Subdomains for Algebraic Information Systems

Michael A. Bukatin
Department of Computer Science
Brandeis University
Waltham, MA 02254
e-mail: bukatin@cs.brandeis.edu

Abstract

In 1976 Dana Scott suggested using retractions to describe subdomains, the use of closure operations being suggested in the algebraic case [Scott 76]. Both choices were only marginally understood at that time. The present paper shows that the use of information systems (see [Scott 82]) naturally leads to closure operations in the algebraic case. Specifically, subdomains are formed due to the fact that some elements lose consistency or deductive closure and thus, are excluded from an original domain, $|A|$. The inconsistency in $|A|$ is treated as the entailment of the contradiction in $|A_\top| = |A| \cup \{\top\}$. Thus, the subdomains of $|A|$ correspond to the intensifications of the entailment relation in $|A_\top|$. These intensifications are approximable mappings and the corresponding continuous functions are precisely the closure operations on $|A_\top|$ ($\lambda x.\top$ is excluded). A closure operation $|r| : |A_\top| \to |A_\top|$ yields the subdomain $|R| \subseteq |A|$, $|R| = \mathrm{Fix}(|r|) \setminus \{\top\}$.

The domain of all subdomains, $|Sub(A)|$, is constructed for every domain $|A|$. $|Sub(A)|$ is a subdomain of $|A_\top \to A_\top|$, and its elements are approximable mappings representing subdomains of $|A|$. The fundamental property of $|Sub(A)|$ is that $r \sqsubseteq_{Sub(A)} s$ iff $|R| \supseteq |S|$ for the subdomains $|R|$ and $|S|$ corresponding to r and s.

The following related matters are also discussed: the idea of data types as objects, the projection of $|Sub(A)|$ onto $|A|$ and the equations such as $|X| \cong |Sub(X)|$, the finitary projections and why their fixed points do not form subdomains, and the somewhat adverse relationship between retraction-based subtypes and intensional type theories.

0. INTRODUCTION. In 1969 Dana Scott and Christopher Strachey introduced continuous cpo's as models for data types and continuous functions between them as models for computable functions. The continuous cpo's (known as *domains*) are ordered by "informational" relations of approximation, $\sqsubseteq$, where $x \sqsubseteq y$ stands for *all information that is true about x, is true about y.* Particular *axioms* governing domains differed and were determined by informal considerations of the underlying notions of "information", "approximation", etc.

In 1982 Dana Scott introduced an alternative to this axiomatic approach in the case of algebraic, consistently complete cpo's [Scott 82] (see [Hoofman 93] for a generalization of this alternative approach to the continuous case). [Scott 82] presents *information systems* that explicitly describe "pieces of information", called *tokens* or *assertions*, and their logic, *consistency* and *entailment* relations. Elements of domains are then identified with sets of assertions that are true about these elements. Domains are explicitly built from information systems by taking all suitable sets of assertions, that is, consistent deductively closed sets, as elements. Continuous functions are described by relations of entailment between assertions about an input argument of a function and assertions about the corresponding output. Such input-output entailment relations are called *approximable mappings* because of their ability to form functional spaces and, hence, to approximate each other. Properties of domains and continuous functions are subsequently inferred, rather than postulated.

The approach involving information systems brings numerous advantages. Among them are concrete representations of domains, a clearer intuition about them, and easier ways to find, to describe, and to motivate various constructions. For a further discussion of these merits see [Scott 82; Larsen, Winskel 84]. A trade-off is involved, however, — the information systems yield less canonical representations of domains. Dealing with domains as abstract cpo's, we usually do not bother to distinguish between isomorphic and identical domains. We might make this distinction more frequently regarding domains defined by information systems, because even the cardinalities of the sets of assertions might differ in descriptions of isomorphic domains. We should say that this failure to distinguish between isomorphic and identical domains usually goes regardless of a representation of domains, when we consider domains of domains, because they contain isomorphic, albeit different, domains as different points.

Our notation mainly follows one in [Larsen, Winskel 84] (see Section 1 for notation and definitions).

The problem. In this paper, we study the question "Which subsets of a domain should be considered subdomains?" for the domains described by Scott's information systems (algebraic information systems in the terminology of [Hoofman 93]). One criterion for a good answer is a requirement that all subdomains of a domain, $|A|$, should themselves form a domain, $|Sub(A)|$. We would also like to be able to solve domain equations such as $|X| \cong |Sub(X)| + \ldots$. It will be shown in the next few paragraphs that the framework of domains as abstract cpo's does not provide sufficient hints for the solution, where information systems do.

