Some topics in domain theory

Achim Jung
University of Birmingham, UK

9–11 June 2021

BLAST 2021 (online)

Motivation — or, why give this tutorial?

1. Domains were introduced by Dana Scott in 1969 to provide a general and intuitive model of computation. Despite the success of this approach (viz. the programming language Haskell), fundamental questions about the nature of computation remain open.
2. Domains exhibit intriguing interactions between order and topology.
3. Domains and domain-theoretic methods appear on both sides of Stone duality.
4. Domain theory presents a number of attractive open problems.

Motivation — or, why give this tutorial?

1. Domains were introduced by Dana Scott in 1969 to provide a general and intuitive model of computation. Despite the success of this approach (viz. the programming language Haskell), fundamental questions about the nature of computation remain open.
2. Domains exhibit intriguing interactions between order and topology.
3. Domains and domain-theoretic methods appear on both sides of Stone duality.
4. Domain theory presents a number of attractive open problems.

NB. The aim of these lectures is to provide an introduction and to tell a particular story from domain theory, not to give a comprehensive overview.

Part I: The origins in denotational semantics

I. Capturing computability: calculators
II. Capturing computability: algebra
III. Algebraic Scott domains
IV. Power domains
V. Summary

The origins

In 1900, David Hilbert asked for a general finite procedure to determine the solvability of Diophantine equations ("Hilbert's 10th problem").

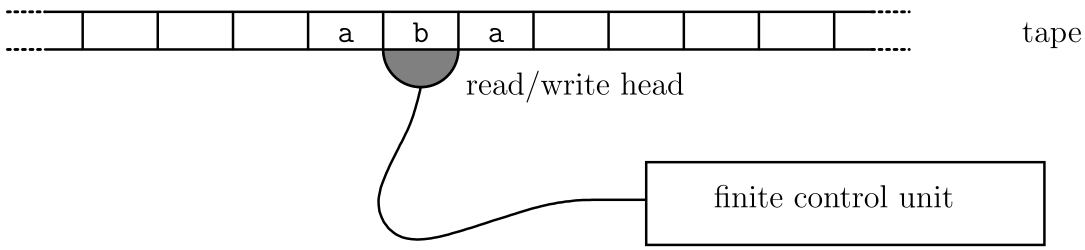
Around 1928 he and Wilhelm Ackermann asked for a general finite procedure to decide logical entailment in first-order logic ("Entscheidungsproblem").

The Entscheidungsproblem was tackled by Post, Herbrand, Gödel, Church, and Turing and shown to be unsolvable. To be able to make such a negative statement, they needed to give a convincing definition of what counts as a "general finite procedure" (and what does not).

There are now many proposals, but as far as Hilbert's problems are concerned, they are all equivalent ("Church Turing thesis").

Turing machines

Turing's approach is the most convincing, as it is modelled directly on the mathematician computing with pencil and paper.



Modern computer hardware is still quite close to this model of computing in the sense that writing to and reading from memory is a key feature.

Turing machine programs

But there is a problem: Turing machines are hard to understand:

δ	a	b	$\sqcup$
$q_0 = 0$	$(1, r)$	$(1, r)$	$(1, r)$
1	$(2, \sqcup)$	$(12, \sqcup)$	$(20, l)$
2	$(2, \sqcup)$	$(2, \sqcup)$	$(3, l)$
3	$(3, l)$	$(3, l)$	$(4, l)$
4	$(4, l)$	$(4, l)$	$(5, a)$
5	$(5, r)$	$(5, r)$	$(6, r)$
6	$(6, r)$	$(6, r)$	$(0, a)$
12	$(12, \sqcup)$	$(12, \sqcup)$	$(13, l)$
13	$(13, l)$	$(13, l)$	$(14, l)$
14	$(14, l)$	$(14, l)$	$(15, b)$
15	$(15, r)$	$(15, r)$	$(16, r)$
16	$(16, r)$	$(16, r)$	$(0, b)$
20	$(1, \sqcup)$	$(1, \sqcup)$	*stop*

Layers of abstraction

Of course, this problem is addressed in computer science and the solution is to bundle primitive operations together into more complex, “higher level” functional units. This results in “layers of abstraction”, ubiquitous in computing.

