Information Categories

Abbas Edalat Michael B. Smyth

Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate
London SW7 2BZ
England.

Abstract

"Information systems" have been introduced by Dana Scott as a convenient means of presenting a certain class of domains of computation, usually known as Scott domains. Essentially the same idea has been developed, if less systematically, by various authors in connection with other classes of domains. In previous work, the present authors introduced the notion of an *I-category* as an abstraction and enhancement of this idea, with emphasis on the solution of *domain equations* of the form $D \cong F(D)$, with F a functor. An important feature of the work is that we are *not* confined to domains of computation as usually understood; other classes of spaces, more familiar to mathematicians in general, become also accessible. Here we present the idea in terms of what we call *information categories*, which are concrete I-categories in which the objects are structured sets of "tokens" and morphisms are relations between tokens. This is more in the spirit of information system work, and enables more specific results to be obtained. Following an account of the general theory, several examples are discussed in some detail: Stone spaces (as an "ordinary" mathematical example), Scott domains, SFP domains, and continuous bounded complete domains.

1 Introduction

In [Sco82], Dana Scott introduced a certain category of "information systems", as a convenient means of presenting a well-known class of domains of computation ("Scott domains"). In this paper we are concerned with the problem of characterizing categories of information systems in general, so as to obtain a uniform treatment of many classes of domains and, indeed, of various classes of spaces that are not usually considered as domains of computation. In previous work [ES91a] we have carried out this program in terms of an abstract notion of of *I-category*. Here we present it in terms of what we call *information categories*, which are concrete I-categories in which the objects are structured sets of "tokens" and morphisms are relations between tokens. This is more in the spirit of information system work, and enables more specific results to be obtained. Several new examples are discussed. We have tried to make the paper reasonably self-contained; it is hoped that the informal discussion in the remainder of this introduction will helpful. Some background in domain theory ideas are desirable, however. For an elementary account one might consult [DP90, Ten91]. (Comprehensive textbooks of domain theory are not yet available.)

An information system, in the sense we intend, is a primitive logic with which to represent, or specify, the elements of a domain of computation. As defined by Scott in [Sco82], an information system has the following ingredients: a set L of *tokens*, or *propositions*; a distinguished member Δ of L (the *least informative* member); a collection *Con* of finite subsets of L (the *consistent* sets of propositions); and a binary relation $\vdash\, \subseteq \mathrm{Con} \times L$ (the *entailment* relation). The consistency predicate must satisfy

(i) any subset of a consistent set is consistent,

(ii) every singleton is consistent,

(iii) $u \cup \{A\}$ is consistent whenever u entails A.

Concerning $\vdash$, we have the usual conditions of reflexivity and transitivity for an entailment (or consequence) relation, and also: Δ is entailed by anything.

As to the specification of domain elements, we could proceed by postulating a relation of satisfaction between tokens and elements. Given some minimal assumptions about this relation, we should find that the set of tokens (propositions) true of a particular element is a *consistent theory*, that is, a set which is closed under entailment and is such that each finite subset is a member of Con. Moreover, we should find that elements are partially ordered by the information ordering:

$$x \sqsubseteq y \iff \text{every proposition true of } x \text{ is true of } y.$$

Alternatively— and more typically— we just define the domain $D(I)$ associated with an information system, I, to be the set of consistent theories of I ordered by inclusion. Notice that the least element, $\perp$, of $D(I)$, is the closure under entailment of $\emptyset$ (equally, of Δ).

One may ask: What domains, exactly, can be constructed, or represented, by information systems? The answer is well known. It is useful to recall some of the basic definitions concerning Scott's domains. A *cpo* is a partial order having a least element and having lubs (least upper bounds) of increasing sequences of elements. Note that we allow a partial order to be large i.e. its carrier to be a proper class. An element a of a cpo D is *compact* if $a \sqsubseteq \bigsqcup_i x_i \Rightarrow a \sqsubseteq x_i$ for some i. A cpo D is *ω-algebraic* if the set $\mathcal{K}_D$ of compact elements of D is (at most) countable and every element of D is the lub of an increasing sequence in $\mathcal{K}_D$. A poset P is *bounded-complete* if every subset of P that is bounded above has a lub. A *Scott domain* is a bounded-complete ω-algebraic cpo. The *Scott topology* of an ω-algebraic cpo D is the topology having as subbase the collection of sets $\uparrow a = \{x \mid a \sqsubseteq x\}$ for a compact; notice that this subbase is a base in case D is a Scott domain. The *Lawson topology* of D is the least topology finer than the Scott topology in which the sets $D - \uparrow a$ (a compact) are also open. When the topology is not specified, the Scott topology is assumed. It is easy to show that the domains associated with information systems having countably many tokens can be characterised as the Scott domains (the countability restriction may be dispensed with, but are natural in the context of computing). Moreover, the propositions of a system I directly determine the topology of $D(I)$: for each proposition (token) a, we have the (sub)basic open set of all the consistent theories which contain a.

Since almost every logical system has notions of entailment and consistency satisfying the above conditions (there are significant exceptions, such as relevance logic and linear logic), the examples are legion. We consider just a few simple examples of information systems relevant to computing:

Example 1.1 LOGIC PROGRAMS. A pure logic program P is a set of program clauses, that is, of Horn clauses having exactly one positive literal (see, for example, [Llo87]). Take L (the tokens) to be the set of ground atoms, and let Con be the collection of all finite subsets of L. We want to regard a program clause containing only ground literals as saying that its head is entailed by its body. Thus, we take $\vdash$ to be the least entailment relation which contains all the ground substitution instances of the clauses of P. (Concretely, $\vdash$ is the "reflexive transitive closure " of the set of ground instances of the clauses.) Identifying the Herbrand models of a logic program with the subsets of the Herbrand base, in the usual way, we see that the domain D associated with the information system just defined has for its elements the Herbrand models of P. Usually in logic programming one works just with $\perp_D$ (the least Herbrand model). But D as a whole is certainly of interest as well, though one may want to consider it under its Lawson rather than its Scott topology (see [BV89]).

Example 1.2 BINARY TREES. A binary tree is given by the set of its nodes, regarded as strings in $0, 1$. Thus we can take L to be $\{0,1\}^*$, Con to be $\mathcal{P}_f(L)$ (that is, every finite set of tokens is consistent), and $\vdash$ to be the least entailment containing the instances of:

$$x0 \vdash x \ , \ x1 \vdash x.$$

Example 1.3 LABELLED BINARY TREES. Let A be a set of atoms (labels). We want to define the "data type" of binary trees over A, understood as including both partial and infinite trees. The type would typically be defined recursively by means of a "domain equation" such as:

$$\text{Btree} \cong (A \times \text{Btree} \times \text{Btree})_\Delta \ .$$

Informally, a binary tree consists of a root which is an atom and a left tree and a right tree, or is undefined. The systematic solution of such domain equations is one of the main objects of our work, although treated only briefly in the present paper. Here we just note that a straightforward approach to the solution yields an inductive definition of an information system, as follows:

$$L : \qquad \Delta \in L; \quad \frac{a \in A \quad b, c \in L}{\langle a, b, c\rangle \in L}$$

$$\text{Con} : \quad \frac{a \in A \quad \{b_1, \cdots, b_n\} \in \text{Con} \quad \{c_1, \cdots, c_n\} \in \text{Con}}{\{\langle a, b_i, c_i\rangle \mid 1 \leq i \leq n\} \in \text{Con}}$$

$$\vdash : \qquad \frac{a \in A \quad b_1 \vdash b_2 \quad c_1 \vdash c_2}{\langle a, b_1, c_1\rangle \vdash \langle a, b_2, c_2\rangle}$$

Note that we have omitted various clauses which hold automatically, such as $\{b\} \in \text{Con}$, and $C \vdash \Delta\,(C \in \text{Con})$.

Of the preceding, Example 1.1 is the least typical and Example 1.3 the most typical as illustrations of what is usually done with information systems, in the way of constructing domains of computation.

Several variations on the definition of an information system are possible. For example, one may, like Larsen and Winskel in [LW84], dispense with Δ (in Example 1.1 above we did not bother to specify Δ). A more significant technical simplification can be achieved by requiring, in effect, that *every* compact element of a domain be represented by a token. We arrive at the (inhabited) "propositional languages" of Fourman and Grayson (see [FG82]):

Definition 1.4 A *propositional language* is given by:

- a set L of *tokens*
- a distinguished element $\Delta \in L$
- a pre-order $\vdash$ over L
- a partial binary operation *conjunction* $\wedge$ over L, satisfying:
    1. $c \vdash \Delta$ for all $c \in L$;
    2. $c \vdash a \wedge b \iff c \vdash a \,\&\, c \vdash b$.

Clause (2) of the definition is to be interpreted as implying that $a \wedge b$ exists if and only if, for some c, $c \vdash a$ and $c \vdash b$. Evidently the conjunction is a meet (greatest lower bound) with respect to the entailment pre-order. The *elements* (or *points*, or *models*) of a propositional language P are filters of P, where a *filter* is a subset x of P satisfying:

- $\Delta \in x$
- $a \in x \,\&\, a \vdash b \Rightarrow b \in x$
- $a \in x \,\&\, b \in x \Rightarrow a \wedge b \in x$.

Propositional languages are completely equivalent to the information systems defined above as means of representing domains. (For detailed comparison: a finite consistent set of tokens of a Scott information system is in effect being taken as a single token of a propositional language.) Propositional languages are our preferred notion of information systems, later to be generalised in various ways, and we will sometimes call them *bounded complete information systems*, or even (in this section) *information systems* without qualification.

The representation of continuous functions between domains must also be taken into account. In fact, we have morphisms between information systems given by:

Definition 1.5 Let $I = (L, \vdash, \wedge, \Delta)$ and $I' = (L', \vdash', \wedge', \Delta')$ be bounded-complete information systems. An *approximable mapping* from I to I' is a relation $f \subseteq L \times L'$ satisfying:

1. $\Delta f \Delta'$
2. $afb', afc' \Rightarrow af(b' \wedge c')$
3. $a \vdash b', bfb', b' \vdash' c' \Rightarrow afc'$.

A pair $(a, a') \in f$ may be regarded as a proposition asserting that, if a is true of the argument, then a' is true of the result. This idea leads to the construction of an information system for the "function space" $[I \to I']$, and indeed to the result that the category of (countable) information systems and approximable mappings is cartesian closed. In the present context, the following facts are particularly relevant: the collection of approximable mappings from I to I', ordered by inclusion, is isomorphic with the cpo $[D(I) \to D(I')]$ of continuous maps from $D(I)$ to $D(I')$; and the category of (countable) information systems is equivalent to the category of Scott domains. These statements are verified by well-known methods.

From the preceding one might conclude that information systems are only a matter of presentation (a way to present Scott domains) and that there is little prospect of a distinctive *theory* of information systems. However, a different conclusion begins to emerge when one looks at information systems in a broader context. The broader context is that of the presentation of topological spaces by means of entailment relations between open sets. The entailment relations need not be so restricted in format as those considered above. A natural generalisation is to allow entailments with several terms on the right:

$$a \vdash b_1, \cdots, b_n \tag{1}$$

Here, the b_i's are read disjunctively, and a "model" of (1) has to be such that if it satisfies a then it satisfies at least one b_i. Adapting our propositional languages to allow for such entailments, we would modify the second clause in the definition of *element* to read:

$$a \in x \ \& \ a \vdash b_1, \cdots, b_n \quad \Rightarrow \quad b_i \in x \text{ for some } i.$$

The elements are now, in effect, *prime* filters.

Example 1.6 INFINITE BINARY STRINGS. Let the tokens be the binary strings, i.e. $L = \{0, 1\}^*$. Let the entailments be generated from:

- $a \vdash b$ if b is an initial segment of a;
- $a \vdash a0, a1$.

The conjunction $a \wedge b$ exists iff either $a \vdash b$ or $b \vdash a$, and is then the longer of the two strings. Then the elements, or prime filters, can be identified with the *infinite* binary strings, and the topology is that of Cantor space.

A far-reaching development of these ideas is provided by Fourman and Grayson [FG82]. For our part, we will be content with entailments in the simple form (one antecedent and one consequent), while allowing for disjunction by having it as an operation on tokens. Thus, in the wider context we will typically think of information systems as having the structure of lattices. They may, for example, be Boolean algebras, representing Stone spaces. As a somewhat different example, we may hope to accommodate various kinds of *event structures* (see [Win86]), representing (species of) stable domains; see Section 7 below.

These remarks may tend to suggest that the theory of information systems should be assimilated to that of Stone-like dualities. However this overlooks the most distinctive feature of information systems, as expounded already in [Sco82, LW84] and in the context of the presentation of domains for sequentiality by "concrete data structures" in [BC81]. We refer to the idea that the collection of all information systems (of a given variety) itself has an "information ordering". The importance of this is that domain equations, of the form

$$D \cong F(D), \tag{2}$$

for the recursive specification of types, can then be handled very simply. Indeed, given that F is at least a monotonic operator, ordinary fixed point techniques suffice for the solution of (2). (Example 1.3 above can provide an illustration of this, if one recasts the inductive definition given there as a least fixed point definition.) The ordering of information systems is, typically, defined as follows. Let I, J be information systems, with token sets L, L'. Then $I \trianglelefteq J$ iff $L \subseteq L'$ and the operations and relations of L are the restrictions to L of those of L'. By requiring (as we may, without loss of generality) that the tokens of information systems all be drawn from a common pool of tokens, we find typically that the collection of information systems under $\trianglelefteq$ is a cpo, even an ω-algebraic cpo. How does this compare with more elaborate category-theory approaches, as in [SP82, MA86]? The answer seems clear: the categorical approach gives us a valuable *characterization* of the intended solution as (usually) an *initial algebra*; the simple order-theoretic method provides no such information. This gives the clue to what we have been aiming at in our work: to find out what needs to be added to the information system method so as to capture the advantages of the systematic category-theoretic approaches.

The key idea is that a global information ordering of *morphisms* $\trianglelefteq^m$, is needed, in addition to (or, including) that for objects. Concretely, we can define:

$$(f : I_1 \to I_2) \trianglelefteq^m (g : J_1 \to J_2) \text{ if } I_n \trianglelefteq J_n (n = 1, 2) \text{ and } f \subseteq g$$

(note that f, g are sets of pairs of tokens). All the required theory of domain equations, at least as far as the (effective) initial algebra theorem, can be developed in this general context (and indeed at the more abstract level of I-categories [ES91a]) and is then available to be applied routinely to various concrete situations, such as Scott domains, stable domains, Stone spaces, and even metric spaces. Various constructions, such as the functor category of two categories of information systems, can also be handled.

