Universal Quasi-Prime Algebraic Domains

Guo-Qiang Zhang
Department of Computer Science
The University of Georgia
Athens, Georgia 30602
E-mail: `gqz@cs.uga.edu+`

Summary. This paper demonstrates the existence of a saturated quasi-prime algebraic domain. It also presents a cpo of quasi-prime generated information systems for solving domain equations.

Key Words: Domain theory, semantics of programming languages, universal structures, lattices, universal algebra.

1 Introduction

Quasi-prime algebraic domains are a class of cpos within the Scott domains. They are introduced by the author in [11] as a new domain-theoretic model for linear logic. Quasi-prime algebraic domains with quasi-linear functions form a monoidal closed category. The unique characteristic of the category is that the morphisms are not 'linear', as the term 'quasi-linear' suggests. This is a bit surprising, since all other known domain theoretic linear categories all use linear functions as morphisms [11].

However, how robust and useful the concept of quasi-prime algebraic domains is depends on whether or not they have other nice domain theoretic properties. One of the desirable properties is the existence of a universal (or even saturated) domain [4] in a certain category. The other related property to have is a framework for solving domain equations by fixed point construction, as in [5, 8]. It is the purpose of the paper to establish these results for quasi-prime generated information systems which represent quasi-prime algebraic domains.

One of the most useful results on universal domains is given in the work of Droste and Göbel [1], who introduced the Fraissé-Jónsson theorem in model theory into the area of domain theory. This makes it much easier to show the existence of certain universal domains because it reduces the existence of a saturated structure to the amalgamation property of the finite objects of a certain category.

We apply the result of Droste and Göbel for showing the existence of a saturated (universal, homogeneous) quasi-prime algebraic domain. Our main definition here is the notion of q-embeddings for quasi-prime algebraic domains. The appropriate notion of embeddings for Scott domains (call them s-embeddings) [5, 2] and for dI-domains (call them r-embeddings – 'r' for rigid) [8] are well-known . However, none of the these embeddings works for quasi-prime algebraic domains, for the following reasons:

- The s-embeddings are too general: under this embedding the colimit of an ω-chain of finite Scott domains (which are quasi-prime algebraic) need not be quasi-prime algebraic, because any Scott domain can be seen as a colimit of this kind.

- The r-embeddings are too specific: there are certain quasi-prime algebraic domains which cannot be represented as a colimit of any chain of finite Scott domains, although the r-embeddings are suitable for the I-domains (dI-domains without axiom d).

With q-embeddings we get the desired algebroidal category of quasi-prime generated information systems. The finite objects of the category are shown to have the amalgamation property. This implies the existence of a saturated quasi-prime algebraic domain.

Based on the notion of q-embeddings, we will also introduce a cpo of quasi-prime generated information systems on which various constructions are shown to induce continuous functions. This implies the existence of recursively defined quasi-prime generated information systems.

Here is the outline of the structure of the paper. In Section 2 we introduce the notions of quasi-primes and quasi-prime algebraic domains. In Section 3 we represent quasi-prime algebraic domains as information systems. This will bring technical convenience to the rest of the paper. Section 4 recalls the result of Droste and Göbel on the existence of universal domains. Section 5 presents a category of quasi-prime algebraic information system with q-embeddings as the morphisms. Section 6 verifies the amalgamation property of the finite objects of the category introduced in Section 5. In the final section, we introduce a cpo of quasi-prime generated information systems and the continuity of various constructions on this cpo.