The systematic use of the operation of taking a certain subset of a domain to form another domain and the systematic consideration of domains, whose elements are domains, first appear in the mid-'70's [Scott 76; Shamir, Wadge 77]. It is notable that nobody has tried to consider arbitrary subsets of a domain, $|A|$, that would satisfy a particular version of domain axioms, although at first glance it would seem to be the most intuitive and general version of a subdomain definition in the framework of domains as abstract cpo's. The reason why nobody has considered such a definition is that such arbitrary subsets might satisfy the domain axioms for random causes, and their collection would be totally unmanageable; in particular, one cannot hope to form a domain, $|Sub(A)|$, with these subsets as elements.

The next idea, which originated in [Scott 76], is to define a subdomain as a set of fixed points of a *retraction* belonging to a certain class. There are two problems here: a) What are the reasons for using retractions? b) What is the appropriate class of retractions? None of these problems receives a clear answer in the framework of domains as abstract cpo's. On the use of retractions Scott writes: "it seems almost an accident that the idea [to use the retractions] can be applied" [Scott 76, p.540].

For the algebraic case, Scott suggests using *closure operations*, that is, retractions $|r|$, such that $|r| \sqsupseteq id$. This suggestion can be motivated by the theorem [Scott 76, Theorem 5.1] saying that the fixed points of a closure operation [on an algebraic lattice] form an algebraic lattice. Although this is not true for arbitrary retractions or projections (i.e. retractions $|r|$, such that $|r| \sqsubseteq id$), if we consider *finitary* retractions or projections (*finitary* means algebraicity of the set of fixed points), then this argument no longer favors closure operations.

In the paper entitled "Data Types as Objects" [Shamir, Wadge 77], the authors suggest that to describe the systems of polymorphic types, one should consider data types that incorporate their own subtypes as elements, and allow an element of a data type to belong simultaneously to different subtypes of this data type. At the end of this Introduction, we discuss the rather adverse relations between retraction-based approaches to subtyping ([Scott 76], [Shamir, Wadge 77], and the present paper are all based on retractions) and the dominant form of modern theories of typing that deals with types of intensional objects (the type polymorphism in ML is, probably, the most widely known example).

Let us briefly present the main features of subtypes in [Shamir, Wadge 77]. The subtype $|Y| = \{x \in |D| \mid x \sqsubseteq y\}$ is associated with each element y of a data type $|D|$. We can view such subtype as the set of fixed points of the (finitary, in the algebraic case) projection $x \mapsto x \sqcap y$. Subtypes are ordered simply by inclusion. Thus, this simpleminded approach yields the result that $|D| \cong |Sub(D)|$ for every domain $|D|$. Each element represents the subtype of its approximations, and an element x belongs to all subtypes $|Y|$ of $|D|$ that are represented by y's, such that $x \sqsubseteq y$.

The solution. The approach based on information systems yields a result that is similar to the Scott's original suggestion to use closure operations for the algebraic case. Specifically, we will show that subdomains of a domain $|A|$ can be obtained as sets of fixed points of closure operations on the domain $|A_\top|$ ($|A|$ with an extra finite top element $\top$). We, of course, should exclude $\top$ from each of these sets of fixed points. We should also exclude the closure operation $\lambda x.\top$, because it would yield an empty subdomain barred by our definitions.

The definition that allows us to obtain this result is the most intuitive definition from the viewpoint provided by information systems:

Definition 0.1. Let A be an information system. Let S be a subset of the corresponding domain $|A|$. Then we call S a *subdomain* of $|A|$ iff there exists an information system B, such that $S = |B|$. (Note that the last equality is an exact equality of sets, and not just isomorphism of partial orders.)

Let us consider informally, what should we do to an information system A to obtain a subdomain $|B| \subseteq |A|$. We would like to destroy some elements of $|A|$. The elements are consistent and deductively closed sets of assertions in A. Therefore, some elements of $|A|$ must lose their consistency or deductive closure to be destroyed. Also notice, that the loss of deductive closure means intensification of entailment.

These informal considerations are backed by Lemma 2.1, which states that $|B| \subseteq |A|$ iff all consistent sets of tokens in B are consistent in A and the entailment relation $\vdash_B$ is stronger (or equal) than $\vdash_A$ restricted onto consistent sets of tokens in B.

We can treat inconsistency as a partial case of entailment, namely the entailment of a contradiction. Consider an information system $A_\top$ that has an extra token ∇ expressing "false" (sets of tokens that are inconsistent in A entail ∇ in $A_\top$; ∇ entails all tokens); then $|A_\top|$ is just $|A|$ with a new finite top element $\top = \overline{\{\nabla\}}$ added. Thus, the loss of consistency and the loss of deductive closure in A can both be treated as cases of intensification of the entailment relation in $A_\top$.