The remainder of the paper is organized as follows. In Section 2, we will recall the various notions of I-categories from [ES91a]. In Section 3, we present the general theory of information categories. In Section 4, we show that the category of Boolean algebras and Boolean homomorphisms is an instance of an information category in which we will solve a domain equation for Stone spaces whose solution is the well known Cantor space. In the subsequent sections we construct information categories, or in other words categories of information systems, equivalent to the categories of Scott domains, SFP objects, dI-domains and continuous bounded complete domains; in each case we will construct the function space explicitly.

2 Preliminaries: I-categories

An *I-category* $(P, \mathrm{Inc}, \sqsubseteq, \Delta)$ consists of

- a category P with a partial order $\sqsubseteq^{A,B}$ on each homset $\hom(A, B)$,

- a subclass Inc $\subseteq$ Mor, called the *inclusion morphisms* of P, such that in each hom-set, $\hom(A, B)$, there is at most one inclusion morphism which we denote by $\mathrm{in}(A, B)$ or $A \rightarrowtail B$,
- a distinguished object $\Delta \in \mathrm{Obj}$,

satisfying the following two axioms:

Axiom 1 (i) *The class of objects Obj and the inclusion morphisms Inc form a partial order represented as a category.*

(ii) *$in(\Delta, A)$ exists, for all $A \in Obj$ and $in(\Delta, A) \sqsubseteq f$ for all morphisms $f \in \hom(\Delta, A)$.*

(iii) *$f; in(A, B) \sqsubseteq g; in(A, B) \Rightarrow f \sqsubseteq g$, for all $f, g \in Mor, in(A, B) \in$ Inc, such that the compositions are defined.*

Axiom 2 *Composition of morphisms is monotone with respect to the partial order on hom-sets, i.e.*

$$f_1 \sqsubseteq f_2 \ \&\ g_1 \sqsubseteq g_2 \ \Rightarrow\ f_1; g_1 \sqsubseteq f_2; g_2$$

whenever the compositions are defined.

The partial orders $\trianglelefteq$ on Obj_P and $\trianglelefteq^m$ on Mor_P of an I-category $(P, \mathrm{Inc}, \sqsubseteq, \Delta)$ are defined as follows:

- $A \trianglelefteq B$ if $\mathrm{in}(A, B)$ exists;
- $f \trianglelefteq^m g$ if
  - (i) $\mathrm{dom}(f) \trianglelefteq \mathrm{dom}(g)$,
  - (ii) $\mathrm{cod}(f) \trianglelefteq \mathrm{cod}(g)$,
  - (iii) $f; \mathrm{in}(\mathrm{cod}(f), \mathrm{cod}(g)) \sqsubseteq \mathrm{in}(\mathrm{dom}(f), \mathrm{dom}(g)); g$.

Note that $f \trianglelefteq^m g$ iff the diagram

$$\begin{tikzcd}
\mathrm{dom}(g) \arrow[r, "g"] & \mathrm{cod}(g) \\
\mathrm{dom}(f) \arrow[u] \arrow[r, "f"'] \arrow[ur, phantom, "\sqsupseteq"] & \mathrm{cod}(f) \arrow[u]
\end{tikzcd}$$

weakly commutes.

A *complete* I-category further satisfies the following axioms:

Axiom 3 *$(Mor, \trianglelefteq^m)$ is a cpo.*

Axiom 4 *$(Inc, \trianglelefteq^m)$ is a subcpo of $(Mor, \trianglelefteq^m)$.*

Axiom 5 *Composition of morphisms is a continuous operation with respect to $\trianglelefteq^m$, i.e. $\bigsqcup_i (f_i; g_i) = (\bigsqcup_i f_i); (\bigsqcup_i g_i)$ whenever $\langle f_i \rangle_{i \geq 0}$ and $\langle g_i \rangle_{i \geq 0}$ are increasing chains in $(Mor, \trianglelefteq^m)$ with $cod(f_i) = dom(f_i)$, for all $i \geq 0$.*

Finally, an *ω-algebraic* I-category satisfies two more axioms:

Axiom 6 *$(Mor, \trianglelefteq^m)$ is ω-algebraic.*

Axiom 7 (i) *$f \in (Mor, \trianglelefteq^m)$ is compact $\Rightarrow$ $cod(f) \in (Obj, \trianglelefteq)$ is compact.*

(ii) *$A, B \in (Obj, \trianglelefteq)$ are compact with $A \trianglelefteq B$ $\Rightarrow$ $in(A, B) \in (Mor, \trianglelefteq^m)$ is compact.*

(iii) *The composition of compact morphisms is compact.*

Any (ω-algebraic, complete) partial order $(Q, \sqsubseteq)$ with least element $\perp$, considered as a category in the usual way, is an example of an (ω-algebraic, complete) I-category $(Q, \mathrm{Inc}, =, \perp)$ with $\mathrm{Inc} = \mathrm{Mor}$, the discrete partial order $=$ on the homsets and distinguished object $\Delta = \perp$. The category, **Sets**, of sets and functions is in fact a complete (large) I-category $(\mathbf{Sets}, \mathrm{Inc}, =, \emptyset)$ where the inclusion morphisms are simply the set inclusions; any full subcategory whose objects have elements from a countable pool is in fact an ω-algebraic I-category. In [ES91a], it is shown that the category of Scott information systems and approximable maps is also a complete I-category which like the category of sets has an ω-algebraic subcategory. Further examples of I-categories are presented in the above paper. However, in order to capture the general framework of all these examples we need to introduce information categories in the next section.

3 Information categories

In practice, I-categories are often *concrete* in the sense that their objects are sets with some internal structure given by operations and predicates defined on the elements of the sets or on their finite subsets, i.e. they are weak second structures in the terminology of [Bar77]; the partial order on objects correspond to the substructure relation between objects; and morphisms are relations between elements or finite subsets of the carrier sets of objects. We will call these concrete I-categories *information categories*, which like the abstract I-categories can be complete or ω-algebraic. In information categories the partial order on objects $A \trianglelefteq B$ corresponds to the notion that A is a substructure of B; whereas the partial order on morphisms $f \trianglelefteq^m g$ simply reduces to $f \subseteq g$ i.e. the inclusion of relations. Similarly in complete information categories the lub of a chain of objects will be the union of the chain of structures and the lub of a chain of morphisms will simply be the set union of the relations representing the morphisms. Finally in ω-algebraic information categories compact objects will be precisely the finite objects i.e. objects with a finite carrier set and the compact morphisms will be precisely the relations between finite objects. These features make information categories conceptually simple, and easy to handle; the task of verifying that a certain category is an information category and therefore an I-category becomes quite straightforward. As a further justification to introduce these information categories we note that all the basic examples of concrete I-categories, treated later in this section or in future papers, are fully covered by these notions. We will now formalise the definition of information categories; we start with the notion of a weak second order structure.

Definition 3.1 A weak second order structure is a tuple $A = (|A|, T^i, f^j)$, where $|A|$ is a set, and T^i and f^j ($i \in I, j \in J$, and I and J are indexing sets) are respectively predicates and (partial or total) operations on $|A|$ or on $\mathcal{P}_f(|A|)$.

One also has the notion of substructures for weak second order structures.

Definition 3.2 Given two weak second order structures of the same signature $A = (|A|, T^i_A, f^j_A)$ and $B = (|B|, T^i_B, f^j_B)$ ($i \in I$ and $j \in J$), we say that A is a substructure of B, denoted by $A \trianglelefteq B$, if for all $i \in I$ and $j \in J$ we have

(i) $|A| \subseteq |B|$;

(ii) $T^i_A(\vec{x}) \iff \vec{x} \in \tilde{A}^{k_i} \;\&\; T^i_B(\vec{x})$,
(where $\tilde{A}$ is $|A|$ or $\mathcal{P}_f(|A|)$ and k_i is the arity of T^i);

(iv) $f^j_A(\vec{x})$ exists in A iff $f^j_B(\vec{x})$ exists in B, in which case:
$y = f^j_A(\vec{x}) \iff \vec{x} \in \tilde{A}^{h_j}$ & $y = f^j_B(\vec{x})$,
(where h_j is the arity of f_j).

In an information category the identity morphism, I, of an object must of course satisfy $I;I = I$, and hence, as a binary relation between the elements or finite subsets of the carrier set of that object, it must be a transitive and interpolative order (a relation $R : A \to A$ is transitive iff $R;R \subseteq R$, it is interpolative iff $R \subseteq R;R$). It is therefore convenient to assume that there is a distinguished transitive and interpolative relation amongst the predicates of an object of an information category. Note that this relation is usually on the carrier set itself and in many cases it turns out to be a pre-order or a partial order and sometimes even the equality relation. We are now ready to define an information category.

Definition 3.3 An *information category*, $(C, <, T^i, f^j, \mathrm{Inc}, \subseteq, \Delta)$, $(i \in I$, $j \in J$, where I, J are indexing sets), is a category C with the following properties:

- Objects of C are weak second order structures of fixed signature of the form $A = (|A|, <_A, T^i_A, f^j_A)$, where $<_A$ is a transitive-interpolative relation on $|A|$ (or $\mathcal{P}_f(|A|)$) and the elements of $|A|$ are from a fixed alphabet.
- Morphisms of C are relations between elements (or finite sets) of the objects with the usual composition rule for relations, which satisfy the following:
  - **C1** The relation $<_A$ is the identity morphism on A for all $A \in \mathrm{Obj}$.
- The *inclusion morphisms* Inc $\subseteq$ Mor, satisfy:
  - **C2** Whenever an object A is a substructure of an object B, the relation $<_A;<_B \subseteq |A| \times |B|$, with $a\,(<_A;<_B)\,b$ iff $\exists a' \in |A|.\, a <_A a'$ & $a' <_B b$, is a morphism and the unique inclusion morphism in $\hom(A, B)$.
  - **C3** For all objects A, $\mathrm{in}(\Delta, A)$ exists and $\mathrm{in}(\Delta, A) \subseteq f$ for all $f \in \hom(\Delta, A)$.
- Hom-sets are partially ordered by subset inclusion $\subseteq$.

We will prove shortly that an information category as defined above is in fact an I-category and that the partial order on morphisms reduces to set inclusion of relations. Before that however a couple of remarks on **C2** are appropriate. First observe that the way inclusion morphisms are defined implies that the partial order on objects induced by inclusion morphisms coincides with the substructure relation between weak second order structures and therefore the use of $\trianglelefteq$ for both of these notions is consistent. Also note that in any I-category $A \trianglelefteq B$ implies that $\mathrm{Id}_A \trianglelefteq^m \mathrm{in}(A,B) \trianglelefteq^m \mathrm{Id}_B$, which in the case of information categories reduces to $<_A \subseteq \mathrm{in}(A,B) \subseteq <_B$. We also have $<_A \subseteq <_A; <_B \subseteq <_B$, which therefore motivates the requirement in **C2** that $\mathrm{in}(A,B) = <_A; <_B$. Note that when the transitive-interpolative orders in the objects are pre-orders, we have $<_A; <_B = <_B \cap (A \times B)$, as can be easily checked. We now have the expected result:

Proposition 3.4 *An information category is an I-category. Moreover, the induced partial orders $\trianglelefteq$ and $\trianglelefteq^m$ on Obj and Mor satisfy: $(f : A \to B) \trianglelefteq^m (g : C \to D)$ iff $A \trianglelefteq C$ and $B \trianglelefteq D$ and $f \subseteq g$.*

Proof First note that if $A \trianglelefteq B \trianglelefteq C$, we get

$$\mathrm{in}(A,B); \mathrm{in}(B,C) = <_A; <_B; <_B; <_C = <_A; <_B; <_C = <_A; <_C = \mathrm{in}(A,C),$$

from which Axiom 1(i) follows. Next observe that for any morphism $f : A \to B$ and inclusion morphisms $\mathrm{in}(C,A)$ and $\mathrm{in}(B,D)$, it follows from **C1** and **C2** that $\mathrm{in}(C,A); f = <_C; <_A; f = <_C; f$ and $f; \mathrm{in}(B,D) = f; <_B; <_D = f; <_D$. To prove Axiom 1(iii), suppose $f; \mathrm{in}(B,D) \subseteq g; \mathrm{in}(B,D)$. Then $f; <_D \subseteq g; <_D$ and we have

$$afb \Rightarrow \exists b' \in |B|.\ afb' <_B b \Rightarrow afb' <_D b \Rightarrow a(f; <_D)b \Rightarrow a(g; <_D)b \Rightarrow agb.$$

To prove the second part, suppose $f; \mathrm{in}(B,D) \subseteq \mathrm{in}(A,C); g$, i.e. $f; <_D \subseteq <_A; g$, then

$$afb \Rightarrow \exists b' \in |B|.\ afb' <_B b \Rightarrow afb' <_D b \Rightarrow a(f; <_D)b \Rightarrow a(<_A; g)b \Rightarrow agb.$$

Suppose next that $f \subseteq g$, then

$$a(f; <_D)d \Rightarrow \exists a' \in |A|\ \exists b \in B.\ a <_A a' f b <_D d \Rightarrow a <_A a' g b <_D d \Rightarrow a <_A a' g d \Rightarrow a(<_A; g)d.$$

This completes the proof. $\Box$

Example 3.5 The category **Sets** is an information category, whereas a partial order is an example of an I-category which is not an information category.

In order to define a complete information category, we first need to define the union of a chain of structures.

Definition 3.6 Let $\langle A_k \rangle_{k \geq 0}$ with $A_k = (|A_k|, T^i_{A_k}, f^j_{A_k})$ be an increasing chain of weak second order structures; then the union of the chain is the structure given by $\bigcup_k A_k = (\bigcup_k |A_k|, \bigcup_k T^i_k, \bigcup_k f^j_k)$, where, for all $i \in I$, $j \in J$,

- $(\bigcup_k T^i_k)(\vec{x}) \iff \exists k.\, \vec{x} \in \tilde{A}^{l_i}\ \&\ T^i_k(\vec{x})$
  (where $\tilde{A}$ is $|A|$ or $\mathcal{P}_f(|A|)$ and l_i is the arity of T^i);
- $y = \bigcup_k f^j_k(\vec{x}) \iff \exists k.\, \vec{x} \in \tilde{A}^{h_j}\ \&\ y = f^j_k(\vec{x})$
  (where h_j is the arity of f_j).