2 Quasi-Prime Algebraic Domains

In a Scott domain D, an element p is called a *complete prime* if

$$p \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X.\ p \sqsubseteq x.$$

On the other hand, an element q is called a *quasi-prime* if

$$q = \bigsqcup X \Rightarrow \exists x \in X.\ p = x.$$

A Scott domain is *prime algebraic* if every element is the least upper bound of complete primes below (less than or equal to) it. Similarly, a Scott domain is *quasi-prime algebraic* if every element is the least upper bound of quasi-primes below it. As far as functions are concerned, for prime algebraic domains, a function f is *linear* if and only if

$$p \sqsubseteq f(x) \Rightarrow \exists r \sqsubseteq x.\ p \sqsubseteq f(r), \quad \text{where } p, r \text{ are complete primes.}$$

For quasi-prime algebraic domains, a function f is *quasi-linear* if and only if

$$q \sqsubseteq f(x) \Rightarrow \exists s \sqsubseteq x.\ q \sqsubseteq f(s), \quad \text{where } q, s \text{ are quasi-primes.}$$

The following table summarizes the relationships between complete primes and quasi-primes, linear functions and quasi-linear functions. For comparison, we also include isolated elements and continuous functions.

	Definition
Isolated Elements	$d \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X.\ d \sqsubseteq x$ (X directed)
Complete Primes	$p \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X.\ p \sqsubseteq x$ (X bounded)
Quasi-Primes	$q = \bigsqcup X \Rightarrow \exists x \in X.\ p \sqsubseteq x$
Continuous Functions	$f(\bigsqcup X) = \bigsqcup \{f(x) \mid x \in X\}$ (X directed)
Linear Functions	$f(\bigsqcup X) = \bigsqcup \{f(x) \mid x \in X\}$ (X bounded)
Quasi-Linear Functions	$q \sqsubseteq f(x) \Rightarrow \exists s \sqsubseteq x.\ q \sqsubseteq f(s)$ (q, s quasi-primes)

It is helpful to note that all *finite* Scott domains are quasi-prime algebraic (we need the next theorem for this). Of course not all finite

Scott domains are prime algebraic. Moreover, not all Scott domains are quasi-prime algebraic. The following theorem can help us find such an example. This theorem is extremely helpful in identifying quasi-primes. This, I believe, is also the key advantage of working with quasi-prime algebraic domains.

Theorem 2.1 *Let D be a Scott domain. An element $q \in D$ is a quasi-prime iff there is a unique element $q^{\prec}$ immediately below q:*

$$x \sqsubset q \Rightarrow x \sqsubseteq q^{\prec}.$$

It is worth pointing out that, as a consequence of the theorem, if x has a unique element immediately below it, then x is an isolated element. Bottoms are never quasi-primes.

It is now easy to see that, if one turns the complete binary tree upside down and adjoining a bottom element, one gets a Scott domain which is not quasi-prime algebraic, since there is no quasi-primes in this domain. Also note that it is easy to show, by mathematical induction on the number of elements below an isolated element, that finite Scott domains are quasi-prime algebraic.

We end this section by remarking that a dual concept of quasi-primes was mentioned in [3] (pages 92-93), called the *completely irreducible elements*. Quasi-primes were introduced in [11] as a by-product of studying quasi-prime algebraic domains. Therefore, our motivation, objectives, and results are, in any case, totally different from that of [3]. More important, we have gone far beyond a particular class of elements. We consider domains generated by these elements, and we consider categories of quasi-prime algebraic domains. We have introduced quasi-linear functions (detail presented in a forthcoming paper) which are the corresponding morphisms for quasi-primes.

3 Quasi-Prime Generated Information Systems

This section introduces a representation of quasi-prime algebraic domains as information systems. This will bring technical convenience for the presentation of the rest of the paper.

An information system [6] is a structure $\underline{A} = (\, A,\ Con,\ \vdash\,)$ where

- A is a countable set of propositions (tokens),
- Con is a collection of finite subsets of A (the consistent sets),
- $\vdash \subseteq Con - \{\emptyset\} \times A$, the entailment relation,

which satisfy

- $(X \subseteq Y \;\&\; Y \in Con) \Rightarrow X \in Con$,
- $a \in A \Rightarrow \{\, a \,\} \in Con$,
- $(X \vdash a \;\&\; X \in Con) \Rightarrow X \cup \{\, a \,\} \in Con$,
- $(a \in X \;\&\; X \in Con) \Rightarrow X \vdash a$,
- $(X \vdash Y \;\&\; Y \vdash c\,) \Rightarrow X \vdash c$.

