\,)\ \&\ X \vdash_{\underline{A}} r$ 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 $\underline{A} \oplus \underline{B} =^{\mathrm{def}} \underline{A}_{\uparrow} + \underline{B}_{\uparrow}$.

Definition 5.3 (Product) Let $\underline{A} = (\, A,\ Con_{\underline{A}},\ \vdash_{\underline{A}}\,)$ and $\underline{B} = (\, B,\ Con_{\underline{B}},\ \vdash_{\underline{B}}\,)$ be prime information systems. Define their *product*, $\underline{A} \times \underline{B}$, to be $\underline{C} = (\, C,\ Con,\ \vdash\,)$ where

- $C = \{\, 0 \,\} \times A \cup \{\, 1 \,\} \times B$
- $W \in Con \Longleftrightarrow \{\, a \mid\ (\, 0,\ a\,) \in W \,\} \in Con_{\underline{A}}\ \&\ \{\, b \mid\ (\, 1,\ b\,) \in W \,\} \in Con_{\underline{B}}$
- $W \vdash c \Longleftrightarrow c = (\, 0,\ r\,)\ \&\ \{\, a \mid\ (\, 0,\ a\,) \in W \,\} \vdash_{\underline{A}} r$ 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 arguments and second arguments of elements in S, respectively.

Definition 5.4 Let $\underline{A} = (\, A,\ Con_{\underline{A}},\ \vdash_{\underline{A}}\,)$ and $\underline{B} = (\, B,\ Con_{\underline{B}},\ \vdash_{\underline{B}}\,)$ be prime information systems. A *molecule* m is a finite stable approximable mapping such that for some $(a,\, p) \in m$, $b \subseteq a$ and $\{\, p\,\} \vdash q$ for any other $(b,\, q)$ in m.

By the corollary to Theorem 3.1, molecules capture complete primes in the function space.

Definition 5.5 (Function Space) Let $\underline{A} = (\, A,\ Con_{\underline{A}},\ \vdash_{\underline{A}}\,)$ and $\underline{B} = (\, B,\ Con_{\underline{B}},\ \vdash_{\underline{B}}\,)$ be prime information systems. Define their *function space,* $[\,\underline{A} \rightarrow \underline{B}\,]$, to be $\underline{C} = (\, C,\ Con,\ \vdash\,)$ where

- $C = \{\, m \mid m \text{ is a molecule}\,\}$
- $X \in Con \iff \bigcup X$ satisfies conditions 1 and 2 of stable approximable mapping
- $\{\, m\,\} \vdash m' \iff m' \subseteq m$

One can regard a molecule as a proposition which is a conjunction of specifications about minimal information.

Proposition 5.1 The function space construction given in the previous definition is an exponentiation in the category of prime information systems and stable approximable mappings.

Proof All we have to do is to show that $Pt\,([\underline{A} \rightarrow \underline{B}])$ is isomorphic to the stable approximable mappings from $\underline{A}$ to $\underline{B}$. Function space is defined by taking special kinds of stable approximable mappings, the molecules, as propositions. The consistency relation corresponding to minimality and compatibility. It can be easily checked that by sending each element x in the function space to the union $\bigcup x$ (which is a stable approximable mapping) of molecules in x, one indeed gets an isomorphism. ∎

Theorem 5.1 Lifting is a continuous function $(\ )_{\uparrow} : \mathbf{CPO}_{\mathrm{pis}} \to \mathbf{CPO}_{\mathrm{pis}}$. Sum, product and function space $+,\ \times,\ \to :\ \mathbf{CPO}^2_{\mathrm{pis}} \to \mathbf{CPO}_{\mathrm{pis}}$ are also continuous functions.