Definition 3.7 An information category is *complete* if:

C4 (Obj, $\trianglelefteq$) and (Mor, $\trianglelefteq^m$) are complete in the sense that

(i) whenever $\langle A_k \rangle_{k \geq 0}$ is a chain of objects of C then $\bigcup_k A_k$ is also an object;

(ii) whenever $\langle g_k \rangle_{k \geq 0}$ is an increasing chain of morphisms of C with $\mathrm{dom}(g_k) = A_k$ and $\mathrm{cod}(g_k) = B_k$ then $\bigcup_k g_k$ is also a morphism, where

  * $\mathrm{dom}(\bigcup g_k) = \bigcup \mathrm{dom}(A_k)$ $\qquad$ $\mathrm{cod}(\bigcup g_k) = \bigcup \mathrm{cod}(B_k)$;
  * $(\forall a \in \bigcup A_k)(\forall b \in \bigcup B_k)[a(\bigcup g_k)b \iff \exists k.\, a g_k b]$.

Proposition 3.8 *A complete information category is a complete I-category.*

Proof It is clear that an increasing chain of morphisms $\langle g_i \rangle_{i \geq 0}$ in a complete information category has lub $\bigcup g_i$ as defined in **C4**(ii) above. Therefore, Mor is a cpo. Given an increasing chain of inclusion morphisms $\langle \mathrm{in}(A_i, B_i) \rangle_{i \geq 0}$, one readily finds by **C2** and **C4**(i) that $\mathrm{in}(\bigcup A_i, \bigcup B_i)$ exists; it is indeed the lub of the chain since:

$$\begin{array}{ccccc} a(\bigcup \mathrm{in}(A_i, B_i))b & \iff & \exists i \geq 0.\ a(\mathrm{in}(A_i, B_i))b & \iff \\ \exists i \geq 0.\ a(<_{A_i}; <_{B_i})b & \iff & \exists i \geq 0\, \exists a' \in |A_i|\, .\ a <_{A_i} a' <_{B_i} b & \iff \\ \exists i \geq 0\ .a(<_{\bigcup A_i}; <_{\bigcup B_i})b & \iff & a(\mathrm{in}(\bigcup A_i, \bigcup B_i))b. & \end{array}$$

This shows that Inc is a subcpo of Mor. Finally, if $\langle (f_i : A_i \to B_i) \rangle_{i \geq 0}$ and $\langle (g_i : B_i \to C_i) \rangle_{i \geq 0}$ are two increasing chains of morphisms whose elements are pairwise composable, then $\bigcup f_i$ and $\bigcup g_i$ are composable as the mappings dom and cod are continuous by **C4**. Moreover, putting $A = \bigcup A_i, B = \bigcup B_i$ and $C = \bigcup C_i$, we have for all $a \in A$ and $c \in C$:

$$\begin{array}{ccccc} a(\bigcup(f_i; g_i))c & \Longleftrightarrow & \exists i \geq 0.\, a(f_i; g_i)c & \Longleftrightarrow \\ \exists i \geq 0\, \exists b \in B.\, af_ib \,\&\, bg_ic & \Longleftrightarrow & \exists b \in B.\, a(\bigcup f_i)b \,\&\, b(\bigcup g_i)c & \Longleftrightarrow \\ a((\bigcup f_i); (\bigcup g_i))c. & & & \end{array}$$

We conclude that composition of morphisms is a continuous operation and the proof is complete.□

When the transitive and interpolative relation in each object of an information category is in fact a pre-order, we have an information category with pre-ordered objects. We will define the notion of ω-algebraicity for such categories.

Definition 3.9 A complete information category with pre-ordered objects, where the pre-order is denoted by $\lesssim$, is *ω-algebraic* if it satisfies:

C5 Every morphism is the union of a countable chain of morphisms between finite objects (i.e. objects with finite carrier sets).

C6 For all objects A and all elements $a \in A$ the set $E_A(a) = \{x \in A \mid x \equiv_A a\}$ is finite and $E_A(a) = E_B(a)$ whenever $A \trianglelefteq B$. ($x \equiv_A a$ means $x \lesssim_A a$ and $a \lesssim_A x$.)

Condition **C6** ensures that infinite objects, i.e. objects with infinite carrier sets, do represent infinite information and it excludes, for example, an infinite object with all its elements equivalent to each other. This condition is trivially satisfied when the pre-order is in fact a partial order. For this reason it is convenient to use partial orders in the construction of ω-algebraic information categories and this is what we will do in all the examples in this paper, although equivalent categories with pre-orders satisfying **C6** can also be constructed. We, again, have the expected result:

Proposition 3.10 *An ω-algebraic information category with pre-ordered objects is an ω-algebraic I-category.*

Proof Consider an ω-algebraic information category. We must show that $(\mathrm{Mor}, \trianglelefteq^m)$ is ω-algebraic and that Axiom 7 holds. We claim that a morphism is compact iff its domain and codomain are finite objects. Suppose then that $f \in \mathrm{Mor}$ is compact. By **C5**, f is the lub of an increasing chain of morphisms between finite objects. But by compactness, f must itself be equal to one of these morphisms and hence have a finite domain and a finite codomain. Conversely, let $(f : A \to B)$ be a morphism with finite objects A and B, and suppose $(f : A \to B) \trianglelefteq^m \bigcup(f_i : A_i \to B_i)$ then by Proposition 3.4 and the continuity of dom and cod we get: $A \subseteq \bigcup A_i$, $B \subseteq \bigcup B_i$ and $f \subseteq \bigcup f_i$. Since A, B and, consequently, f (which is a subset of $A \times B$) are all finite sets, we can find $i \geq 0$ with $A \subseteq A_i$, $B \subseteq B_i$ and $f \subseteq f_i$, i.e. $f \trianglelefteq^m f_i$ and hence f is compact. This proves our claim. Since objects of the category are sets from a countable alphabet, the finite objects and, hence, the morphisms between them, i.e. the compact morphisms, are countable. This together with **C5** shows that $(\mathrm{Mor}, \trianglelefteq^m)$ is ω-algebraic. Axiom 7 follows immediately from the claim. □

3.1 Completeness and ω-algebraicity

In practice, in order to verify that a category is an ω-algebraic information category, the completeness condition (**C4**) and the ω-algebraicity condition (**C5**) are the only non-trivial conditions to check . We will now show that in concrete cases these two conditions can be verified by quite general methods. Assume that the objects and morphisms of an information category are models of a weak second order theory, i.e. they are given by a finite number of weak second order axioms where quantification can be carried out both over the elements and over the finite subsets of the carrier sets of the objects. In this context, two objects A and B satisfy $A \trianglelefteq B$ iff A is a submodel of B iff A is a model of the theory and also a substructure of B. We now discuss the question of completeness and ω-algebraicity separately.

Completeness: We present some simple model-theoretic results concerning the type of axioms for the objects and morphisms of an information category with which the completeness condition **C4** will be guaranteed to hold. Recall that in first order i.e. ordinary model theory, where all predicates and operations are defined on the elements of the carrier sets of models, a first order axiom satisfied by all the models of a chain is preserved in the union of the chain if and only if the axiom is equivalent to a universal-existential (u.e.) axiom, i.e. one which in prenex normal form has precisely a number of universal quantifiers followed by a number of existential quantifiers. See, for example, [Gra79]. We would like to extend this result to information categories where the predicates and operations in an object are defined on the elements or the finite subsets of the carrier set of that object, i.e. to extend the result from first order to weak second order model theory. For convenience, we use small letters for quantification over elements and capital letters for quantification over finite subsets. We have a distinguished predicate of mixed argument namely $\in \subseteq A \times \mathcal{P}_f(|A|)$ such that (a, S) is in $\in$ iff $a \in S$; we also have, in all objects A, a constant $\emptyset \in \mathcal{P}_f(|A|)$, which satisfies $\forall a. \neg(a \in \emptyset)$.

It can be easily shown by extending the proof in the case of first order model theory that in weak second order theory a u.e. axiom, where each quantification is now over A or $\mathcal{P}_f(A)$, is preserved in the union of a chain. We conclude that any formula which is equivalent to a u.e. formula will be preserved in the union. The argument can be extended to morphisms as follows. A morphism $f : A \to B$ in an information category can be regarded as a weak second order model with two basic sorts A and B, and two distinguished predicates $\exists_A \subseteq A \times \mathcal{P}_f(A)$ and $\exists_B \subseteq B \times \mathcal{P}_f(B)$. This model is characterised by weak second order axioms for the objects A and B, and also for the relation f. If all these axioms are equivalent to u.e. axioms, then the category contains all unions of chains of morphisms, i.e. the completeness condition **C4** will be satisfied.

ω-algebraicity: The conditions for ω-algebraicity (**C5**) can be reduced to a simple criterion. We say that a weak second order model is *locally finite* if any finite subset of the carrier set of the model is contained in the carrier set of a finite submodel of that model; we also say that a complete information category with pre-ordered objects is *locally finite* if its morphisms (and hence its objects) are locally finite as weak second order models. Now suppose we have a complete information category P with pre-ordered objects which satisfies **C6** and whose objects have their elements from a countable pool. We claim that P is ω-algebraic iff it is locally finite. For assume that P is ω-algebraic and let $f : A \to B$ be any morphism of P with $A_0 \in \mathcal{P}_f(A)$, $B_0 \in \mathcal{P}_f(B)$ and $f_0 \subseteq f \cap (A_0 \times B_0)$. Then by **C5** there exist morphisms f_i between finite objects A_i and B_i $(i \geq 1)$ such that $f = \bigcup_i (f_i : A_i \to B_i)$. We can now find $i \geq 1$ with $A_0 \subseteq A_i$ and $B_0 \subseteq B_i$ which implies that $f_0 \subseteq f_i$. Hence P is locally finite. In the other direction assume that P is locally finite and let $f : A \to B$ be any morphism of P. Since the carrier sets $|A|$ and $|B|$ are both countable, we can choose finite sets M_i and N_i $(i \geq 1)$ with $|A| = \bigcup M_i$ and $|B| = \bigcup N_i$ and put $g_i = f \cap (M_i \times N_i)$. Define $f_i : A_i \to B_i$ inductively

as follows: Let $f_1 : A_1 \to B_1$ be a finite submodel of $f : A \to B$ with $g_1 \subseteq f_1$, $M_1 \subseteq |A_1|$ and $N_1 \subseteq |B_1|$. Assuming $f_i : A_i \to B_i$ has been defined, let $f_{i+1} : A_{i+1} \to B_{i+1}$ be a finite submodel of $f : A \to B$ with $(f_i \cup g_{i+1}) \subseteq f$, $(|A_i| \cup M_{i+1}) \subseteq |A_{i+1}|$, and $(|B_i| \cup N_{i+1}) \subseteq |B_{i+1}|$. Then $f = \bigcup f_i$ and hence P is ω-algebraic. This proves the claim.

Finally we make the following simple remark. Let us say that an information category is *finitary* if its objects are all finite. Then it is clear that the chain completion of any finitary information category, by simply adding the lubs of increasing chains of objects and morphisms to the category, gives rise to an ω-algebraic information category, and, conversely, every ω-algebraic information category can be obtained by taking the chain completion of the (full) subcategory of its finite objects.

3.2 The Initial Algebra Theorem

Given an endofunctor $F : C \to C$ on a category C, the category of *F-algebras* has as objects the pairs (A, f), with $A \in \mathrm{Obj}_C$ and $f \in \hom(F(A), A)$, and as morphisms, between objects (A, f) and (B, g), those $h \in \hom(A, B)$ for which the following diagram:

$$\begin{tikzcd} A \arrow[d, "h"'] & F(A) \arrow[l, "f"'] \arrow[d, "F(h)"] \\ B & F(B) \arrow[l, "g"] \end{tikzcd}$$

commutes. An *initial algebra* of F is defined to be an initial object, if it exists, of the category of F-algebras; and we then say that F has an initial algebra, or a least fixed point, in C. An initial F-algebra (A, f) as above gives a canonical solution of the domain equation $X \cong F(X)$; hence its importance in computing science. (see, for example, [MA86] for details).

We now recall some definitions from [ES91a]. A morphism $f : A \to B$ of an I-category K is *strict* if $\mathrm{in}(\Delta, A); f = \mathrm{in}(\Delta, B)$. We can immediately see that Δ is an initial object for the subcategory of strict morphisms K^s. A functor between I-categories is *standard* if it preserves inclusion morphisms; a functor between complete I-categories is *object-continuous* (*morphism-continuous*) if the induced function on the cpo of objects (morphisms) is continuous. In [ES91a], we showed that every standard and morphism-continuous endofunctor on a complete I-category K has an initial algebra in K^s, and that for an effectively given ω-algebraic I-category the initial algebra is effectively given. In fact for the general (i.e. non-effective) theory, we can do better using the following result in [SP82], which is known as the "basic lemma".

Lemma 3.11 *Let K be a category with initial object $\perp$ and let $F : K \to K$ be a functor. Define the ω-chain T to be*

$$\perp \xrightarrow{\quad m \quad} F(\perp) \xrightarrow{F(m)} F^2(\perp) \xrightarrow{F^2(m)} F^3(\perp) \xrightarrow{\qquad} \cdots$$

where m is the unique morphism from $\perp$ to $F(\perp)$. Suppose that both $\mu : T \to A$ and $F(\mu) : F(T) \to F(A)$ are colimiting cones. Then the initial F-algebra exists and is (A, α) where $\alpha : F(A) \to A$ is the unique mediating morphism from $F(\mu)$ to μ.

Theorem 3.12 *A standard object-continuous endofunctor $F : K \to K$ on a complete I-category has an initial algebra in K^s.*

Proof Consider the chain T as in the basic lemma:

$$\Delta \rightarrowtail F(\Delta) \rightarrowtail F^2(\Delta) \rightarrowtail F^3(\Delta) \rightarrowtail \cdots$$

By lemma 3.6 in [ES93b], the lub of any ω-chain is a colimit in K^s. It follows that both $\mu : T \to D$ and $F(\mu) : F(T) \to F(D)$, where $D = \bigsqcup_i F^i(\Delta)$ and $\mu_i = \mathrm{in}(T_i, D)$, are colimiting as F is object-continuous. Furthermore, Δ is an initial object in K^s. Therefore the result follows by the basic lemma.□

For information categories, we refer to standard functors as *monotonic* functors; and there is a simple way of characterizing object-continuous functors which generalizes a corresponding result in [LW84]. We say that a functor $F : K_1 \to K_2$ between complete information categories is *continuous on token sets* if whenever $\langle A_k \rangle_{k \geq 0}$ is an increasing chain in K_1, then $|F(\bigcup_k A_k| = |\bigcup_k F(A_k)|$.