Here $X \vdash Y$ is the abbreviation for $X \vdash b$ for every $b \in Y$. Thus $X \vdash \emptyset$ is vacuously true. Note that the information systems we consider here are not exactly the same as those introduced by Scott. We do not assume a distinguished element Δ, standing for *true*. To compensate for this, we require that X is non-empty when we write $X \vdash a$. This has the effect that the bottom element of the corresponding domain is always the empty set.

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

- consistent: $X \subseteq^{fin} x \Rightarrow X \in Con$, and
- deductively closed: $X \subseteq x \;\&\; X \vdash a \Rightarrow a \in x$.

Let $X \dashv\vdash Y$ be the abbreviation for $X \vdash Y$ and $Y \vdash X$. For technical convenience, we only consider information systems which are *antisymmetric* in this paper:

$$\forall a, b \in A\ [\{a\} \dashv\vdash \{b\} \Rightarrow a = b].$$

Definition 3.1 *Let $\underline{A} = (A, Con, \vdash)$ be an information system. A token $a \in A$ is a quasi-prime token if*

$$X \dashv\vdash a \Rightarrow a \in X.$$

We write A^q for the set of quasi-prime tokens of $\underline{A}$. The information system is called quasi-prime generated iff for each $a \in A$, there is a finite set $X \subseteq A^q$ such that $X \dashv\vdash \{a\}$.

Let $\overline{S}$ stand for the deductive closure of any token set S, i.e.

$$\overline{S} = \{a \mid \exists X \subseteq^{fin} S . X \vdash a\}.$$

We first show that for each quasi-prime token a, $\overline{a}$ is a quasi-prime element. For this purpose, let $y = \overline{a} - \{a' \mid a' \vdash a\}$. Then y is again an ideal element. This is because from the assumption that a is a quasi-prime token we know that for any $X \subseteq^{fin} y$ (therefore $a \vdash X$), $X \vdash a$ implies $b = a$. Now let $z \sqsubset \overline{a}$ be an ideal element. It is clear that $a \notin z$. Therefore $z \sqsubseteq y$, and y is the unique element immediately below $\overline{a}$. By Theorem 2.1, $\overline{a}$ is a quasi-prime.

Suppose x is a quasi-prime in the domain determined by a quasi-prime generated information system. That means there is a unique element y covered by x. Let $a \in x - y$. Clearly $\overline{a} \subseteq x$. If $\overline{a} \neq x$, we must have $\overline{a} \subseteq y$, since y is the unique element immediately below x, and every element strictly below x must therefore below y. This would lead to $a \in y$, contradicting our assumption $a \in x - y$. The only alternative is $\overline{a} = x$. Let $X \subseteq x$ be such that $X \vdash a$. We can also assume that propositions in X does not entail each other unless they are the same. If for each $b \in X$, $\overline{b} \subset x$, then $\overline{b} \subseteq y$ for each $b \in X$, and $\bigsqcup\{\overline{b} \mid b \in X\} \subseteq y$. This is impossible because $X \vdash a$.

Therefore $\overline{b} = \overline{a}$ for some $b \in X$, which means, by antisymmetry, $b = a$.

The above two paragraphs show that an ideal element x of an information system is a quasi-prime if and only if $x = \overline{a}$ for some quasi-prime token a.

In general, we have the following theorem, whose proof can be found in [11].

Theorem 3.1 *For each quasi-prime generated information system* $\underline{A}$*,*

$$(\mid \underline{A} \mid, \subseteq)$$

is a quasi-prime algebraic domain. On the other hand, for any quasi-prime algebraic domain D*, there is a quasi-prime generated information system* $\underline{A}$ *such that* $D \cong \mid \underline{A} \mid$ *.*

We remark that this representation theorem can be put in a stronger form: we can require every token of a quasi-prime generated information system to be a quasi-prime.

4 Universality and Amalgamation

A unified theory of universal objects can be found in [3]. The basic theorem is that in any algebroidal category in which all morphisms are monic, the existence of a universal, homogeneous object is equivalent to the amalgamation property of the (finite objects of the) category. For reference purposes we recall some of the relevant definitions.