The entailment relation, $\vdash^*$, in any domain is an approximable mapping and the corresponding continuous function is just id (see Lemma 1.10). The intensifications of the entailment relation can also be treated as approximable mappings:

Definition 0.2. An approximable mapping $r : A_\top \to A_\top$ is called an *intensified entailment relation (IER)* on $A_\top$, if

(i) $u \vdash^*_{A_\top} v \Rightarrow urv$ (r is stronger than $\vdash^*_{A_\top}$);

(ii) $urv, vrw \Rightarrow urw$ (transitivity of entailment);

(iii) $\emptyset \not r \{\nabla\}$ (non-degeneracy condition — empty set must remain consistent in A in order to satisfy Definition 1.1(i); this can be read as "no empty subdomains").

Continuous functions corresponding to IERs are closure operations (the non-degeneracy condition excludes the closure operation $\lambda x.\top$):

Definition 0.3. A continuous function $|r| : |A_\top| \to |A_\top|$ is called a *non-trivial closure operation* on $A_\top$, if

(i) $|r| \sqsupseteq id$;

(ii) $|r| \circ |r| = |r|$;

(iii) $|r| \neq \lambda x.\top$ (or, equivalently, $|r|(\bot) \neq \top$).

The main theorem of Section 2 (Theorem 2.5) gives the following characterization of subdomains. It states that the subdomains of $|A|$ are in 1-1 correspondence with the IERs on $A_\top$ and therefore, with the non-trivial closure operations $|r| : |A_\top| \to |A_\top|$. Moreover, this correspondence can be expressed by the formula $|R| = \mathrm{Fix}(|r|) \setminus \{\top\}$.

This immediately yields the variation of [Scott 76, Theorem 5.1] stating that sets of fixed points of closure operations (on algebraic domains) are algebraic.

This material is presented in Section 2. Section 3 is devoted to the domain of subdomains.

Domain of subdomains. Obviously, IERs on $A_\top$ form a subset of $|A_\top \to A_\top|$, and remarkably (Theorem 3.1) they form a subdomain of $|A_\top \to A_\top|$. This is the only theorem in the present paper with a somewhat non-trivial proof. This theorem is a generalization of [Scott 76, Theorem 5.5]. Taking into account the 1-1 correspondence between the IERs on $A_\top$ and the subdomains of $|A|$ we call this subdomain of $|A_\top \to A_\top|$ the *domain of subdomains* of $|A|$ and denote it as $|Sub(A)|$.

Among the most important properties of $|Sub(A)|$ is the fact (Theorem 3.2) that for any $r, s \in |Sub(A)|$ $r \sqsubseteq_{Sub(A)} s$ iff $|R| \supseteq |S|$ for the corresponding subdomains. This differs favorably from [Shamir, Wadge 77] and from the other approaches to subdomains as sets of fixed points of projections, where subdomains are ordered by inclusion. Indeed, the smaller a subdomain is as a set, the more information we have about each of its elements, and hence, the larger this subdomain should be informationally ($integers \sqsupseteq reals$ is desirable).

We can construct a projection of $|Sub(A)|$ onto $|A|$, although some light non-triviality is involved (Theorem 3.3 and the related discussion). We can use the inverse limit construction to obtain solutions of domain equations involving Sub, e.g. $|X| \cong |Sub(X)|$, $|X| \cong |Sub(X)| + |A|$, etc. Because $Sub(A)$ represents all possible ways to intensify our means of entailment in A, it might be of interest to study explicitly the iterations of this construction, $Sub(Sub(A))$, $Sub^3(A)$, etc., and their limit D ($|D| \cong |Sub(D)|$).

Discussion. In the Appendix, we explain why the sets of fixed points of finitary projections do not generally form true subdomains. Note that this reasoning is not applicable to the simple projections from [Shamir, Wadge 77] $x \mapsto x \sqcap y$. Indeed, the sets $\{x \in |A| \mid x \sqsubseteq y\}$ do form subdomains. The corresponding closure operation is $|r_y| = \lambda x : |A_\top|.if\ x \sqsubseteq y\ then\ x\ else\ \top$. This yields a reversal of the relation of approximation between subdomains in [Shamir, Wadge 77]. Certainly, we cannot simply reverse the ordering in a domain and still have the axioms satisfied, so the addition of many other subdomains is necessary.

We conclude this Introduction with a discussion of the relationship between the retraction-based approaches to subtyping and modern systems of types of intensional objects (ML types, etc.). At this point, these two ways of thinking about subtypes seem to be orthogonal and essentially incompatible. Indeed, the retraction-based approach suggests the extensional view of types, namely that in the type $|A \to B|$ there is at most one element representing a particular function from $|A|$ to $|B|$. On the other hand, when we deal with types of intensional objects (programs), even if we start with an extensional model, say, $|D| \cong |D \to D|$ for the underlying untyped λ-calculus, we say that a function f (think about f as a denotation of a λ-term, $f \in |D \to D|$) has the type

$A \to B$ iff $\forall x \in |A| \subseteq |D|.f(x) \in |B| \subseteq |D|$. Then, obviously, we can have two functions f and g of type $A \to B$, such that $\forall x \in |A|.f(x) = g(x)$, but for some $x_0 \in |D| \setminus |A| \quad f(x_0) \neq g(x_0)$. Thus, the same function from $|A|$ to $|B|$ has different representations, f and g.