Lemma 3.13 *A monotonic functor of complete information categories is object-continuous iff it is continuous on the token sets.*

Proof The "only if" part is evident. For the "if" part, note first from Definition 3.2 of the substructure relation $\trianglelefteq$ that $A \trianglelefteq B$ and $|A| = |B|$ implies $A = B$. Suppose now that $F : K_1 \to K_2$ is a monotonic functor continuous on token sets. If $\langle A_k \rangle_{k \geq 0}$ is an increasing chain of objects in K_1, then by monotonicity of F we have $\bigcup_k F(A_k) \trianglelefteq F(\bigcup_k A_k)$. Since these two objects have, by the continuity of F on token sets, the same carrier sets, it follows by the above note that these are in fact the same objects, i.e. F is object-continuous.$\square$

Combining the last two results with Proposition 3.8, we get:

Theorem 3.14 *A monotonic endofunctor which is continuous on the token sets of a complete information category has an initial algebra in the subcategory of strict morphisms. The initial algebra is obtained by taking the union of the chain of iterates of the functor on the distinguished object* Δ.

4 Stone Spaces

In this section we construct an information category, denoted by **Bool-ISys**, for presenting the category of Stone spaces and continuous functions. This is based on Stone duality, but the extra structure in the information category enables domain equations to be solved in a straightforward manner. Objects of **Bool-ISys** are Boolean algebras of the form $(A, \wedge, \vee, \neg, \bot, \top)$ where A is the carrier set with elements from a countable pool, $\wedge$, $\vee$ and $\neg$ are the Boolean operations and $\bot, \top \in A$ are respectively the least and the greatest elements of the algebra. It is easy for the reader to check that the axioms satisfied by the Boolean operations are all of universal-existential form. Also can regard a Boolean homomorphism between two objects A and B as a relation $f : A \to B$ defined by u.e. $(\forall\,\exists)$ axioms as follows:

- $\forall a\, \exists b.\, afb \qquad afb \;\&\; afb' \;\Rightarrow\; b = b'$
- $\bot f \bot \qquad \top f \top$
- $afb \;\&\; a'fb' \;\Rightarrow\; (a \wedge a')f(b \wedge b') \qquad afb \;\&\; a'fb' \;\Rightarrow\; (a \vee a')f(b \vee b')$.

Given objects A and B, the substructure relation $A \trianglelefteq B$ means that A is a Boolean subalgebra of B and the inclusion morphism is simply the Boolean inclusion. The distinguished object Δ is the trivial Boolean algebra consisting of the two elements $\bot$ and $\top$. We therefore obtain an information category (**Bool-ISys** ; $\wedge, \vee, \neg$; Inc ; $=$; Δ) which is complete, since all the axioms of the category are easily seen to be of u.e. ($\forall\exists$) type. Furthermore, the category is clearly locally finite and therefore ω-algebraic.

By Stone duality, **Bool-ISys** is dual to the category of countably based Stone spaces [Hal63]. It is instructive to solve a domain equation in **Bool-ISys**, whose solution represents a well known Stone space. For this, consider the domain equation $X \cong A \times X = G(X)$ in the category of Stone spaces and continuous functions, where A is the two element set with the discrete topology and $- \times -$ represents the product of two Stone spaces with the product topology. In the dual category **Bool-ISys**, the dual equation is $Y \cong B + Y = F(Y)$, say, where B is the four element Boolean algebra consisting of $\bot, \top, 0, 1$, say, and $- + -$ represents the coproduct of two Boolean algebras defined as follows. The coproduct $B + Y$ is the algebra generated by the union of the two sets $\{\ell\} \times B \setminus \{\bot, \top\}$ and $\{r\} \times Y \setminus \{\bot, \top\}$ subject to the relations inherited form B and Y. The initial algebra of the functor F is obtained by taking the union of the chain $\Delta \trianglelefteq F(\Delta) \trianglelefteq F^2(\Delta) \trianglelefteq F^3(\Delta) \ldots$, i.e.

$$\Delta \trianglelefteq B + \Delta \trianglelefteq B + (B + \Delta) \trianglelefteq B + (B + (B + \Delta)) \trianglelefteq \ldots.$$

These Boolean algebras are generated respectively by the following sets

$$\emptyset, \ \{\ell 0, \ell 1\}, \ \{\ell 0, \ell 1, r\ell 0, r\ell 1\}, \ \{\ell 0, \ell 1, r\ell 0, r\ell 1, r^2\ell 0, r^2\ell 1\}, \ \ldots,$$

where, for convenience, we have dropped all brackets in representing the tokens, e.g. $r^2\ell 1$ stands for $(r, (r, (\ell, 1)))$. Notice that each set of generators is a subset of the next. In general, $F^n(\Delta)$ is generated by the set

$$L_n = \{\ell 0, \ell 1, r\ell 0, r\ell 1, r^2\ell 0, r^2\ell, \ldots, r^{n-1}\ell 0, r^{n-1}\ell 1\}.$$

The initial algebra is obtained by taking the union $D = \bigcup_{n \geq 0} F^n(\Delta)$, which is generated by the set $\bigcup_n L_n$. To appreciate what these tokens in fact represent as open sets, we construct the Stone spaces corresponding to the Boolean algebras in the chain. These finite discrete spaces are obtained below by the iteration of the action of the functor G on the one point space $\{*\}$, with the two element space A conveniently taken as $\{0, 1\}$:

$$\{*\}, \ \{0*, 1*\}, \ \{00*, 01*, 10*, 11*\}, \ \{000*, 001*, 010*, 011*, 100*, 101*, 110*, 111*\}, \ \ldots$$

We can now easily identify the token $r^k\ell i$ ($i = 0, 1;\ k \geq 0$) in $F^n(\Delta)$ ($n > k$) or indeed in D as the set of all those elements having i as the

k^{th} entry. These tokens therefore represent a set of subbasic open sets for the Cantor space $\{0,1\}^\omega$ which provides the final co-algebra solution of the original domain equation in the category of Stone spaces and continuous functions.

5 Scott information systems

We will construct two different but of course equivalent complete information categories for Scott domains. The first which is given in [DP90] uses the original presentation of these domains by Scott himself in [Sco82] and later by Larsen and Winskel in [LW84], whereas the second is based on the presentation given by Gunter in [Gun87] which in fact we have already mentioned in the introduction. We will develop this second presentation more elaborately to obtain an ω-algebraic information category.

5.1 ISys

In [LW84, Sco82], a (Scott) information system is defined as a structure $A = (|A|, \vdash, \mathrm{Con})$, where

- $|A|$ is a set of tokens from a fixed pool,
- Con is a non-null subset of $\mathcal{P}_f(|A|)$ (the consistent sets),
- $\vdash$ is a subset of $\mathrm{Con} \times A$ (the entailment relation).

Since $\vdash$ extends easily to a subset of $\mathrm{Con} \times \mathrm{Con}$, it is convenient to assume from the start, as we do here, that $\vdash$ is a binary relation on $\mathcal{P}_f(|A|)$. We therefore require the following axioms which are equivalent to those in the above papers:

$$\begin{array}{rl}
\text{(i)} & X \subseteq Y \ \&\ Y \in \mathrm{Con} \Rightarrow X \in \mathrm{Con} \\
\text{(ii)} & a \in |A| \Rightarrow \{a\} \in \mathrm{Con} \\
\text{(iii)} & X \vdash Y \Rightarrow X \in \mathrm{Con}_A \ \&\ Y \in \mathrm{Con}_B \\
\text{(iv)} & X \vdash Y \Rightarrow X \cup Y \in \mathrm{Con}_B \\
\text{(v)} & X \in \mathrm{Con} \ \&\ Y \subseteq X \Rightarrow X \vdash Y \\
\text{(vi)} & X \vdash Y \ \&\ Y \vdash Z \Rightarrow X \vdash Z.
\end{array}$$

Note that by (v) we have $X \vdash X$ and this together with (vi) implies that $\vdash$ is a pre-order on $\mathcal{P}_f(|A|)$. A morphism, called an approximable mapping, $r : A \to B$ between two objects is a relation $r \subseteq \mathcal{P}_f(|A|) \times \mathcal{P}_f(|B|)$ satisfying:

$$\begin{array}{rl}
\text{(i)} & XrY \Rightarrow X \in \mathrm{Con}_A \;\&\; Y \in \mathrm{Con}_B \\
\text{(ii)} & \emptyset r \emptyset \\
\text{(iii)} & XrY \;\&\; XrY' \Rightarrow Xr(Y \cup Y') \\
\text{(iv)} & X' \vdash_A X \;\&\; XrY \;\&\; Y \vdash_B Y' \Rightarrow X'rY'
\end{array}$$

It is trivial to check that $\vdash_A$ is the identity morphism on A and that whenever A is a substructure of B, $\vdash_B$ restricted to $\mathcal{P}_f(|A|) \times \mathcal{P}_f(|B|)$ is a morphism, i.e. the inclusion morphism $\mathrm{in}(A, B)$. Finally we put $\Delta = (\emptyset, \{(\emptyset, \emptyset)\}, \{\emptyset\})$ which is trivially an information system and for any approximable mapping $r : \Delta \to A$, we have

$$\mathrm{in}(\Delta, A) \;=\; \emptyset \subseteq r.$$

We now have an information category $(\mathbf{ISys}, \vdash, \mathrm{Con}, \mathrm{Inc}, \subseteq, \Delta)$, which is complete since all the axioms are of u.e. $(\forall\exists)$ type.

5.2 BC-ISys

We have already defined the category of bounded complete information systems in Section 1 (see also [Gun87]). Assuming that each system has the same distinguished token Δ, we denote this category by **BC-ISys**. Then it can easily be shown that $(\mathbf{BC\text{-}ISys}, \vdash, \mathrm{Inc}, \subseteq, \Delta)$ is a complete information category where the partial order $(A, \vdash_A, \wedge_A) \trianglelefteq (B, \vdash_B, \wedge_B)$ is captured by the approximable mapping $R \subseteq A \times B$ with aRb iff $a \vdash_B b$, $\subseteq$ is the inclusion for relations, and Δ, ambiguously, denotes the trivial object having $\{\Delta\}$ as set of tokens. In fact conditions **C1**-**C3** are trivial to check and **C4** follows from the fact that all the axioms defining objects and morphisms of **BC-ISys** are of universal $(\forall)$ type.

To obtain an ω-algebraic information category consider the full subcategory **BC-ISys*** of **BC-ISys** consisting of all objects A with elements of $|A|$ from a fixed countable pool and with $\vdash_A$ a partial order. These two categories are equivalent since every object $(A, \vdash, \wedge)$ is isomorphic to its Lindenbaum algebra $(A_{/\equiv}, \vdash_{/\equiv}, \wedge_{/\equiv})$ which is an object of **BC-ISys***. We make a slight change of notation and write the partial order $\vdash_I$ of an object I of **BC-ISys*** as $\leq_I$ with $a \leq_I b$ iff $a \vdash_I b$. We now claim that $(\mathbf{BC\text{-}ISys^*}, \leq, \mathrm{Inc}, \subseteq, \Delta)$ is an ω-algebraic information category. The completeness axiom **C4** holds as before.

Furthermore **BC-ISys*** is locally finite: Given an object I and a finite subset S of $|I|$, the set of conjunctions of all consistent subsets of S is finite and hence gives rise to a finite submodel of I. This proves that our category is locally finite and therefore ω-algebraic. It is also well known that **BC-ISys**, and therefore **BC-ISys*** is equivalent to the category of Scott domains and continuous functions; details can be found in [ES93b].

5.3 Function space constructor

All the usual constructors on Scott domains have their counterparts as constructors on **BC-ISys***. We will present just the function space constructor here. Recall that the initial algebra theorem holds for covariant functors, but the general function space constructor

$$(-) \to (-) : \mathbf{BC\text{-}ISys*}^{op} \times \mathbf{BC\text{-}ISys*} \to \mathbf{BC\text{-}ISys*}$$

is contravariant in its first argument. However if we restrict to the subcategory of inclusion morphisms denoted by $\mathbf{BC\text{-}ISys*}^{E}$, we can obtain a covariant functor

$$(-) \to^{E} (-) : \mathbf{BC\text{-}ISys*}^{E} \times \mathbf{BC\text{-}ISys*}^{E} \to \mathbf{BC\text{-}ISys}^{E}.$$

(This is of course a standard technique of domain theory first formulated in [Sco72]. For a general formulation in the context of categories with ordered hom-sets, see [SP82].) We will now describe the action of $(-) \to^{E} (-)$ on objects and inclusion morphisms. For convenience we will drop the superscript E.

Given objects I and J of **BC-ISys***, $I \to J$ is defined as follows.

- $|I \to J|$ consists of finite sets f of pairs of elements of I and J, i.e. $f = \{(a_k, b_k) | a_k \in I, b_k \in J \setminus \{\Delta\}, k \in K, K \text{ finite }\}$ satisfying the following conditions:

  H(i) whenever $L \subseteq K$ and $\{a_l | l \in L\}$ is bounded, we have $\bigwedge_{l \in L} a_l = a_k$ for some $k \in K$;

  H(ii) whenever $(a, b), (a', b') \in f$ and $a \lneq a'$, we have $b \lneq b'$;

  H(iii) whenever $(a, b), (a, b') \in f$, we have $b = b'$.

- $f_1 \leq_{I \to J} f_2$ iff for all $(a, b) \in f_2$ there exists $(a', b') \in f_1$ with $a \leq a'$ and $b' \leq b$.

- Given bounded $f_1, f_2 \in |I \to J|$, i.e. $f \le f_1$ and $f \le f_2$ for some $f \in |I \to J|$, $f_1 \wedge f_2 = \mathrm{cl}(f_1 \cup f_2)$, where $\mathrm{cl}(f_1 \cup f_2)$ is the closure of $f_1 \cup f_2$ under the H conditions above. This closure is obtained from $f_1 \cup f_2$ by the following two steps:

  cl(i) We start with $f_1 \cup f_2$ and for any of its subsets $\{(a_l, b_l) | l \in L\}$ with $\{a_l | l \in L\}$ bounded below, we add the pair $(\bigwedge_{l\in L} a_l, \bigwedge_{l\in L} b_l)$ to it. (Note that $\bigwedge_{l\in L} b_l$ exists since f satisfies H(i).)

  cl(ii) We now remove redundancies by imposing H(ii) and H(iii).