Let $\mathbf{C}$ be a category where all the morphisms are monic (corresponding to the intuitive notion of one-to-one). Let $\mathbf{C}_f$ be the finite objects of $\mathbf{C}$. An object U of $\mathbf{C}$ is universal in $\mathbf{C}$ (or, $\mathbf{C}$-universal) if for any object A in $\mathbf{C}$, there is an arrow $f : A \to U$. U is homogeneous (or, $\mathbf{C}_f$-homogeneous) if for any finite object A with a pair of arrows $f, g : A \to U$, there is an isomorphism $h : U \to U$ such that $f = h \circ g$. U is saturated if for any A, B of $\mathbf{C}_f$ and arrows $f : A \to U$, $g : A \to B$, there is an $h : B \to U$ such that $h \circ g = f$.

A category $\mathbf{C}$ is said to have the amalgamation property if for any arrows $f_1 : A \to B_1$, $f_2 : A \to B_2$ in $\mathbf{C}$, there are arrows $g_1 : B_1 \to B$, $g_2 : B_2 \to B$ in $\mathbf{C}$ such that the following diagram commute.

$$\begin{tikzcd} A \arrow[r, "f_1"] \arrow[d, "f_2"'] & B_1 \arrow[d, "g_1"] \\ B_2 \arrow[r, "g_2"'] & B \end{tikzcd}$$

The result of Droste and Göbel is based on the notion of an algebroidal category.

Definition 4.1 *An algebroidal category is one which has the following properties:*

It has an initial object,
Every object of the category is a colimit of an ω-chain of finite objects,
Every ω-chain of finite objects has a colimit, and
The number of (up to isomorphism) finite objects is countable.

Theorem 4.1 *(Droste and Göbel) Let $\mathbf{C}$ be an algebroidal category with all morphisms monic. Let $\mathbf{C}_f$ be the full subcategory of finite objects of $\mathbf{C}$. The existence of a $\mathcal{C}$-universal, $\mathbf{C}_f$-homogeneous object is equivalent to the amalgamation property of $\mathbf{C}_f$, which is in turn equivalent to the existence of a $\mathbf{C}_f$ saturated object.*

Note that in various categories of information systems, finite objects are often exactly those with a finite token set.

5 Q-Embeddings

To be able to apply Theorem 4.1, we need to introduce an algebroidal category of quasi-prime generated information systems. The morphisms for this category are q-embeddings.

Definition 5.1 *Let* $\underline{A} = (A, Con_A, \vdash_A)$*,* $\underline{B} = (B, Con_B, \vdash_B)$ *be quasi-prime generated information systems. A function* $f : A \to B$ *is a q-embedding of* $\underline{A}$ *into* $\underline{B}$ *if*

1. f *is one-to-one*;
2. $\forall X \subseteq A \forall a \in A$
   $X \in Con_A \iff f(X) \in Con_B$,
   $X \vdash_A a \iff f(X) \vdash_B f(a)$;
3. $f(A^q) \subseteq B^q$.

We remark that given a q-embedding from $\underline{A}$ to $\underline{B}$, if $f(a)$ is a quasi-prime for some $a \in A$, then a itself must be a quasi-prime. Indeed, suppose $X \dashv\vdash_A a$. Then $f(X) \dashv\vdash_B f(a)$. But $f(a)$ is a quasi-prime; so $f(a) \in f(X)$, which means $a \in X$ for f is one-one.

It is informative to show an example which is a usual embedding on information systems (i.e., that satisfies conditions 1, 2 above) [5, 2] but not a q-embedding.

Example. Let

$$\underline{A} = (\{1,3\}, Con_A, \vdash_A)$$

where Con_A includes $\{1,3\}$, and $3 \vdash_A 1$, and let

$$\underline{B} = (\{1,2,3\}, Con_B, \vdash_B)$$

where Con_B includes $\{1,2,3\}$, and $3 \vdash_B 1$, $3 \vdash_B 2$, and $\{1,2\} \vdash_B 3$. It is clear that $\underline{A}$ embeds (by identity) into $\underline{B}$ in the usual sense, but not in the sense of a q-embedding. This is because $3 \in A^q$, but $3 \notin B^q$.