Intensional type systems are characterized by the contravariance of subtyping: the rule $f : A \to B$ iff $\forall x \in |A| \subseteq |D|.f(x) \in |B| \subseteq |D|$ and the rule $A \trianglelefteq B$ (A is a subtype of B) iff $\forall x \in |A|.x \in |B|$ together imply that $A \trianglelefteq B \Rightarrow (B \to C) \trianglelefteq (A \to C)$. The simple cardinality considerations show that, if subtypes are subsets, this contravariance should not be compatible with extensionality. Indeed, if $Card(A) \leq Card(B)$, we have $Card(C^A) \leq Card(C^B)$, and we cannot expect that the restriction of the class of acceptable functions would reverse this relation.

But even if we forgo extensionality, retractions still give some weak version of covariance, and it is hard to believe that this can be consistently combined with the contravariance property. Indeed, if $A \trianglelefteq B$ means $|A| = Fix(|r|)$, where $|r|$ is a retraction of $|B|$, and, in turn, $|B| = Fix(|s|)$, where $|s|$ is a retraction of the universal type, $|D|$, then for every $f : A \to C$ we can construct $f' : B \to C$ (keep in mind, that $|f|, |f'| : |D| \to |D|$) by taking $|f'| = \lambda x : |D|.|f|(|r|(|s|(x)))$. Then $|f'|$ maps all elements of $|B|$ (and, actually, all elements of $|D|$) into $|C|$, hence $f' : B \to C$, and $\forall x \in |A|.|f'|(x) = |f|(x)$, so f' is extensionally equivalent to f as a function from $|A|$ to $|C|$. It is not quite $(A \to C) \trianglelefteq (B \to C)$, but it is close enough. Because in the present paper subtypes of $|A|$ are retractions of $|A_\top|$ and not of $|A|$, matters are actually more complicated that this, and covariance does not generally hold, but the basic incompatibility with contravariance remains.

I do not think that the relationship between extensional types and typing of intensional objects is well understood at this point. [Castagna 94] is worth mentioning as a recent effort to reconcile covariance and contravariance in the setting of object-oriented programming. There is a number of papers developing semantics of second-order λ-calculus based on retractions of domains (see [Coquand, Gunter, Winskel 89] and references therein). It would be interesting to study the relationship between this kind of semantics and the contravariance of subtyping.

1. DEFINITIONS AND NOTATION. In this brief review, we mainly follow [Larsen, Winskel 84]. Notions and results of this section should be mostly credited to Dana Scott [Scott 82]. These sources contain extensive discussions of the subject.

Notation 1.0. If D is a set, $P(D)$ is a set of all its subsets, $P_{Fin}(D)$ is a set of all its finite subsets. $u \subseteq_{Fin} x$ means that a set u is a finite subset of a set x.

Definition 1.1. $A = (D_A, Con_A, \vdash_A)$, where D_A is a set of distinctive *tokens* (*assertions*), $Con_A \subseteq P_{Fin}(D_A)$ (the *consistent* finite conjunctions of assertions), $\vdash_A \subseteq Con_A \times D_A$ (the *entailment* relation), is called an *information system* (*IS*) if

(i) $\emptyset \in Con_A$ (non-degeneracy; empty domains are barred);

(ii) $\forall d \in D_A.\{d\} \in Con_A$ (no "junk");

(iii) $\forall u \subseteq v \in Con_A.u \in Con_A$;

(iv) $\forall d \in u \in Con_A.u \vdash_A d$ (reflexivity of entailment);

(v) $\forall u \in Con_A, d \in D_A.u \vdash_A d \Rightarrow u \cup \{d\} \in Con_A$ (entailment preserves consistency);

(vi) $\forall u, v \in Con_A, d \in D_A.(\forall d' \in v.u \vdash_A d'), v \vdash_A d \Rightarrow u \vdash_A d$ (transitivity of entailment).

Notation 1.2. We avoid the usual overloading of $\vdash_A$. Define $\vdash_A^* \subseteq Con_A \times Con_A$ by $u \vdash_A^* v$ iff $\forall d \in v.u \vdash_A d$.

Definition 1.3. Consider an information system A.

$|A| = \{x \subseteq D_A \mid$

(i) $\forall u \subseteq_{Fin} x.u \in Con_A$ (consistency);

(ii) $\forall u \subseteq_{Fin} x.\forall d \in D_A.u \vdash_A d \Rightarrow d \in x$ (deductive closure) $\}$

is called a *domain* described by the information system A.

For $x, y \in |A|$ define $x \sqsubseteq_A y$ iff $x \subseteq y$.