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

First we check that $\to$ preserves prime information systems. Let $\underline{A}$, $\underline{B}$ be prime information systems as in Definition 5.5. It is easy to see that $[\underline{A} \to \underline{B}]$ is an information system. By Definition 5.5 axiom 6 holds. Suppose a is a molecule. We want to show that $\{\, b \mid \{\, a \,\} \vdash b \;\&\; b \text{ is a molecule} \,\}$ is finite. But by definition, $\{\, a \,\} \vdash b$ if $b \subseteq a$ and a, b are finite sets. Hence $\{\, b \mid \{\, a \,\} \vdash b \;\&\; b \text{ is a molecule} \,\}$ 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 5.1, 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

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

$$\forall \alpha, \beta \in \bigcup_i c_i.\ \{\, \pi_1 \alpha \,\} = \{\, \pi_1 \beta \,\} \&\ \pi_0 \alpha \cup \pi_0 \beta \in \mathrm{Fin}_{\underline{A'}}$$

$$\Longrightarrow \pi_0 \alpha = \pi_0 \beta.$$

However,

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

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

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 prime information systems. Let m be a molecule of $[(\bigcup_i \underline{A_i}) \to \underline{B}]$. Then $\bigcup \pi_0 m \subseteq^{\mathrm{fin}} \bigcup_i A_i$. 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 5.2 we deduce that $\to$ is continuous in its first argument. By a similar but easier proof we get that $\to$ is continuous in its second argument hence it is continuous. ∎

Theorem 5.1 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 (a, p) should read as: The set of propositions a entails the proposition p, and a is a weakest such set. If we take (a, p) 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 $(\{1, 2\}, Con, \vdash)$, 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 $\{1, 2\}$, written as $(\{1, 2\}, 2) \in x$, 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: $(\{1, 2\}, 1) \in x$, $(\{1\}, 1) \in x$ and $(\emptyset, 1) \in x$. Therefore $(\{1, 2\}, 2)$ entails $(\{1, 2\}, 1)$, or $(\{1\}, 1)$, or $(\emptyset, 1)$, but not all of them at the same time since they are inconsistent. This illustrates why we cannot get a global entailment by using propositions of the form (a, p) 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 $(\{1, 2\}, 1)$. For example, $\{(\emptyset, 1)\}$ is one of the molecule. Clearly, these nine molecules correspond to the nine complete primes in $[Pt(\underline{A}) \to_s Pt(\underline{A})]$, where $\underline{A}$ is the prime 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 prime information systems with a trivial entailment relation: $X \vdash a$ iff $a \in X$. It can be shown that these prime information systems are closed under all the constructions proposed in Section 5 except lifting. 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 prime 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).

6. Conclusion

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

In prime 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 prime 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 characterization of stable functions in terms of some kind of 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. One might first develop a theory about disjunctive locales in the light of Stone duality, and then naturally arrives at the intended disjunctive logic.

For readers interested in this direction see (Zhang 1990) for the topological study of stable functions in various categories, and (Zhang 1989b), for progress on disjunctive logics of stable functions.

Acknowledgments

I was introduced to the world of semantics 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. Thanks to P.-L. Curien and an anonymous referee for suggested improvements of the presentation.

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 $\lambda-$calculi, Lecture Notes in Computer Science 62

Berry, G., Curien P.-L. and Levy J.-J (1985), Full abstraction for sequential languages: the state of the art, *Algebraic methods in semantics* Nivat, M. and Reynolds, J.C. ed, Cambridge University Press

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 concrets, 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, Lecture Notes in Computer Science 140

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

Winskel, G. (1983), A representation of complete distributive algebraic lattices, Report 83-154 of the Computer Science Department, Carnegie-Mellon University

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

Zhang, G. Q. (1989), A representation of SFP, submitted to Information and Computation

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

Zhang, G. Q. (1989b), *Logics of Domains*, PhD thesis, University of Cambridge (available as technical report no. 185 at Computer Laboratory, Cambridge University, as well as technical report no. 298, Computer Science Department, Aarhus University)

Zhang, G. Q. (1990), Stable Neighbourhoods, submitted to Theoretical Computer Science