Proposition 5.1 *Quasi-prime generated information systems with q-embeddings form a category, written as* $\mathbf{Q}$.

We now present several propositions leading to the main conclusion that $\mathbf{Q}$ is algebroidal.

Proposition 5.2 *Colimits exist in* $\mathbf{Q}$ *for* ω*-chains of finite information systems.*

Proposition 5.3 *Every quasi-prime generated information system is the colimit of an* ω*-chain of finite information systems in* $\mathbf{Q}$.

The above propositions, together with the observation that the empty information system is initial, implies the following.

Theorem 5.1 *Quasi-prime generated information systems with q-embeddings form an algebroidal category.*

It is easy to see that, corresponding to q-embeddings, there is a notion of embedding-projection pairs on quasi-prime algebraic domains. These are just the usual embedding-projection pairs with the additional requirement that they preserve quasi-primes.

6 Existence of a Saturated Quasi-Prime Algebraic Domain

The purpose of this section is to show that the finite objects of the category $\mathbf{Q}$ have the amalgamation property. In light of Theorem 4.1 and Theorem 5.1, this means there exists a saturated quasi-prime generated information system. Note that our proof below follows the style of [2].

Let

$$\underline{A} = (A, Con_A, \vdash_A),$$

$$\underline{B}_1 = (B_1, Con_1, \vdash_1),$$

and

$$\underline{B}_2 = (B_2, Con_2, \vdash_2)$$

be finite quasi-prime generated information systems such that $f_1 : \underline{A} \to \underline{B}_1$ and $f_2 : \underline{A} \to \underline{B}_2$ are q-embeddings. By renaming the tokens, it is enough to consider the case where f_i's are inclusions (partial identities) and $A = B_1 \cap B_2$.

Definition 6.1 *We construct the information system*

$$\underline{B} = (B, Con, \vdash)$$

from $\underline{B}_1$ *and* $\underline{B}_2$*, where*

- $B = B_1 \cup B_2$;
- $X \in Con$ *if and only if*

$$\begin{array}{l} \exists Z \exists Y \supseteq X \cap B_1.\ [Y \vdash_1 Z \vdash_2 X \cap B_2] \ \ \textit{or} \\ \exists Z \exists Y \supseteq X \cap B_2.\ [Y \vdash_2 Z \vdash_1 X \cap B_1]; \end{array}$$

- $\vdash = \bigcup_{i \geq 0} \vdash^i$*, where* $\vdash^i$*'s are specified as follows:*

$$\begin{array}{l} \vdash^0 = \{(X, a) \mid X \in Con \ \& \ \ \textit{either } B_1 \cap X \vdash_1 a \\ \qquad\qquad\qquad\qquad\qquad\quad \textit{or } \ B_2 \cap X \vdash_2 a\}, \\ \vdash^{i+1} = \{(X, a) \mid \exists Y.\ X \vdash^i Y \vdash^i a\}. \end{array}$$

Before getting into technical details, we would like to give the reader an intuitive feeling of what is going on. $\underline{B}_1$ and $\underline{B}_2$ are two given structures sharing a common substructure $\underline{A}$. The question is whether $\underline{B}_1$ and $\underline{B}_2$ can both be seen living in a larger structure. The larger structure must respect the already-existing relationships between $\underline{B}_1$, $\underline{B}_2$, and $\underline{A}$. In particular, $\underline{A}$ serves as a "bridge" between $\underline{B}_1$ and $\underline{B}_2$, so some X from B_1 may entail some Y from A, which