Definition 1.4. For any $x \subseteq D_A$ possessing the property of consistency ($\forall u \subseteq_{Fin} x.u \in Con_A$), $\overline{x} = \overline{x_A} = \{d \in D_A \mid \exists u \subseteq_{Fin} x.u \vdash_A d\}$ is called a *deductive closure* of x (in A).

One can see that $\overline{x} \in |A|$.

If $u \in Con_A$ then $\overline{u}$ is called a *finite element* of $|A|$.

Notation 1.5. $\perp_A = \overline{\emptyset_A}$. It is easy to see, that $\perp_A$ is the least element of $|A|$.

Exercise 1.6. Show that for $x_1, \ldots, x_n, \ldots \in |A|$, such that $x_1 \sqsubseteq \ldots \sqsubseteq x_n \sqsubseteq \ldots$, there is the exact upper bound $\sqcup\{x_i \mid i \in N\} = \cup\{x_i \mid i \in N\} \in |A|$.

We have defined *information systems* and *domains*, and now proceed to the definition of *continuous functions*. We start with the definition of *input-output entailment relations* which are also called *approximable mappings* (mathematicians should be warned that these *mappings* are *not* functions).

Definition 1.7. Consider information systems A and B. $f \subseteq Con_A \times Con_B$ is called an *approximable mapping* (or *input-output entailment relation*) between A and B (we write $f : A \rightarrow B$) if

(i) $\emptyset f \emptyset$;

(ii) $ufv, ufw \Rightarrow (v \cup w) \in Con_B, uf(v \cup w)$;

(iii) $u' \vdash_A^* u, ufv, v \vdash_B^* v' \Rightarrow u'fv'$ (transitivity of the input-output entailment relation with the entailment relations in A and B).

Definition 1.8. Let f be an approximable mapping. Then $|f| : |A| \rightarrow |B|$ defined by $|f|(x) = \{d \in D_B \mid \exists u \subseteq_{Fin} x.uf\{d\}\}$ is called a *continuous function* from $|A|$ to $|B|$.

Exercise 1.9. Show that $f(x) \in |B|$ (correctness). Show that $|f|(x) = \cup\{v \in Con_B \mid \exists u \subseteq_{Fin} x.ufv\}$ yields an equivalent definition of a continuous function. Show that f is monotonic and preserves the exact upper bounds of sequences from Exercise 1.6. Show that $f \subseteq g \Leftrightarrow \forall x \in |A|.|f|(x) \sqsubseteq_B |g|(x)$.

Lemma 1.10. 1) $| \vdash_A^* | = id$. 2) $|g| \circ |f| = |g \circ f|$, where $\forall u \in Con_A, w \in Con_C. u(g \circ f)w \Leftrightarrow \exists v \in Con_B. ufv, vgw$. In particular, a continuous function $|r| : |A| \to |A|$ is a retraction ($|r| = |r| \circ |r|$) iff $\forall u, w \in Con_A. urw \Leftrightarrow \exists v \in Con_A. urv, vrw$.

Definition 1.11. Consider information systems A and B. Then the *information system of approximable mappings*, $A \to B$, is defined as follows:

$$D_{A \to B} = Con_A \times Con_B;$$

$$\{(u_1, v_1), \ldots, (u_n, v_n)\} \in Con_{A \to B} \quad \text{iff}$$
$$\forall I \subseteq \{1, \ldots, n\}. \cup\{u_i \mid i \in I\} \in Con_A \Rightarrow \cup\{v_i \mid i \in I\} \in Con_B;$$

$$\{(u_1, v_1), \ldots, (u_n, v_n)\} \vdash_{A \to B} (u, v) \text{ iff } \cup \{v_i \mid u \vdash_A^* u_i\} \vdash_B^* v.$$

Exercise 1.12. Prove that $A \to B$ satisfies the definition of an information system. Prove that $|A \to B| = \{f \subseteq Con_A \times Con_B \mid f$ is an approximable mapping between A and $B\}$. Prove that different approximable mappings define different continuous functions, and therefore, approximable mappings and continuous functions are in 1-1 correspondence.

The reader who can solve Exercises 1.6, 1.9, 1.12 should be able to complete all missing proofs and parts of proofs in the present paper.

Notation 1.13. $|A \to B|$ is the domain of all approximable mappings between A and B, but, due to the 1-1 correspondence between approximable mappings and continuous functions, it is said that $|A \to B|$ is the *domain of continuous functions* between $|A|$ and $|B|$, and it is called the *functional space.*

Definition 1.14. A pair of continuous functions $|i| : |B| \to |A|, |j| : |A| \to |B|$ is said to be a *retraction* of $|A|$ *onto* $|B|$ if $|j| \circ |i| = id_B$.