- $\Delta = \emptyset$

It is routine to check that $I \to J$ is an object of **BC-ISys***. As we will see later, conditions H(i)-(iii) ensure that a compact element of the function space of two Scott domains has a unique representation in the function space of the corresponding information systems and hence the latter space is in fact a partial order and not simply a pre-order.

The functor $(-) \to (-)$ acts on inclusion morphisms as follows. If $I \trianglelefteq I'$ and $J \trianglelefteq J'$, it is readily seen that $(I \to J) \trianglelefteq (I' \to J')$. We therefore define

$$(\mathrm{in}(I, I') \to \mathrm{in}(J, J')) \;=\; \mathrm{in}((I \to J), (I' \to J')),$$

which makes the functor $(-) \to (-)$ covariant in both arguments and standard as desired.

We must verify that our construction does capture the intended meaning of function space of Scott information systems. To make our notations clear, we denote the function space constructor in the category of domains by $(-) \to_{\mathrm{d}} (-)$.

Proposition 5.1 *Given Scott domains A and B, the information system $F(A \to_d B)$ is isomorphic to $F(A) \to F(B)$ in* **BC-ISys***.

Proof It is sufficient to prove that they are isomorphic as posets. For this we need to identify the compact elements of $A \to B$ in terms of step functions $a \searrow b$, which for elements $a \in A$ and $b \in B$ are defined by:

$$\begin{array}{rrcl} a \searrow b: & A & \longrightarrow & B \\ & x & \longmapsto & \left\{ \begin{array}{ll} b & \text{if } a \sqsubseteq x \\ \bot & \text{otherwise.} \end{array} \right. \end{array}$$

Step functions defined by a pair of compact elements are themselves compact and any compact element of $A \to B$ can then be expressed in terms of such step functions as follows. Consider a finite collection of pairs $f = \{(a_k, b_k) | a_k \in \mathcal{K}_A, b_k \in \mathcal{K}_B, k \in K, K \text{is finite}\}$ satisfying the following conditions which are the same as the H conditions with the ordering reversed:

H$'$(i) whenever $L \subseteq K$ and $\{a_l | l \in L\}$ is bounded, we have $\bigsqcup_{l \in L} a_l = a_k$ for some $k \in K$;

H$'$(ii) whenever $(a, b), (a', b') \in f$ and $a' \sqsubseteq a$, we have $b' \sqsubseteq b$;

H$'$(iii) whenever $(a, b), (a, b') \in f$, we have $b = b'$.

This defines a compact element $\hat{f}$ of the function space by $\hat{f} = \bigsqcup_{k \in K}(a_k \searrow b_k)$. In fact any compact element of the function space can be uniquely expressed as $\hat{f}$ for some f satisfying the above conditions. H$'$(i) is a completeness condition; whereas H$'$(ii)-(iii) eliminate all redundant step functions in the definition of $\hat{f}$. The mapping $h : (F(A) \to F(B)) \to F(A \to_{\mathrm{d}} B)$ given by $f \longmapsto \hat{f}$ now establishes the required isomorphism. $\Box$

It remains to verify that the function space constructor is morphism-continuous.

Proposition 5.2 *The functor*

$$(-) \to^E (-) : \mathbf{BC\text{-}ISys*}^E \times \mathbf{BC\text{-}ISys*}^E \to \mathbf{BC\text{-}ISys}^E$$

is morphism-continuous.

Proof First we show that it is continuous on objects. We simply check that it is continuous in each argument separately. Let $\langle I_i \rangle_{i \geq 0}$ be an increasing chain of objects and J a fixed object of **BC-ISys***. Since $(-) \to J$ is standard, i.e. monotonic, we already know that $\bigcup_i (I_i \to J) \subseteq (\bigcup I_i) \to J$. To show the converse let $f \in ((\bigcup I_i) \to J)$. Since f is a finite set of pairs of elements of $\bigcup I_i$ and J satisfying conditions (i) and (ii) and since $\leq$ and $\wedge$ are preserved in a chain, it follows that f belongs to $I_i \to J$ for some $i \geq 0$ and hence to $\bigcup_i (I_i \to J)$. Therefore $(-) \to (-)$ is continuous in its first argument on objects. Similarly it is continuous in the second argument on objects. Now we assume that $\langle e_i : A_i \rightarrowtail B_i \rangle_{i \geq 0}$ and $\langle e'_i : A'_i \rightarrowtail B'_i \rangle_{i \geq 0}$ are two increasing chains of inclusion morphisms and calculate

$$\begin{array}{rcl}\bigcup(e_i \to e_i') & = & \bigcup \mathrm{in}(A_i \to A_i' \,,\, B_i \to B_i') \\ & = & \mathrm{in}(\bigcup(A_i \to A_i') \,,\, \bigcup(B_i \to B_i')) \\ & = & \mathrm{in}(\bigcup A_i \to \bigcup A_i' \,,\, \bigcup B_i \to \bigcup B_i') \\ & = & \bigcup e_i \to \bigcup e_i'.\end{array}$$

This proves the continuity of $(-) \to^E (-)$.□

Other constructors, including various power domain functors, can be similarly treated.

6 SFP information systems

The category of SFP domains with continuous mappings, denoted by **SFP**, is the largest cartesian closed full subcategory of the category of ω-algebraic cpo's with continuous mappings, and is closed under the three main power functors. See [Plo81, Smy83, Jun88]. Recall that an ω-algebraic cpo, A, is $^2\!/_3$ SFP if for every finite $S \subseteq \mathcal{K}_A$, the set of minimal upperbounds of S, denoted by $\mathrm{Mub}(S)$, is finite and complete in the sense that for every upperbound x of S, there exists some $y \in \mathrm{Mub}(S)$ with $y \sqsubseteq x$. Furthermore, A is SFP if, in addition, for every finite $S \subseteq \mathcal{K}_A$, $\mathcal{U}^*(S)$ is finite where

$$\begin{array}{rcl}\mathcal{U}^*(S) & = & \bigcup_{k\in\omega} \mathcal{U}^k(S) \\ \mathcal{U}^0(S) & = & S \\ \mathcal{U}^{k+1}(S) & = & \bigcup\{\mathrm{Mub}(T) | T \subseteq \mathcal{U}^k(S)\}\end{array}$$

SFP is the category of SFP domains with continuous mappings; it is the largest cartesian closed full subcategory of ω-**ALG**, the category of ω-algebraic cpo's with continuous mappings, and is closed under the three main power functors.

We will use the above definition to construct a category of information systems **SFP-ISys** equivalent to **SFP**. The objects of this category, called SFP information systems, are given by tuples $(A, \vdash, \mathrm{Mlb}, V, \Delta)$ where $(A, \vdash)$ is, as before, a pre-order, Δ a distinguished token implied by all tokens in all objects and Mlb and V are unary operations on

$\mathcal{P}_f(A)$, satisfying the following axioms:

$$\begin{array}{rl}
\mathrm{S(i)} & x \in \mathrm{Mlb}(T) \Rightarrow x \vdash T. \\
\mathrm{S(ii)} & x \in \mathrm{Mlb}(T) \;\&\; x \vdash y \;\&\; y \vdash T \Rightarrow x \equiv y. \\
\mathrm{S(iii)} & x \vdash T \Rightarrow \exists y.\; y \in \mathrm{Mlb}(T) \;\&\; x \vdash y. \\
\mathrm{S(iv)} & T \subseteq V(T). \\
\mathrm{S(v)} & T' \subseteq V(T) \Rightarrow \mathrm{Mlb}(T') \subseteq V(T). \\
\mathrm{S(vi)} & T \subseteq T' \Rightarrow V(T) \subseteq V(T'). \\
\mathrm{S(vii)} & V(V(T)) \subseteq V(T).
\end{array}$$

Here $x \vdash T$ means $x \vdash t$ for all $t \in T$. The first three axioms say that for every finite $S \subseteq A$, $\mathrm{Mlb}(S)$ is a complete set of maximal lower bounds of S with respect to $\vdash$. Axioms S(iv)-(v) say that every finite subset T of A is contained in a finite subset $V(T)$ which is closed under Mlb. Axioms S(vi)-(vii) ensure local finiteness and are only needed, as we will see later, for constructing an ω-algebraic information category and not for a complete information category. A morphism, f, between SFP information systems A and B is an approximable mapping, i.e. a relation $f \subseteq A \times B$ satisfying

- $\Delta f \Delta$.
- $a \vdash a' f b' \vdash b \Rightarrow afb$.
- $afb_1 \;\&\; afb_2 \Rightarrow \exists b.\, b \in \mathrm{Mlb}\{b_1, b_2\} \;\&\; afb$.

Given objects A and B we define two approximable mappings:

$$\begin{array}{lll}
\mathrm{in}(A,B) : A \to B & \text{with} & a(\mathrm{in}(A,B))b \iff a \vdash_B b \\
\mathrm{pr}(B,A) : B \to A & \text{with} & b(\mathrm{pr}(B,A))a \iff b \vdash_B a.
\end{array}$$

It can be easily checked that these are in fact approximable mappings satisfying $\mathrm{in}(A,B);\mathrm{pr}(B,A) = \mathrm{Id}(A)$ and $\mathrm{pr}(B,A);\mathrm{in}(A,B) \subseteq \mathrm{Id}(B)$.

We check that $(\text{SFP-ISys}, \vdash, \mathrm{Mlb}, V, \mathrm{Inc}, \subseteq, \Delta)$ is a complete information category: In fact conditions **C1**-**C3** are straightforward to verify; and all our axioms are of u.e. ($\forall\exists$) type which ensures completeness. In order to obtain an ω-algebraic information category, we do exactly as we did with **BC-ISys**, and consider the subcategory **SFP-ISys*** of partially ordered objects with elements from a countable pool. **SFP-ISys*** is locally finite: Given an object $(A, \leq, \mathrm{Mlb}, V)$ of this category and a finite subset $T \subseteq A$, we can easily see by the S axioms that $(V(T), \leq, \mathrm{Mlb}, V)$ (in which the partial order and the two operations are restricted to $V(T)$) is a finite submodel of A. Therefore, **SFP-ISys*** is an ω-algebraic information category.

The equivalence of the category of **SFP** and **SFP-ISys** is given by the functors

$$\mathbf{SFP} \underset{F}{\overset{G}{\rightleftarrows}} \mathbf{SFP\text{-}ISys}$$

as follows. Given an SFP domain $(D, \sqsubseteq)$, we put $F(D) = (\mathcal{K}_D, \vdash, \mathrm{Mlb}, \mathcal{U}^*)$ where $\vdash$ is the inverse of $\sqsubseteq$, and Mlb=Mub (the maximal lower bounds wrt $\vdash$ are the same as the minimal upper bounds wrt $\sqsubseteq$) and $\mathcal{U}^*$ is as defined in the beginning of this section. It is easily seen that this is an object of **SFP-ISys** since $\mathcal{U}^*$ satisfies axioms (iv)-(vii) for V. Given a continuous map $f : D \to E$ between SFP domains, $F(f) = R_f$ with $aR_f b$ iff $f(a) \sqsupseteq b$. On the other hand, given an **SFP-ISys** object $(A, \vdash, \mathrm{Mlb}, V)$, we put $G(A) = (\mathrm{Fil}(A), \subseteq)$, the set of filters of A ordered by inclusion. To see that this is in fact SFP, note that the basis of this domain is isomorphic to $(A, \sqsubseteq)$ where $\sqsubseteq$ is the inverse of $\vdash$, and therefore Mlb now gives the minimal upperbounds w.r.t. $\sqsubseteq$, and S(iv)-(v) imply that any finite subset of A is contained in a finite set closed under the minimal upperbound operation. (However note that for $T \subseteq_f A$ we only have $\mathcal{U}^*(T) \subseteq V(T)$; in order to obtain $\mathcal{U}^* = V$ one has to add further S axioms which we prefer to avoid.) Given an approximable mapping $R : A \to B$ between two objects in **SFP-ISys** objects, we define $G(R) = f_R$ with $f_R : \mathrm{Fil}(A) \to \mathrm{Fil}(B)$ where $f_R(x) = \{b | \exists a \in x.\ aRb\}$. It can be checked that this defines an equivalence of the two categories and that for SFP information systems A and B with $A \trianglelefteq B$, the approximable mappings $\mathrm{in}(A, B)$ and $\mathrm{pr}(B, A)$ are mapped by G to an embedding projection pair between the SFP domains $(\mathrm{Fil}(A), \subseteq)$ and $(\mathrm{Fil}(B), \subseteq)$.

We will now explicitly define the function space constructor in **SFP-ISys***. Assume that A and B are objects of this category and define $A \to B$ by:

- $|A \to B|$ consists of finite sets of pairs of elements of A and B of the form $f = \{(a_k, b_k) | a_k \in A, b_k \in B \setminus \{\Delta\}, k \in K, K \quad \text{finite}\}$ satisfying:

  T(i) $\{a_k | k \in K\}$ is closed under Mub on any of its subsets.

T(ii) $\forall k, k' \in K. a_k \lneq a_{k'} \Rightarrow b_k \lneq b_{k'}$.

T(iii) $\forall k, k' \in K. a_k = a_{k'} \Rightarrow b_k = b_{k'}$.

- $f_1 \leq_{A \to B} f_2$ iff for all $(a, b) \in f_2$ there exists $(a', b') \in f_1$ with $a \leq a'$ and $b' \leq b$.

Compare the T conditions above with the H conditions in Section 5.3. We now show that $A \to B$ has the required operations Mlb and V. Assume that SFP objects D and E are such that $F(D) \cong A$ and $F(E) \cong B$. Since the function space $D \to_{\mathrm{d}} E$ is SFP, $F(D \to_{\mathrm{d}} E)$ is an SFP information system with the required operations Mlb and V. Moreover, we can easily see that $F(D \to_{\mathrm{d}} E)$ and $A \to B$ are isomorphic as posets. We therefore conclude that $A \to B$ has the required operations Mlb and V as well.

The function space constructor, as in the case of **BC-ISys***, gives rise to a standard and morphism-continuous covariant functor on the subcategory of inclusion morphisms of **SFP-ISys***.