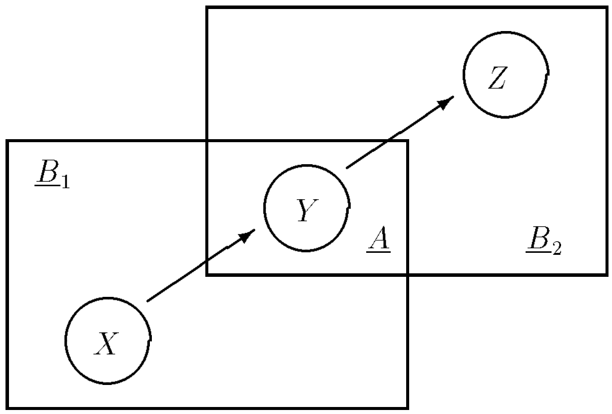

Figure 1: $\underline{A}$ acting as a bridge between $\underline{B}_1$ and $\underline{B}_2$

may in turn entail some Z in B_2. Entailment must be transitive; therefore, in the larger structure we should have X entail Z (see Figure 1). That is why our construction for $\vdash$ is a kind of transitive closure.

The consistency predicate Con, however, must come before $\vdash$ can be defined: the $\vdash$ relation is only defined on consistent sets. On the surface, it may look that there can be several choices for specifying Con. Letting $X \in Con$ if and only if $X \cap B_1 \in Con_1$ and $X \cap B_2 \in Con_2$, for example, may seem to be a reasonable choice. A second thought, however, reveals that this is not the case.

Example. Let $A = \{1, 2\}$, with a trivial consistency predicate and a trivial entailment relation. Let $B_1 = \{\alpha, 1, 2\}$, with $\alpha \vdash_1 1$, and let $B_2 = \{1, 2, a, b\}$ such that $\{a, b\} \notin Con_2$, and $\{1, 2\} \vdash_2 b$, $a \vdash_2 2$. It is easy to check that we have three quasi-prime generated information systems in this way, and A q-embeds into B_1 and B_2. In the bigger system to be constructed, do we want to have $\{\alpha, a\} \in Con$ because $\{\alpha\} \in Con_1$ and $\{a\} \in Con_2$? The answer is no: the given conditions $\alpha \vdash_1 1$, $a \vdash_2 2$, and $\{1, 2\} \vdash_2 b$ would imply $\{\alpha, a\} \vdash$

$\{a, b\}$, which must be inconsistent.

$\square$

Our original specification in Definition 6.1 seems to be the only right choice. We now check that $\underline{B}$ is indeed a quasi-prime generated information system which makes the diagram required by the amalgamation commute. This is achieved by a sequence of lemmas.

Lemma 6.1 *The structure $\underline{B}$ given in Definition 6.1 is an information system.*

Our next lemma is the key to the proofs of various results later.

Lemma 6.2 *For the information system $\underline{B}$ given in Definition 6.1, $X \subseteq B_1$ and $X \vdash Y$ imply*

$$\begin{array}{l} X \vdash_1 Y \cap B_1 \ \ \textit{and} \\ \exists Z \subseteq A.\ X \vdash_1 Z \vdash_2 Y \cap B_2. \end{array}$$

There are a couple of ways Lemma 6.2 can be interpreted. It says that for X's from B_1, $\vdash$ is the same as $\vdash^1$. This means when X is restricted to B_1, $X \vdash Y$ if and only if $X \vdash^1 Y$. It also says that

$$X \subseteq B_1 \ \&\ X \vdash Y \Rightarrow \exists X'.\ X \vdash_1 X' \vdash^0 Y,$$

by taking X' to be $(Y \cap B_1) \cup Z$. These corollaries are sometimes more handy to use.

Lemma 6.3 *For $X \subseteq B_1$ and $a \in B_1$,*

$$X \in Con_1 \Leftrightarrow X \in Con \ \ \textit{and} \ \ X \vdash_1 a \Leftrightarrow X \vdash a.$$

We now come to another important lemma which deals with quasi-primes. It shows that quasi-primes in $\underline{B}_1$ or $\underline{B}_2$ remain to be quasi-primes in $\underline{B}$.