Lemma 1.15. If $|i|, |j|$ form a retraction of $|A|$ onto $|B|$, then $|r| = |i| \circ |j|$ is a retraction of $|A|$ ($|r| = |r| \circ |r|$).

Definition 1.16. If $|i|, |j|$ form a retraction of $|A|$ onto $|B|$, and the retraction $|r|$ from Lemma 1.15 is a projection ($|r| \sqsubseteq id_A$), then we say that $|i|, |j|$ form a *projection* of $|A|$ *onto* $|B|$.

Definition 1.17. Consider a domain $|A|$ and a retraction (resp. projection) $|r| : |A| \to |A|$. If there exist a domain $|B|$ and a pair of continuous functions $|i|, |j|$ that forms a retraction of $|A|$ onto $|B|$, such that $|r| = |i| \circ |j|$, then $|r|$ is called a *finitary retraction* (resp. *finitary projection*).

2. SUBDOMAINS. We use the Definition 0.1 of a subdomain. The following Lemma expresses the fact that subdomains can be obtained via the exclusion of some elements by making them lose their consistency or deductive closure. The loss of deductive closure corresponds to intensification of entailment from the consistent sets of assertions in B.

Lemma 2.1. For information systems A, B $|B| \subseteq |A|$ iff

(i) $D_B \subseteq D_A$;

(ii) $Con_B \subseteq Con_A$;

(iii) $\forall u \in Con_B. \forall d \in D_A. u \vdash_A d \Rightarrow u \vdash_B d$.

If an information system B describing $S \subseteq |A|$ in this way exists, then it is unique.

The following construction allows us to treat inconsistency as an entailment of a contradiction.

Definition 2.2. Consider an information system A. Define the information system $A_\top$ by

$$D_{A_\top} = D_A \cup \{\nabla\}, \nabla \notin D_A \text{ (∇ stands for "false")};$$
$$Con_{A_\top} = P_{Fin}(D_{A_\top});$$
$$u \vdash_{A_\top} d \text{ iff } u \vdash_A d \text{ or } u \notin Con_A.$$

Lemma 2.3. $|A_\top| = |A| \cup \{\top\}$, where $\top = \overline{\{\nabla\}} = D_{A_\top}$ is the top element of $|A_\top|$.

Both the loss of consistency and the loss of deductive closure in A can be treated as cases of intensification of the entailment relation in $A_\top$. Definition 0.2 introduces *intensified entailment relations* (IERs) on $A_\top$ — approximable mappings from $A_\top$ to $A_\top$ corresponding to intensifications of entailment in $A_\top$ (in the same way, the approximable mapping $\vdash^*_{A_\top}$ corresponds to $\vdash_{A_\top}$).

Lemma 2.4. Continuous functions corresponding to IERs are precisely *non-trivial closure operations*, introduced by Definition 0.3.

Theorem 2.5. The subdomains of $|A|$ are in 1-1 correspondence with the IERs on $A_\top$, and therefore, with the non-trivial closure operations on $|A_\top|$. This correspondence is defined by

(i) Given an IER $r : A_\top \to A_\top$, construct the information system R such that $|R| \subseteq |A|$ as follows:

$$D_R = \{d \in D_{A_\top} \mid \{d\} \not r \{\nabla\}\};$$
$$Con_R = \{u \in Con_{A_\top} \mid u \not r \{\nabla\}\};$$
$$\forall u \in Con_R, d \in D_R. u \vdash_R d \text{ iff } ur\{d\}.$$

(ii) Given an information system R such that $|R| \subseteq |A|$, construct the IER $r : A_\top \to A_\top$ as follows:

$$urv \text{ iff } u \vdash^*_R v \text{ or } u \notin Con_R.$$

Theorem 2.6. The correspondence between the subdomains of $|A|$ and the non-trivial closure operations on $|A_\top|$ is described by the formula $|R| = \text{Fix}\,(|r|) \setminus \{\top\}$.

Corollary 2.7. Fixed points of a closure operation on any domain $|A|$ form a subdomain of $|A|$. Closure operations, therefore, are finitary retractions.

3. DOMAIN OF SUBDOMAINS.

Theorem 3.1. The set of all IERs on $A_\top$ is a subdomain of $|A_\top \to A_\top|$. Because the IERs on $A_\top$ are in 1-1 correspondence with the subdomains of $|A|$, we call this domain the *domain of subdomains* of $|A|$ and denote it by $|Sub(A)|$.

The draft of the proof. We must construct the intensified entailment relation $sub(A)$ on $(A_\top \to A_\top)_\top$, and then prove that $|Sub(A)| = \{\text{IERs on } A_\top\}$.

We start with the *elementary entailment relation* $\succ$, expressing desirable properties of $sub(A)$. Then we obtain $sub(A)$ via a variant of usual transitive closure of $\succ$.