7 Information systems for dI-domains

We now consider dI-domains which were introduced in [Ber78] in the context of studying the full abstraction problem for typed λ-calculi. Recall that a dI-domain is a bounded complete (Scott) domain D which satisfies the following two conditions:

dI(i) for all $x, y, z \in D$, if $y \uparrow z$ then $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$;

dI(ii) there are only a finite number of elements below each compact element.

A function $f : D \to E$ between dI-domains is *stable* if it is continuous and satisfies:

$$x \uparrow y \Rightarrow f(x \sqcap y) = f(x) \sqcap f(y).$$

Given stable functions $f, g : D \to E$, we say f is less than g in the *stable ordering*, denoted by $f \sqsubseteq_s g$, if

$$\begin{array}{ll} \forall x \in D. & f(x) \sqsubseteq g(x) \\ \forall x, y \in D. & x \sqsubseteq y \Rightarrow f(x) = f(y) \sqcap g(x). \end{array}$$

A stable function $f : D \to E$ is a *rigid embedding* if there exists a stable function $g : E \to D$, called a projection with $f; g = \mathrm{Id}(E)$ and $g; f \sqsubseteq_s \mathrm{Id}(D)$.

The category of dI-domains with stable functions, denoted by **DI**, is cartesian closed, products are formed as cartesian product ordered pointwise and the function space of dI-domains D and E consists of the stable functions $f : D \to E$ with the stable ordering. The category is closed under direct limits with rigid embeddings, and a great deal of denotational semantics can be done in this category including the solving of recursive domain equations involving product, sum, and function space. For more details see [CGW87].

Here, we will present a category of information systems for dI-domains, which we denote by **DI-ISys**. As in the previous examples, we will represent a dI-domain by its compact elements. We thereby stress the fact that all domains of computations can be represented bt their compact elements. A different treatment of dI-domains is provided by Zhang [Zha89] who uses the *complete prime* elements of the domain in its representation.

The objects of **DI-ISys**, called dI-information systems, are tuples $(A, \vdash, \wedge, \vee, h, \Delta)$, where $(A, \vdash, \wedge, \Delta)$ is a Scott information system, $\vee$ is a (total) binary disjunction and $h : A \to \mathcal{P}_f(A)$ is an operation which satisfy:

D(i) $\Delta \vee \Delta = \Delta$.

D(ii) $x \vdash x \vee y$.

D(iii) $y \vdash x \vee y$.

D(iv) $x \vdash z \;\&\; y \vdash z \;\Rightarrow\; x \vee y \vdash z$.

D(v) $x \vdash y \;\Rightarrow\; \exists y'.\, y' \in h(x) \;\&\; y' \equiv y$.

D(vi) $y \in h(x) \;\Rightarrow\; x \vdash y$.

D(vii) $u \vdash x \;\&\; u \vdash y \;\Rightarrow\; (z \vee x) \wedge (z \vee y) \;\vdash\; z \vee (x \wedge y)$.

Note that axioms for h imply that the set of equivalent classes of tokens implied by a token is finite, and for each a, $h(a)$ consists of a finite set of tokens implied by a, which is complete in the sense that each token implied by a has an equivalent in $h(a)$. Note also that D(vii) implies that $(z \vee x) \wedge (z \vee y) \equiv z \vee (x \wedge y)$, when $x \wedge y$ exists, since the inverse implication always holds.

Morphisms of **DI-ISys** are called *traces* and are defined as follows. A trace $R : A \to B$ of dI-information systems A and B is a relation

$R \subseteq A \times B$ satisfying:

tr(i) $\Delta R \Delta$.

tr(ii) $aRb \ \&\ a'Rb \ \&\ c \vdash a \ \&\ c \vdash a' \ \Rightarrow\ a \vdash a'$.

tr(iii) $aRb \ \&\ b \vdash b' \ \Rightarrow\ \exists a'.\ a' \in h(a) \ \&\ a'Rb'$.

tr(iv) $aRb \ \&\ a'Rb' \ \&\ c \vdash a \ \&\ c \vdash a' \ \Rightarrow\ \exists d.\ d \vdash b \ \&\ d \vdash b' \ \&\ (a \wedge a')R(b \wedge b')$.

tr(v) $a \equiv a' \ \&\ b \equiv b' \ \&\ aRb \ \Rightarrow\ a'Rb'$.

The identity morphism $\mathrm{Id}(A)$ of A is the diagonal trace given by

$$(a, a') \in \mathrm{Id}(A) \iff a \equiv a'.$$

More generally, given objects A and B with $A \trianglelefteq B$, we define the inclusion $\mathrm{in}(A, B)$ and the projection $\mathrm{pr}(B, A)$ by

$$(a, b) \in \mathrm{in}(A, B) \iff a \equiv b \qquad \text{and} \qquad (b, a) \in \mathrm{pr}(B, A) \iff b \equiv a.$$

It is trivial to check that these in fact define traces with $\mathrm{in}(A, B); \mathrm{pr}(B, A) = \mathrm{Id}(A)$ and $\mathrm{pr}(B, A); \mathrm{in}(A, B) \subseteq \mathrm{Id}(B)$.

We leave the proof of the following simple proposition to the reader.

Proposition 7.1 *Any trace $R : A \to B$ determines an approximable mapping $\check{R} : A \to B$ defined by $a\check{R}b$ iff $\exists a'.\ a \vdash a' \ \&\ a'Rb$.*

We now see that (**DI-ISys** ; $\equiv$; $\vdash$; $\wedge, \vee, h$; Inc ; $\subseteq$; Δ) is a complete information category. (Notice that the order in each object inducing the inclusion morphisms is given here by $\equiv$ and not by $\vdash$ as has been the case in the previous examples. See the definition of an information category in 3.) Conditions **C1**-**C3** are easy to check and the completeness condition **C4** follows from the fact that all the axioms of **DI-ISys** are of u.e. ($\forall\exists$) type.

To obtain an ω-algebraic information category, we do as in the previous cases and consider the full subcategory **DI-ISys*** of partially ordered objects with elements from a countable pool. We replace $\vdash$ by $\leq$ and write the objects of this subcategory as $(A, \leq, \wedge, \vee, h)$. Since we have a partial order now, D(v) and D(vi) imply that $h(a) = \uparrow a = \{x | a \leq x\}$. **DI-ISys*** is clearly equivalent to **DI-ISys** and is a complete information category. It is also locally finite: Given an object $(A, \leq, \wedge, \vee, h)$ of **DI-ISys*** and a finite subset $T \subseteq A$, consider the closure, S, of T under $\wedge$. Since $(A, \leq, \wedge)$ is a Scott information systems S is finite. Therefore, $\uparrow S = \bigcup\{\uparrow x | x \in S\}$ is also finite. We can now see that the object given by the set $\uparrow S$ with the partial order and the operations of A restricted to it is a finite submodel of A. This shows that **DI-ISys*** is ω-algebraic, as required.

We will now define functors

$$\mathbf{DI} \underset{F}{\overset{G}{\rightleftarrows}} \textbf{DI-ISys*}$$

which make an equivalence of two categories. We start by defining F. For an object $(D, \sqsubseteq)$ of **DI**, we define $F_o(A) = (\mathcal{K}_D, \leq, \wedge, \vee, h, \Delta)$ where

- $\leq$ is the inverse of $\sqsubseteq$;
- $a_1 \wedge a_2 = a_1 \sqcup a_2$ if this exists;
- $a_1 \vee a_2 = a_1 \sqcap a_2$;[1]
- $h(a) = \downarrow a = \{x | x \sqsubseteq a\}$;
- $\Delta = \perp$

It is readily seen on the basis of dI(i) and dI(ii) that $F_o(A)$ satisfies D(i)-D(vii). Next consider a stable mapping $f : D \to E$ of dI-domains and define $F_m(f) = R_f$ where $R_f \subseteq \mathcal{K}_D \times \mathcal{K}_E$ is given by $aR_f b$ iff a is a minimal element of $f^{-1}(\uparrow b)$. Since minimal elements of Scott open sets are compact and $f^{-1}(\uparrow b)$ is open (f is continuous and $\uparrow b$ is open as b is compact), this definition makes sense.

Proposition 7.2 *R_f is a trace.*

Proof We show that R_f satisfies tr(i)-tr(v), with $\vdash$ and $\equiv$ replaced by $\leq$ and $=$ respectively. Recall that $\leq$ in $F_o(D)$ is the inverse of $\sqsubseteq$ in D and for convenience we shall refer to $\sqsubseteq$ in the following proof. Since $\perp$ is the least element of $f^{-1}(\perp) = D$, we see that tr(i) holds. To prove tr(ii), assume a and a' are bounded above (wrt $\sqsubseteq$) and are minimal elements of $f^{-1}(\uparrow b)$. Since f is stable we have $f(a \sqcap a') = f(a) \sqcap f(a') \sqsupseteq b$ and therefore $a = a'$ by the minimality of a. Next to show tr(iii), assume a is a minimal element of $f^{-1}(\uparrow b)$ and $b \sqsubseteq b'$. Then there is a minimal element a' of $f^{-1}(\uparrow b')$ below $a \in f^{-1}(b')$. To show tr(iv), assume a and a' are bounded above and are minimal elements of $f^{-1}(\uparrow b)$ and $f^{-1}(\uparrow b')$ respectively. Then $f(a) \sqsupseteq b$

[1]Note that in a dI-domain $a_1 \sqcap a_2$ is compact if a_1 and a_2 are compact.

and $f(a') \sqsupseteq b'$, and therefore b and b' are bounded above by $f(a \sqcup a')$. Furthermore $a \sqcup a'$ is a minimal element of $f^{-1}(\uparrow(b \sqcup b'))$: For suppose $x \sqsubseteq a \sqcup a'$ and $b \sqcup b' \sqsubseteq f(x)$, then by the stability of f we have

$$f(x \sqcap a) \;=\; f(x) \sqcap f(a) \;\sqsupseteq\; (b \sqcup b') \sqcap b \;=\; b$$

and by minimality of a we get $x \sqcap a = a$, i.e. $x \sqsupseteq a$. Similarly $x \sqsupseteq a'$ so that $x \sqsupseteq a \sqcup a'$ and therefore $x = a \sqcup a'$, which establishes tr(iv). Finally note that because we have a partial order tr(v) is now a tautology. □

We will now define the functor $G : \textbf{DI-ISys*} \to \textbf{DI}$. Given an object $(A, \leq, \wedge, \vee, \uparrow(-))$ of **DI-ISys***, $G_o(A) = (\mathrm{Fil}(A), \subseteq)$ is the set of filters of A ordered by inclusion. This is a Scott domain since $(A, \leq, \wedge)$ is a Scott information system. The compact elements are the principal filters $\uparrow a = \{a' \mid a \leq a'\}$ for any $a \in A$. For $x, y \in (\mathrm{Fil}(A), \subseteq)$, we have $x \sqcap y = x \cap y$ and, if x and y are bounded above, $x \sqcup y = \{a \wedge b \mid a \in x, b \in y\}$. We show that this domain satisfies the distributivity axiom dI(i) as follows. The basis of this domain is isomorphic with the poset $(A, \sqsubseteq)$ where $\sqsubseteq$ is the inverse of $\leq$. By D(vii), this poset satisfies dI(i) and therefore so does the basis of $G_o(A)$. Since the operations $\sqcup$ and $\sqcap$ are continuous operations in any cpo, it follows that $G_o(A)$ itself satisfies dI(i). Furthermore, since $\uparrow a$ is finite in $(A, \leq)$, it follows that $\downarrow a$ is finite in $(A, \sqsubseteq)$, i.e. dI(ii) holds as well. Hence, $G_o(A)$ is an object of **DI** as required.

Next, consider a trace $R : A \to B$ and put $G_m(R) = f_R$ where $f_R : (\mathrm{Fil}(A), \subseteq) \to (\mathrm{Fil}(B), \subseteq)$ is the continuous map corresponding to the approximable mapping $\check{R} : A \to B$ (see Proposition 7.1). We will prove that f_R is a stable map using the following lemma.

Lemma 7.3 *Let $g : D \to E$ be a continuous map between dI-domains D and E and let $\check{g} : (\mathcal{K}_D, \leq, \wedge, \vee) \to (\mathcal{K}_E, \leq, \wedge, \vee)$ be the approximable mapping corresponding to g. Then g is stable if and only if $\check{g}$ satisfies the following condition:*

$$a_1 {\downarrow} a_2 \;\&\; a_1 \check{g} b \;\&\; a_2 \check{g} b \;\Rightarrow\; (a_1 \vee a_2) \check{g} b.$$

Proof Since g is continuous and $\sqcap$ is a continuous operation we only need to consider the stability condition for compact elements. Therefore,

g is stable iff
$a_1, a_2 \in \mathcal{K}_D \;\&\; a_1 \uparrow a_2 \;\Rightarrow\; g(a_1) \sqcap g(a_2) \;\sqsubseteq g(a_1 \sqcap a_2)$ iff
(the reverse inequality always holds by monotonicity of g)
$a_1, a_2 \in \mathcal{K}_D \;\&\; a_1 \uparrow a_2 \;\&\; b \in \mathcal{K}_E \;\Rightarrow (b \;\sqsubseteq g(a_1) \sqcap g(a_2) \;\Rightarrow\; b \;\sqsubseteq g(a_1 \sqcap a_2))$ iff
$a_1, a_2 \in \mathcal{K}_D \;\&\; a_1 \uparrow a_2 \;\&\; b \in \mathcal{K}_E \;\Rightarrow (b \;\sqsubseteq g(a_1) \;\&\; b \;\sqsubseteq g(a_2) \;\Rightarrow\; b \;\sqsubseteq g(a_1 \sqcap a_2))$ iff
$a_1 \downarrow a_2 \;\&\; a_1 \check{g} b \;\&\; a_2 \check{g} b \;\Rightarrow\; (a_1 \vee a_2) \check{g} b$
(since the ordering in the information system is reversed and $a_1 R b \iff b \sqsubseteq g(a)$).