Lemma 6.4 *Let $\underline{B}$ be the information system specified in Definition 6.1. Then every quasi-prime of its component remains to be a quasi-prime, i.e.,*

$$B_1^q \cup B_2^q \subseteq B^q.$$

Note that the equality $B_1^q \cup B_2^q = B^q$ follows easily from this lemma.

As a corollary of the previous lemmas, we have

Theorem 6.1 *$\underline{B}$ is a quasi-prime generated information system. Moreover, both $\underline{B}_1$ and $\underline{B}_2$ q-embed into $\underline{B}$.*

In summary, we have proved

Theorem 6.2 *The finite objects of the category* **Q** *has the amalgamation property.*

As a consequence, we have shown the following.

Theorem 6.3 *There exists a saturated quasi-prime algebraic domain.*

7 A CPO of Quasi-Prime Generated Information Systems

One of the purposes of a universal domain is to ensure the existence of solutions to domain equations for denotational semantics of programming languages. Another way to achieve the same goal is to introduce a cpo of quasi-prime generated information systems using a substructure relation. Various constructions on information systems can be shown to induce continuous functions on the cpo. The existence of the least fixed point for continuous functions then guarantees the existence of solution to domain equations. The advantage of this approach is that the solution is up to equality, rather than up to an isomorphism. We describe this approach now, which uses the idea first introduced in [5].

Definition 7.1 *Let* $\underline{A} = (A, Con_A, \vdash_A)$ *and* $\underline{B} = (B, Con_B, \vdash_B)$ *be quasi-prime generated information systems.* $\underline{A} \trianglelefteq \underline{B}$ *if*

1. $A \subseteq B$
2. $X \in Con_A \iff X \subseteq A \ \&\ X \in Con_B$,
3. $X \vdash_A a \iff X \cup \{a\} \subseteq A \ \&\ X \vdash_B a$, *and*
4. $A^q \subseteq B^q$.

When $\underline{A} \trianglelefteq \underline{B}$ we call $\underline{A}$ a subsystem of $\underline{B}$. Our definition of subsystem is similar to that of Larsen and Winskel [5], with item 4 the only extra requirement. Note that for quasi-prime generated information system $\underline{A}$ and $\underline{B}$, if $A = B$ and $\underline{A} \trianglelefteq \underline{B}$, then $\underline{A} = \underline{B}$.

The relation $\trianglelefteq$ is a complete partial order on the *class* of quasi-prime generated information systems.

Theorem 7.1 *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*

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

Write **CPO** for the class of quasi-prime generated information systems under $\trianglelefteq$. **CPO** is not a cpo in the usual sense simply because they are not a set but a class. The subsystem relation $\trianglelefteq$ can be easily extended to $n-$tuples in the same way as described in [8]. A useful observation is that a 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 $\bigcup_i F\,(\,\underline{A}_i\,)$.

Many constructions can be introduced on quasi-prime generated information systems such as sum, product, lifting, function space, and even quasi-linear function space.

In the rest of the section we illustrate that function space $\rightarrow$ corresponds to a continuous operation

$$\rightarrow:\ \mathbf{CPO}^2 \rightarrow \mathbf{COP}.$$

Other constructions can be shown, in a similar way, to induce continuous operations on **CPO**.

The function space construction is given as follows.

$$\begin{aligned}
&\underline{A} \rightarrow \underline{B} = (Con_A \times B^q, Con_{A\rightarrow B}, \vdash_{A\rightarrow B}), \quad \text{where}\\
&\qquad X \in Con_{A\rightarrow B} \ \text{ iff } \ \bigcup \pi_A(X) \in Con_A \Rightarrow \pi_B(X) \in Con_B,\\
&\qquad X \vdash_{A\rightarrow B} (u,b) \text{ iff } \ \{b' \mid \exists u'.u \vdash_A u' \ \&\ (u',b') \in X\} \vdash_B b
\end{aligned}$$

Note that $\rightarrow$ preserves quasi-prime generated information systems. $\rightarrow$ is monotonic in its first argument. Suppose $\underline{A} \trianglelefteq \underline{A'}$. Write

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

and

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

We check condition 4 in Definition 7.1, to show that $\underline{C} \trianglelefteq \underline{C'}$. However, this is trivial because every token of the function space is a quasi-prime.