Define $\succ \subseteq Con_{A_\top \to A_\top} \times D_{A_\top \to A_\top}$ as follows: $\{(u_1, v_1), \dots, (u_n, v_n)\} \succ (u, v)$ iff one of the following conditions holds:

(i) $u \vdash^*_{A_\top} v$ (corresponds to Definition 0.2(i));

(ii) $\cup\{v_i \mid u \vdash^*_{A_\top} u_i\} \vdash^*_{A_\top} v$ (corresponds to definition of $\vdash_{A_\top \to A_\top}$ — see Definition 1.11);

(iii) $\exists i, j. u = u_i, v_i = u_j, v_j = v$ (corresponds to Definition 0.2(ii)).

Define $\succ_+ \subseteq Con_{A_\top \to A_\top} \times Con_{A_\top \to A_\top}$ as follows: $\omega \succ_+ \omega'$ iff $\omega' = \omega \cup \{(u, v)\}$ and $\omega \succ (u, v)$.

Define $sub(A) \subseteq Con_{A_\top \to A_\top} \times Con_{A_\top \to A_\top}$ as follows: $\omega\, sub(A)\, \omega'$ iff $\exists n \geq 0. \exists \omega_0, \dots, \omega_n \in Con_{A_\top \to A_\top}. \omega = \omega_0 \succ_+ \omega_1 \succ_+ \dots \succ_+ \omega_n$ and $[\omega' \subseteq \omega_n$ or $\nabla_{(A_\top \to A_\top)_\top} \in \omega_n$ or $(\emptyset, \{\nabla_{A_\top}\}) \in \omega_n]$. The $(\emptyset, \{\nabla_{A_\top}\}) \in \omega_n$ clause corresponds to Definition 0.2(iii).

The remainder consists of the straightforward inductive proofs showing:

(1) $sub(A)$ is an approximable mapping from $(A_\top \to A_\top)_\top$ to $(A_\top \to A_\top)_\top$;

(2) $sub(A)$ is an IER on $(A_\top \to A_\top)_\top$;

(3) $r \in |Sub(A)|$ iff r is an IER on $A_\top$.

□

Notice, that we call the domain $|Sub(A)|$ a *domain of subdomains*, where in reality $|Sub(A)|$ consists of IERs representing these subdomains. It is similar to the convention of calling $|A \to B|$ a *domain of continuous functions* from $|A|$ to $|B|$, where in reality $|A \to B|$ consists of approximable mappings representing these functions (see Notation 1.13).

Theorem 3.2. For any IERs r, s on $A_\top$ $r \sqsubseteq_{Sub(A)} s$ iff $|R| \supseteq |S|$ for the corresponding subdomains of $|A|$.

The idea behind this theorem is simple: the stronger an entailment relation, the fewer elements of $|A_\top|$ that remain deductively closed and, therefore, intact.

The meaning of Theorem 3.2 is discussed in the Introduction.

Theorem 3.3. The continuous functions $|i| : |A| \to |Sub(A)|$ and $|j| : |Sub(A)| \to |A|$ defined below yield a projection of $|Sub(A)|$ onto $|A|$. In terms of closure operations

$$|i|(y) = \lambda x : |A_\top|. x \sqcup_{A_\top} y;$$

$$|j|(r) = |r|(\bot).$$

In terms of subdomains themselves

$$|i|(y) = \{x \in |A| \mid x \sqsupseteq y\};$$

$$|j|(S) = \min_{|A|} S, \text{ where } S \text{ is a subdomain of } |A|.$$

Notice that some light non-triviality is involved in the construction of the embedding $|i| : |A| \to |Sub(A)|$. All one-element subsets of $|A|$ form subdomains (the corresponding closure operation is $|r_y| = \lambda x : |A_\top|. if\ x \sqsubseteq$

y then y else $\top_{A_\top}$). But due to Theorem 3.2, they are all incomparable (in fact, they are total elements in $|Sub(A)|$). Therefore, the function $x \mapsto \{x\}$ would not be monotonic. This implies that this function is not continuous (Exercise 1.9) and cannot be used as an embedding.

APPENDIX. Finitary projections: why sets of their fixed points do not generally form subdomains.

Let us start with a basic fact.

Lemma A.1. A continuous function $|p| : |A| \to |A|$ is a finitary projection iff the corresponding approximable mapping p has the following property:

$$upw \Rightarrow (\text{and, trivially, } \Leftarrow) \exists v \in Con_A . u \vdash^*_A v, vpv, v \vdash^*_A w.$$

Given an information system A and a finitary projection $|p| : |A| \to |A|$, consider the following construction of the information system P.

Definition A.2. $P = (D_P, Con_P, \vdash_P)$ is defined as follows:

$D_P = \{u \in Con_A \mid upu\}$;

$Con_P = \{\{u_1, ..., u_n\} \subseteq_{Fin} D_P \mid u_1 \cup \ldots \cup u_n \in Con_A\}$;

$\{u_1, ..., u_n\} \vdash_P u$ iff $u_1 \cup \ldots \cup u_n \vdash^*_A u$.