This completes the proof. □

Proposition 7.4 *f_R is stable.*

Proof We only need to check that the condition in the lemma holds for the approximable map $\check{R} : A \to B$. Suppose a_1 and a_2 are bounded below in A and $a_1 \check{R} b$ and $a_2 \check{R} b$. Then there exists $a'_1 \geq a_1$ and $a'_2 \geq a_2$ with $a'_1 R b$ and $a'_2 R b$. Since a'_1 and a'_2 are bounded below, tr(ii) implies that $a'_1 = a'_2 = a$, say. Therefore, $(a_1 \vee a_2) \leq a R b$, and as $\check{R}$ is an approximable mapping we finally get $(a_1 \vee a_2) \check{R} b$. □

We have now completely defined the functors

$$\mathbf{DI} \underset{F}{\overset{G}{\rightleftarrows}} \mathbf{DI\text{-}ISys^*};$$

it is routine to verify that they indeed induce an equivalence of the two categories. We will now show that the stable ordering of morphisms is captured by the inclusion of traces.

Proposition 7.5 *$f \sqsubseteq_s g$ iff $R_f \subseteq R_g$.*

Proof Assume $f, g : D \to E$ with $f \sqsubseteq_s g$. Then we have

$$R_f, R_g : (\mathcal{K}_D, \leq, \wedge, \vee) \to (\mathcal{K}_E, \leq, \wedge, \vee).$$

Let $a R_f b$. Since $b \sqsubseteq f(a) \sqsubseteq g(a)$, there exists $a' \in \mathcal{K}_A$ with $a' \sqsubseteq a$ and $a' R_g b$. By the stable order condition however, $f(a') = f(a) \sqcap g(a') \sqsupseteq b$ and hence $a = a'$, since a is minimal. Therefore $R_f \subseteq R_g$. In the other direction assume $R_f \subseteq R_g$. By continuity, it is sufficient to prove the stable ordering conditions on the compact elements of A. Let $a \in \mathcal{K}_A$, then for all $b \in \mathcal{K}_B$ we have

$$b \sqsubseteq f(a) \Rightarrow \exists a' \sqsubseteq a.\, a' R_f b \Rightarrow a' R_g b \Rightarrow b \sqsubseteq g(a).$$

This implies $f \sqsubseteq g$. Now let $a_1, a_2 \in \mathcal{K}_A$ with $a_1 \sqsubseteq a_2$ and take any $b \in \mathcal{K}_B$ with $b \sqsubseteq f(a_2) \sqcap g(a_1)$. Then $b \sqsubseteq f(a_2)$ and $b \sqsubseteq g(a_1)$ and hence there exists $a_1' \sqsubseteq a_1$ and $a_2' \sqsubseteq a_2$ with $a_2 R_f b$ and $a_1 R_g b$. By tr(ii) we obtain $a_1' = a_2'$ and therefore $a_1' R_f b$ which implies $b \sqsubseteq f(a_1)$. This gives us $f(a_1) \sqsupseteq f(a_2) \sqcap g(a_1)$. Since the reverse inequality holds by monotonicity of f and $f \sqsubseteq g$, we obtain $f \sqsubseteq_s g$. $\square$

Corollary 7.6 *A pair of inclusion and projection traces in* **DI-ISys*** *represent a rigid embedding projection pair in* **DI**.

7.1 Function space constructor

In this subsection, we will define the function space constructor in **DI-ISys***. Given objects A and B of **DI-ISys***, the function space $A \to B$ is defined as follows. $|A \to B|$ consists of finite traces $R : A \to B$, i.e. finite subsets of $A \times B$ satisfying tr(i)-tr(iv) with $\vdash$ replaced by $\leq$. (Recall that tr(v) is now redundant since $\leq$ is a partial order.) The partial order on $A \to B$ is given by

$$R_2 \leq R_1 \iff R_1 \subseteq R_2$$

When $R \leq R_1$ and $R \leq R_2$, we define

$$R_1 \wedge R_2 = \{(a_1 \wedge a_2, b_1 \wedge b_2) \mid a_1 {\downarrow} a_2, a_1 R_1 b_1, a_2 R_2 b_2\}.$$

Note that in the above definition the existence of $b_1 \wedge b_2$ is ensured by tr(iv) applied to R. As we will see in the next proposition, $R_1 \wedge R_2$ is the smallest trace containing R_1 and R_2. In order to define the operation $\vee$ in $A \to B$, it is convenient to put $R^{(a,b)} = \{(a', b') \in R \mid a \leq a', b \leq b'\}$, for each $R \in |A \to B|$ and $(a, b) \in R$. Now we define

$$R_1 \vee R_2 = \{(a, b) \in R_1 \cap R_2 \mid R_1^{(a,b)} = R_2^{(a,b)}\}.$$

We will see shortly that $R_1 \vee R_2$ is the largest trace contained in both R_1 and R_2. The condition $R_1^{(a,b)} = R_2^{(a,b)}$ is a *rigidity* condition; it ensures that we do in fact get a trace. Finally, we let $h(R) = \{R' \mid R' \subseteq R\}$ and $\Delta = \{(\Delta, \Delta)\}$.

Proposition 7.7 $(|A \to B|, \leq, \wedge, \vee, h, \Delta)$ *is a dI-information system.*

Proof The image of h is clearly finite, and h satisfies axioms D(v)-D(vi). We therefore need to show that the operations $\wedge$ and $\vee$ do indeed give us elements of $|A \to B|$ and that they satisfy the required axioms.

(i) We start with $\wedge$. Let $R \leq R_1$ and $R \leq R_2$; we will verify tr(i)-tr(iv) for $R_1 \wedge R_2$. Since $\Delta R_1 \Delta$ and $\Delta R_2 \Delta$, we have

$$(\Delta, \Delta) \;=\; (\Delta \wedge \Delta, \Delta \wedge \Delta) \in R_1 \wedge R_2,$$

i.e. tr(i) holds. To show tr(ii), let $a_1 \wedge a_2, a_1' \wedge a_2'$ be bounded below and assume we have $(a_1 \wedge a_2, b_1 \wedge b_2) \in R_1 \wedge R_2$ and $(a_1' \wedge a_2', b_1 \wedge b_2) \in R_1 \wedge R_2$. Then a_1, a_1' will be bounded below as well and moreover we have $a_1 R_1 b_1$ and $a_1' R_1 b_1$. By tr(ii) applied to R_1 we get $a_1 = a_1'$. Similarly $a_2 = a_2'$ and hence $a_1 \wedge a_2 \;=\; a_1' \wedge a_2'$ i.e. tr(ii) holds for $R_1 \wedge R_2$. To prove tr(iii), consider $(a_1 \wedge a_2, b_1 \wedge b_2)$ with $a_1 R_1 b_1$ and $a_2 R_2 b_2$, and let $b_1 \wedge b_2 \leq b$. Then

$$b \;=\; b \vee (b_1 \wedge b_2) \;=\; (b \vee b_1) \wedge (b \vee b_2),$$

with $b_i \leq b \vee b_i, i = 1, 2$. Hence, by tr(iii) applied to R_1 and R_2, there exists, for $i = 1, 2$, a_i' with $a_i \leq a_i'$ and $a_i' R_i (b \vee b_i)$. Therefore

$$(a_1' \wedge a_2', (b \vee b_1) \wedge (b \vee b_2)) \in R_1 \wedge R_2.$$

Since $a_1 \wedge a_2 \leq a_1' \wedge a_2'$ and $(b \vee b_1) \wedge (b \vee b_2) \;=\; b$, we have verified tr(iii). Finally, to show tr(iv), consider $(a_1 \wedge a_2, b_1 \wedge b_2)$ and $(a_1' \wedge a_2', b_1' \wedge b_2')$ with $a_i R_i b_i, a_i' R_i b_i'$ $(i = 1, 2)$ and $c \leq a_1 \wedge a_2, c \leq a_1' \wedge a_2'$. We get $c \leq a_i, c \leq a_i'$ $(i = 1, 2)$ and hence by tr(iv) applied to R_i we obtain $(a_i \wedge a_i') R_i (b_i \wedge b_i')$ $(i = 1, 2)$. Therefore,

$$((a_1 \wedge a_2) \wedge (a_1' \wedge a_2'), (b_1 \wedge b_2) \wedge (b_1' \wedge b_2')) \in R_1 \wedge R_2,$$

as required. We conclude that $R_1 \wedge R_2 \in |A \to B|$. Now we show that $R_1 \wedge R_2$ is in fact the conjunction of R_1 and R_2. First we check that $R_1 \wedge R_2 \leq R_i$ $(i = 1, 2)$. Let $(a, b) \in R_1$, then $(a, b) \;=\; (a \wedge \Delta, b \wedge \Delta) \in R_1 \wedge R_2$. Hence $R_1 \subseteq R_1 \wedge R_2$, i.e. $R_1 \wedge R_2 \leq R_1$. Similarly $R_1 \wedge R_2 \leq R_2$. Furthermore, $R \leq R_1 \wedge R_2$, for consider $(a_1 \wedge a_2, b_1 \wedge b_2)$ with $a_i R_i b_i$ $(i = 1, 2)$. Then $(a_i, b_i) \in R$ since $R \leq R_i$ $(i = 1, 2)$. Hence by tr(iv) applied to R we have $(a_1 \wedge a_2, b_1 \wedge b_2) \in R$ as required.

(ii) We now consider $\vee$. To show that $R_1 \vee R_2$ is a trace, the only non-trivial axiom to check is tr(iii). Let $(a,b) \in R_1 \vee R_2$ and $b \leq b'$. Then aR_1b, aR_2b and $R_1^{(a,b)} = R_2^{(a,b)}$. By tr(iii) applied to R_1, there exists a' with $a \leq a'$ and $a'R_1b'$. Now $R_1^{(a,b)} = R_2^{(a,b)}$ implies that $a'R_2b'$ and hence $(a',b') \in R_1 \vee R_2$, since $R_1^{(a',b')} = R_2^{(a',b')}$ as $a \leq a'$ and $b \leq b'$. To show that $\vee$ satisfies the axioms of disjunction, note first that since $R_1 \vee R_2 \subseteq R_i$ we have $R_i \leq R_1 \vee R_2$ $(i = 1, 2)$. Assume now that we have $R_1 \leq R, R_2 \leq R$, and suppose $(a,b) \in R$. Then $(a,b) \in R_1 \cap R_2$. To check that $R_1^{(a,b)} = R_2^{(a,b)}$, assume $(a',b') \in R_1^{(a,b)}$. Since $b \leq b'$, tr(iii) applied to R implies that there exists $a'' \in A$ with $a \leq a''$ and $a''Rb'$. Hence we have $a'R_1b'$ and $a''R_1b'$ with $a \leq a'$ and $a \leq a''$ which, by tr(ii) applied to R_1, imply that $a' = a''$, i.e. $a'Rb'$ and hence $a'R_2b'$. This shows that $R_1^{(a,b)} \subseteq R_2^{(a,b)}$ and therefore by symmetry $R_1^{(a,b)} = R_2^{(a,b)}$. We conclude that $(a,b) \in R_1 \vee R_2$, i.e. $R_1 \vee R_2 \leq R$.

(iii) It remains to check the distributivity axiom D(vii) in $A \to B$. Let R, S, T be elements of $|A \to B|$ with S, T bounded below; we must show that

$$R \vee (S \wedge T) \subseteq (R \vee S) \wedge (R \vee T).$$

Let $(a,b) \in R \vee (S \wedge T)$, then we have $(a,b) \in R \cap (S \wedge T)$ and the rigidity condition $R^{(a,b)} = (S \wedge T)^{(a,b)}$. By the definition of $S \wedge T$, there exists $(a',b') \in S$ and $(a'',b'') \in T$ with $a = a' \wedge a''$ and $b = b' \wedge b''$. Since $(a',b') = (a' \wedge \Delta, b' \wedge \Delta) \in S \wedge T$ and $a' \wedge a'' \leq a', b' \wedge b'' \leq b'$, the rigidity condition implies that $(a',b') \in R$, i.e. $(a',b') \in R \cap S$. Next we show that $R^{(a',b')} = S^{(a',b')}$. Let $(x,y) \in S^{(a',b')}$, then $a \leq a' \leq x$, $b \leq b' \leq y$ and xSy. Hence $(x,y) \in S \wedge T$, and $R^{(a,b)} = (S \wedge T)^{(a,b)}$ implies that xRy. Therefore $S^{(a',b')} \subseteq R^{(a',b')}$. To obtain the reverse inequality, suppose that $(x,y) \in R^{(a',b')}$. Then $a \leq a' \leq x, b \leq b' \leq y$ and xRy. Clearly we have $(x,y) \in R^{(a,b)}$, and therefore by $R^{(a,b)} = (S \wedge T)^{(a,b)}$, we get $(x,y) \in S \wedge T$. On the other hand, since $a'Sb'$ and $b' \leq y$, there exists x' with $a' \leq x'$ and $x'Sy$, and hence $(x',y) \in S \wedge T$. Now tr(ii) applied to $S \wedge T$ gives $x = x'$. Hence xSy and we get $R^{(a',b')} \subseteq S^{(a',b')}$. This establishes $(a',b') \in R \vee S$. Similarly $(a'',b'') \in R \vee T$. We conclude that

$$(a,b) = (a' \wedge a'', b' \wedge b'') \in (R \vee S) \wedge (R \vee T),$$

as required. $\square$

8 Information systems for continuous domains

Semantic domains are most commonly taken to be algebraic dcpo's of some kind. It is well known that a category P of algebraic domains typically can be extended to a category of *continuous* domains, the objects of which may be considered as the retracts (or alternatively as projections) of the objects of P. Thus we have the continuous lattices as the retracts (or projections) of the algebraic lattices, the bounded complete continuous cpo's as the retracts of the Scott domains, and so on.

Abstractly, the extension to the category of retracts may be considered as the construction of the *Karoubi envelope* [AGV72, Fre64, LS86] of the original category: if P is any category, the Karoubi envelope $\mathrm{Kar}(P)$ of P has as objects the pairs (A, r) where A is an object of P and $r : A \to A$ is a retraction (that is, $r; r = r$), while a morphism from (A, r) to (B, s) is a morphism $f : A \to B$ of P such that $r; f; s = f$. It should be clear that in the usual cases, where we are dealing with categories of spaces of some kind, $\mathrm{Kar}(P)$ is equivalent to the category of retracts of the objects of P. If $r : A \to A, s : B \to B$ are retractions of the spaces A, B and $h : A \to B$ any continuous map, then $r; h; s$ is in effect a map of the image $r(A)$ into $r(B)$; indeed there is an obvious one to one correspondence between the continuous maps from $r(A)$ to $s(B)$ and the continuous maps from A to B of the form $r; h; s$ (equivalently, the maps $f : A \to B$ such that $r; f; s = f$). In categories having a distinguished ordering of hom-sets, the same construction can be carried out using projections (retractions $r : A \to A$ such that $r \leq \mathrm{Id}_A$) instead of general retractions.