Let

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

be a chain of quasi-prime generated information systems. Let (u, a) be a token of $[(\bigcup_i \underline{A_i}) \rightarrow \underline{B}]$. Then clearly (u, a) is a token of $[\underline{A_j} \rightarrow \underline{B}]$ for some j, and thus is a token of $\bigcup_i [\underline{A_i} \rightarrow \underline{B}]$. From this we can deduce that $\rightarrow$ is continuous in its first argument. By a similar but easier proof we get that $\rightarrow$ is continuous in its second argument hence it is continuous.

As an application, we illustrate how to find a solution to the equation

$$X = X_{\uparrow} \rightarrow X$$

within quasi-prime generated information systems. Here $(\ )_{\uparrow}$ is the lifting construction specified as follows. Its use here makes sure that a non-trivial solution is obtained.

Let $\underline{A} = (A, Con, \vdash)$ be a quasi-prime generated information system. Define the *lift* of $\underline{A}$ to be $\underline{A}_{\uparrow} = (A', \vdash')$, where

- $A' = (\{0\} \times A) \cup \{0\}$,
- $X \vdash' Y \Leftrightarrow [\, 0 \in Y \ \text{ or } \ \{c \mid (0, c) \in X\} \vdash_{\underline{A}} \{b \mid (0, b) \in Y\}\,]$.

It is easy to show that lifting preserves quasi-prime generated systems. It is an operation which, given a structure, produces a new one by joining a new token weaker than all the old ones. One can easily check that it gives a continuous operation.

Since the composition of continuous functions remains continuous, and any continuous function $F(x, y)$ of two variables gives rise to a continuous function $F(x, x)$ of one variable, the operation

$$X \longmapsto [X_{\uparrow} \rightarrow X]$$

is a continuous operation on quasi-prime generated information systems. It has a least fixed point

$$\underline{A} = \underline{A}_{\uparrow} \rightarrow \underline{A}.$$

This can be used as a model for the un-typed lambda calculus.

Acknowledgments Thanks go to Michael Huth for pointing out to me that, I had, as a by-product of considering quasi-prime algebraic domains, rediscovered a notion of completely irreducible elements. I am grateful to Michael Mislove who provided travel support for me to present the results in the 9th International Conference on the Foundations of Programming Semantics.

References

[1] Droste, M., and Göbel, R. Universal domains in the theory of denotational semantics of programming languages. Proc. of the IEEE 5th annual symposium on logic in computer science, 1990.

[2] Droste, M., and Göbel, R. Universal information systems. *International Journal of Foundations of Computer Science* vol. 1, no. 4, 1991.

[3] Gierz, G., Hofmann, K.H., Keimel, K., Lawson, J.D., Mislove, M., and Scott, D.S. *A Compendium of Continuous Lattices*. Springer-Verlag, 1980.

[4] Gunter, C. Universal profinite domains. *Information and Computation* 72, 1987.

[5] Larsen, K. and Winskel, G. Using information systems to solve recursive domain equations effectively. In *Lecture Notes in Computer Science 173*, 1984.

[6] Scott, D.S. Domains for denotational semantics. In *Lecture Notes in Computer Science 140*, 1982.

[7] Winskel, G. An introduction to event structures. *Lecture Notes in Computer Science 354*. 1988.

[8] Zhang, G.-Q. DI-domains as prime information systems. *Information and Computation* 100, 1992. Also in Lecture Notes in Computer Science 372 (ICALP-1989, Italy).

[9] Zhang, G.-Q. Some monoidal closed categories of domains and event structures *Mathematical Structures in Computer Science*, to appear. Also: Proceedings of the 7th Conference on Mathematical Foundations of Programming Semantics, Pittsburgh, 1991.

[10] Zhang, G.-Q. *Logic of Domains*. Birkhauser, Boston, 1991.

[11] Zhang, G.-Q. Quasi-prime algebraic domains: a linear category of non-linear functions. To appear. 1992.