Lemma A.3. Fix($|p|$) and $|P|$ are isomorphic as algebraic cpo's.

The construction of P includes two features — conjunctive completion and elimination of assertions, such that $d \not p d$.

The first step — conjunctive completion — is not so essential and can be avoided if we start with a conjunctively complete IS A from the very beginning (A is conjunctively complete if $\forall u \in Con_A . \exists d \in D_A . u \vdash_A d, \{d\} \vdash^*_A u$). The role of conjunctive completeness can be illustrated by the following example. Consider the domain $|D| = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$. To differentiate between the projection onto $\{\emptyset, \{1,2\}\}$ and the identity projection, we need an element ∇, such that $\{\nabla\} \vdash^* \{1,2\}, \{1,2\} \vdash \nabla$. On the other hand, if we consider $|D'| = \{\emptyset, \{1\}, \{2\}, \{1,2,\nabla\}\}$, conjuctive completion would not be necessary.

Assuming the property of conjunctive completeness we can modify Definition A.2 so, that, in particular, $D_P = \{d \in D_A \mid dpd\}$. The elimination of assertions d, such that $d \not p d$, unfortunately remains essential, and we cannot get rid of it. To illustrate this, consider conjunctively complete $|D| = \{\emptyset, \{1\}, \{1,2\}, \{1,3\}\}$ and its projection onto $\{\emptyset, \{1,2\}, \{1,3\}\}$. The corresponding $|P|$ is equal to $\{\emptyset, \{2\}, \{3\}\}$, which is *isomorphic and informationally equivalent but not equal* to $\{\emptyset, \{1,2\}, \{1,3\}\}$. Note, that $\{\emptyset, \{1,2\}, \{1,3\}\}$ cannot be described by a closure operation on $|D_\top|$. In fact, $\{\emptyset, \{1,2\}, \{1,3\}\}$ cannot be obtained as domain from any information system (consider, what would be $\overline{\{1\}}$ to see that), although it is isomorphic to such a domain. We cannot thus obtain $|P| =$Fix($|p|$) and have to live with $|P| \cong$Fix($|p|$).

But we cannot afford to include subsets of $|A|$ that are just isomorphic to other domains, as subdomains of $|A|$, because this would lead to the inclusion of too many subsets of $|A|$ and an inability to manage them and to build a domain of subdomains from them.

Obviously, if we restrict ourselves by the finitary projections of $|A|$, they do form an algebraic cpo, isomorphic to the set of fixed points of a finitary projection of $|A \to A|$. Also our objections are not valid in some simple cases of finitary projection such as $x \mapsto x \sqcap y$ from [Shamir, Wadge 77]. In these cases, the resulting sets of fixed points can be described by closure operations on $|A_\top|$ as well, as we discussed in the Introduction.

ACKNOWLEDGEMENTS. I am very grateful to Alexander Saevsky and Alexander Shen for numerous fruitful discussions on Scottery in Moscow. I would like to thank Elena Machkasova, Joshua Scott, and especially Ataliya Kagan, Jeff Sanchez, and Ross Viselman, who read the initial version of the present paper and helped me with various suggestions. I am grateful to Will Clinger, Robert Muller and Mitch Wand for the helpful discussion and suggestions during the presentation of this work at Northeastern University. This work was partially supported by NSF Grant CCR-9216185.

REFERENCES:

[**Castagna 94**] Castagna G. *Covariance and contravariance: conflict without a cause.* LIENS-94-18 technical report, 1994. Available via anonymous ftp at ftp.ens.fr in the file /pub/reports/liens/liens-94-18.A4.dvi.Z.

[**Coquand, Gunter, Winskel 89**] Coquand T., Gunter C.A., Winskel G. *Domain theoretic models of polymorphism.* Information and Computation, **81**, #2, pp.123-167, 1989.

[**Hoofman 93**] Hoofman R. *Continuous information systems.* Information and Computation, **105**, #1, pp.42-71, 1993.

[**Larsen, Winskel 84**] Larsen K., Winskel G. *Using information systems to solve recursive domain equations effectively.* Lecture Notes in Computer Science, Vol.173, pp.109-129, Springer-Verlag, 1984.

[**Scott 76**] Scott D.S. *Data types at lattices.* SIAM Journal on Computing, **5**, #3, pp.522-587, 1976.

[**Scott 82**] Scott D.S. *Domains for denotational semantics.* Lecture Notes in Computer Science, Vol.140, pp.577-613, Springer-Verlag, 1982.

[**Shamir, Wadge 77**] Shamir A., Wadge W.W. *Data types as objects.* Lecture Notes in Computer Science, Vol.52, pp.465-479, Springer-Verlag, 1977.