These constructions have been made in the context of the abstract I-categories in [ES91a, ES93b]. But they also make good sense in the setting of information categories. Since now a morphism is a relation over token sets, a Karoubi object (A, r) is formed by adding a relation, denoted by $<$, satisfying appropriate axioms, to the structure A. Morphisms of the extended category have to satisfy some further axioms, and the inclusion morphisms have to take account of the additional structure.

8.1 Continuous bounded complete posets

We illustrate this by presenting information systems for continuous bounded complete posets [Smy77], i.e. the projections (equivalently retracts) of the Scott domains. By the equivalence of the category of Scott domains and that of the Scott information systems, a category of information systems for continuous bounded complete posets is obtained simply by taking the Karoubi envelope of **BC-ISsy**.

An object of this envelope takes the form $(A, \vdash, \wedge, \Delta, <)$, where $I = (A, \vdash, \wedge, \Delta)$ is a Scott information system i.e. an object of **BC-ISys**, and $<$ is a projection (or retraction) of I. To axiomatise this, we must add to the axioms which state that I is Scott information system the following:

$$
\begin{array}{rl}
\text{cs(i)} & \Delta < \Delta. \\
\text{cs(ii)} & a \vdash b, b < c, c \vdash d \Rightarrow a < d. \\
\text{cs(iii)} & a < b, a < c \Rightarrow a < b \wedge c. \\
\text{cs(iv)} & a < b, b < c \Rightarrow a < c. \\
\text{cs(v)} & a < c \Rightarrow \exists b.\ a < b < c. \\
\text{cs(vi)} & a < b \Rightarrow a \vdash b.
\end{array}
$$

Here, cs(i)-cs(iii) say that $<$ is a Scott morphism, cs(iv)-cs(v) that this morphism is a retraction ($<;< = <$), and the (optional) cs(vi) that this retraction is a projection.

Next we consider how to describe morphisms. A morphism from $(I, <)$ to $(I', <')$, where $I = (A, \vdash, \wedge, \Delta), I' = (A', \vdash', \wedge', \Delta)$ is a Scott morphism $f : I \to I'$ satisfying $<; f; <' = f$. Thus, to the axioms for a Scott morphism we add:

$$afa' \Rightarrow \exists b, b'.\ a < b \ \&\ bfb' \ \&\ b' < a'.$$

Note that, in the presence of cs(vi), the converse of this axiom holds automatically. If cs(vi) is dispensed with, the converse needs to be added as an axiom. Suppose in particular that $(I, <)$ is a substructure of $(I', <')$, so that $<$ is the restriction of $<'$ to $A \times A'$. Put $e = <; <'$. It is easy to check that $e = <; i; <'$, where

$$i \ = \ \mathrm{in}(A, A') \ = \vdash_{A'} \cap (A \times A').$$

We also have

$$<; e; <' \ = \ <; <; <'; <' \ = \ <; <' \ = \ e,$$

so that e is a morphism; it is of course the *inclusion* morphism of $(I, <)$ in $(I', <')$.

We thus obtain an information category **CS-ISsy** having objects $(I,<)$ (with $<$ as the distinguished transitive order) and morphisms as described above. Since the axioms, both for objects and morphisms are of u.e. $(\forall\,\exists)$ form, **CS-ISsy** is complete. As a category **CS-ISsy** is equivalent to the category of bounded-complete ω-continuous cpo's [Smy77].

8.2 Functors

Functors defined on a category P extend in a rather trivial way to $\mathrm{Kar}(P)$. Indeed, if $F : P_1 \to P_2$ is a functor from P_1 to P_2, we have the extension $F' : \mathrm{Kar}(P_1) \to \mathrm{Kar}(P_2)$, where $F'((A,r)) = (F(A), F(r))$ and F' is defined on morphisms exactly as F. Obviously this extends to functors of several arguments. In suitable cases one may check that the extended functor F' satisfies the "same" universal condition as F: for example, products and coproducts in P go over to products and coproducts in $\mathrm{Kar}(P)$ (this is an exercise in [Fre64]). Furthermore, by a theorem of Scott in [Sco80], $\mathrm{Kar}(P)$ is cartesian closed if P is cartesian closed.

Suppose now that P_1 and P_2 are complete information categories and that $F : P_1 \to P_2$ is an object-continuous (morphism-continuous) functor. We will show that $F' : \mathrm{Kar}(P_1) \to \mathrm{Kar}(P_2)$ is also continuous. Let $\langle (A_i, r_i) \rangle_{i \geq 0}$ be an increasing chain of objects in $\mathrm{Kar}(P_1)$. Then we have:

$$\begin{array}{rcl} F'(\bigcup_i (A_i, r_i)) & = & F'((\bigcup_i A_i, \bigcup_i r_i)) \\ & = & (F(\bigcup_i A_i), F(\bigcup_i r_i)) \\ & = & (\bigcup_i F(A_i), \bigcup_i F(r_i)) \\ & = & \bigcup_i F'((A_i, r_i)) \end{array}$$

so that F' is continuous on objects; continuity over morphisms is a trivial extension of this.

Note, however that in the context of information categories, e.g. in **CS-ISys**, the notion of the substructure relation between objects i.e.

$$(I,<) \trianglelefteq (I',<') \qquad \text{iff} \qquad I \trianglelefteq I' \ \&\ < \,=\, <' \cap (A \times A)$$

is not categorical and therefore we cannot deduce by the usual categorical methods that standard functors between two I-categories give rise to standard functors between their Karoubi envelopes, i.e. cannot in general deduce that $F(<) = F(<') \cap (F(A) \times F(A))$. Although it is possible to impose some categorical conditions which would imply the above equation, we will not dwell on them and prefer to verify in each particular case that the usual functors are standard. For continuous Scott information systems, we will treat here the function space which is the most subtle amongst the usual functors.

8.3 Function space constructor

We will construct $(-) \to_{\mathrm{C}} (-)$, the function space functor on **CS-ISys***, the Karoubi envelope of **BC-ISys***, using $(-) \to (-)$, the function space functor on **BC-ISys*** (see Section 5.3). Given objects $(I, <)$ and $(I', <')$ of **CS-ISys***, where $<$ and $<'$ are retractions on the bounded complete information systems I and I' respectively, we let

$$((I, <) \to_{\mathrm{C}} (I', <')) = (I \to I', < \!\to\! <'),$$

where the approximable mapping

$$< \!\to\! <' \colon (I \to I') \to (I \to I')$$

is defined by $f(< \!\to\! <')g$ iff $g \subseteq <; f; <'$. It is easy to check that $< \!\to\! <'$ is a retraction i.e. $(I \to I', < \!\to\! <')$ is an object of **CS-ISys***. To prove that $(-) \to_{\mathrm{C}} (-)$ is standard, suppose $(I, <_I) \trianglelefteq (J, <_J)$ and $(I', <_{I'}) \trianglelefteq (J', <_{J'})$, i.e. $I \trianglelefteq J$, $<_I = <_J \cap (|I| \times |I|)$, $I' \trianglelefteq J'$, and $<'_I = <'_J \cap (|I'| \times |I'|)$. We now have

$$(<_I \!\to\! <_{I'}) = (<_J \!\to\! <_{J'}) \cap (|I \to I'| \times |I \to I'|),$$

since

$$f(<_I \!\to\! <_{I'})g \iff g \subseteq <_I; f; <_{I'} \iff g \subseteq <_J; f; <_{J'} \iff f(<_J \!\to\! <_{J'})g.$$

Therefore $(-) \to_{\mathrm{C}} (-)$ is standard and we define

$$(\mathrm{in}((I, <_I), (J, <_J)) \to_{\mathrm{C}} \mathrm{in}((I', <_{I'}), (J', <_{J'}))) = \mathrm{in}((I \to I', <_I \!\to\! <_{I'}), (J \to J', <_J \!\to\! <_{J'})).$$

It can be easily checked that our construction captures the intended meaning of the function space; for details see [ES91b, page 49].

9 Final remarks

As stated in the introduction, information systems were developed in the context of domains for denotational semantics, as a means of presenting such domains. The majority of the examples presented above are of this type; it is regretted that this has the consequence that they are unlikely to be familiar to most mathematicians. But as we have seen in the case of Boolean algebras/Stone spaces, the technique may be available also for the types of spaces more usually studied by mathematicians. The key to this is really very simple. In constructing the points (elements) of a *domain*, we in effect use arbitrary filters, or consistent theories, of the information system (see the discussion of propositional languages in the introduction). However, with more structure available in the information system, we have the possibility of constructing points in a more sophisticated way, for example via *prime* filters (as in Stone duality). As a more elaborate example one could cite the use of proximity lattices to represent stably compact spaces via "proximal" filters [Smy92]. By adding metric structure to information systems in an appropriate way, certain classes of metric spaces can be captured, via a notion of Cauchy filter [ES93a]. Thus the predominance of specialized computer science examples may be seen as a historical accident of the development of the technique, rather than as essential to it.

References

[AGV72] M. Artin, A. Grothendick, and J. L. Verdier. Théorie des topos et cohomologie étale des schémas (sga 4). *Lecture Notes in Mathematics*, 269:413, 1972.

[Bar77] K. J. Barwise. An introduction to first order logic. In K. J. Barwise, editor, *The Handbook of Mathematical Logic*, Studies in Logic and Foundations of Mathematics, pages 5–46. North Holland, 1977.

[BC81] G. Berry and P.-L. Curien. Sequential algorithms on concrete data structures. Technical report, Report of Ecole Nationale Superieure des Mines de Paris, Centre de Mathematiques Appliquées, Sophia Antipolis, 1981.

[Ber78] G. Berry. Stable Models of Typed λ-calculi. In *Proceedings of the 5th International Colloquium on Automata, Languages and Programming*, volume 62 of *Lecture Notes in Computer Science*, pages 72–89. Springer Verlag, 1978.

[BV89] A. Batarekh and V.S.Subrahmanian. The query topology in logic programming. In B. Monien R. Cori, editor, *STACS 89*, pages 375–387. Springer-Verlag, 1989. Lecture Notes in Computer Science Vol.349.

[CGW87] T. Coquand, C. Gunter, and G. Winskel. dI-domains as a model of polymorphism. In *Third Workshop on the Mathematical Foundations of Programming Language Semantics*, pages 344–363. Springer-Verlag, 1987.

[DP90] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 1990.

[ES91a] A. Edalat and M. B. Smyth. Categories of information systems. In D. H. Pitt, P. L. Curien, S. Abramsky, A. M. Pitts, A. Poigne, and D. E. Rydeheard, editors, *Category theory and computer science*, pages 37–52. Springer-Verlag, 1991.

[ES91b] A. Edalat and M. B. Smyth. Categories of information systems. Technical Report Doc 91/21, Department of Computing, Imperial College, 1991.

[ES93a] A. Edalat and M. B. Smyth. Compact metric information systems. In *Semantics: Foundations and Applications, REX workshop*, volume 666 of *Lecture Notes in Computer Science*, pages 154–173. Springer-Verlag, 1993.

[ES93b] A. Edalat and M. B. Smyth. I-categories as a framework for solving domain equations. *Theoretical Computer Science*, 115:77–106, 1993.

[FG82] M. P. Fourman and R. J. Grayson. Formal spaces. In A. S. Trolstra and D. van Dalen, editors, *The L.E.J.Brouwer Centenary Symposium*, pages 107–121. North Holland, 1982.

[Fre64] P. Freyd. *Abelian categories: An introduction to the theory of functors.* Harper and Row, New York, 1964.

[Gra79] G. Graetzer. *Universal algebra.* Springer Verlag, Berlin, Heidelberg, New York, 2nd edition, 1979.

[Gun87] C. Gunter. Universal profinite domains. *Information and Computation*, 72(1):1–30, 1987.

[Hal63] P. R. Halmos. *Lectures on Boolean Algebras.* Number 1 in Van Nostrand Mathematical Studies. Princeton, 1963.

[Jun88] A. Jung. *Cartesian Closed Categories of Domains.* PhD thesis, Technische Hochschule Darmstadt, 1988.

[Llo87] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1987.

[LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic.* Cambridge Studies in Advanced Mathematics Vol. 7. Cambridge University Press, 1986.

[LW84] K. G. Larsen and G. Winskel. Using information systems to solve recursive domain equations effectively. In D. B. MacQueen G. Kahn and G. Plotkin, editors, *Semantics of Data Types*, pages 109–130, Berlin, 1984. Springer-Verlag. Lecture Notes in Computer Science Vol. 173.

[MA86] E. Manes and M. A. Arbib. *Algebraic Approaches to Program Semantics.* Springer-Verlag, 1986.

[Plo81] G. D. Plotkin. Post-graduate lecture notes in advanced domain theory (incorporating the "Pisa Notes"). Dept. of Computer Science, Univ. of Edinburgh. Available in LaTeX, 1981.

[Sco72] D. S. Scott. Continuous lattices. In F.W. Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, pages 97–136. Springer-Verlag, Berlin, 1972. Lecture Notes in Mathematics 274.

[Sco80] D. S. Scott. Lambda calculus: Some models, some philosophy. In J. Barwise, H. J. Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 223–265. North-Holland Publishing Company, 1980.

[Sco82] D. S. Scott. Domains for denotational semantics. In M. Nielson and E. M. Schmidt, editors, *Automata, Languages and Programming: Proceedings 1982*. Springer-Verlag, Berlin, 1982. Lecture Notes in Computer Science **140**.

[Smy77] M. B. Smyth. Effectively given domains. *Theoretical Computer Science*, 5:257–274, 1977.

[Smy83] M. B. Smyth. The largest cartesian closed category of domains. *Theoretical Computer Science*, 27:109–119, 1983.

[Smy92] M. B. Smyth. Stable compactification i. *J. London Math. Soc. (2)*, 45:321–340, 1992.

[SP82] M. B. Smyth and G. D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM J. Computing*, 11:761–783, 1982.

[Ten91] R. D. Tennent. *Semantics of Programming Languages.* Prentice Hall, 1991.

[Win86] G. Winskel. Event structures. In *Foundations of Software technology and Theoretical Computer Science*, pages 325–392, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Vol. 255.

[Zha89] G. Q. Zhang. dI-domains as information systems. In G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors, *Automata, Languages and Programming*, volume 372 of *Lecture*

Notes in Computer Science, pages 773–788, Berlin, 1989. Springer Verlag.