But now two new problems arise:

1. The higher-level functional units should be general and customizable, i.e., applicable in and adaptable to many situations.

2. At the same time, they should be precisely specified and comprehensible to the programmer.

Layers of abstraction

Of course, this problem is addressed in computer science and the solution is to bundle primitive operations together into more complex, "higher level" functional units. This results in "layers of abstraction", ubiquitous in computing.

But now two new problems arise:

1. The higher-level functional units should be general and customizable, i.e., applicable in and adaptable to many situations.

2. At the same time, they should be precisely specified and comprehensible to the programmer.

Much of computer science is concerned with finding the right balance between these two competing requirements.

Algebra of computable functions

We take certain mathematical operations as basic, such as incrementing a number by one.

Then we focus on the following processes, related to the first item on the last slide:

- combining "primitive" functions to "higher-level" ones;
- parameterizing functions to make them more general.

Computation will be done symbolically, exactly as we do in algebra, reducing complex expressions to combinations of the basic operations.

Using λ-calculus to define functions

Example:

$t := 3 + 4$	a term that evaluates to 7
$\varphi := \lambda x.x + 4$	a function that adds 4 to its argument
$\psi := \lambda xy.x + y$	a function that adds its two arguments
$\Gamma := \lambda fxy.f\,x\,y$	a functional that evaluates a function of two arguments at x and y

$$\begin{aligned} t &\longrightarrow 7 \\ \varphi(3) &\longrightarrow 7 \\ \psi(3,4) &\longrightarrow 7 \\ \Gamma(+,3,4) &\longrightarrow 7 \end{aligned}$$

Recursion

A key ingredient of an algebra of computable function is some form of iteration or recursion. This can be achieved by naming expressions.

Example:

```
add(x,y) := if (x=0) then y
                     else add(x-1,y+1)
```

Note that this is a circular definition, not normally considered meaningful in mathematics.

It *is* meaningful if we read the "equation" as a re-write rule from left to right:

$$\text{add}(3,5) \longrightarrow \text{add}(2,6) \longrightarrow \text{add}(1,7) \longrightarrow \text{add}(0,8) \longrightarrow 8$$

Fixpoints

Recursion is neatly (and fully) captured by a fixpoint combinator. First note that `add` is a fixpoint of the operation Φ

$$f \quad \mapsto \quad \lambda xy.\, \texttt{if}\ (x = 0)\ \texttt{then}\ y\ \texttt{else}\ f(x-1, y+1)$$

which takes an arbitrary function of two arguments (called f) and returns another function of two arguments.

Let Y be the operator ("combinator") which returns the fixpoint of such an operation, so

$$\texttt{add} := \mathsf{Y}(\Phi)$$

and the definition of `add` is no longer circular.

For our re-writing interpretation, the only rule we need for Y is

$$\mathsf{Y}(\Phi) \longrightarrow \Phi(\mathsf{Y}(\Phi))$$

Dana Scott's algebra LCF of computable functions

Names for types (objects):

$\sigma \quad ::= \quad \mathsf{bool} \mid \mathsf{nat} \mid \sigma \to \sigma \mid \sigma \times \sigma$

Names for terms (elements & morphisms):

$\mathsf{tt}, \mathsf{ff} : \mathsf{bool}$	$\mathsf{if}_\sigma : \mathsf{bool} \to \sigma \to \sigma \to \sigma$	
$0, 1, \ldots : \mathsf{nat}$	$\mathsf{succ}, \mathsf{pred} : \mathsf{nat} \to \mathsf{nat}$	$\mathsf{zero?} : \mathsf{nat} \to \mathsf{bool}$
$\mathsf{x}^\sigma, \mathsf{y}^\sigma, \ldots : \sigma$	$\lambda x^\sigma .\, M \qquad M N$	
$\mathsf{Y}_\sigma : (\sigma \to \sigma) \to \sigma$		

As an example, the operator Φ from the last slide has type

$(\mathsf{nat}^2 \to \mathsf{nat}) \to (\mathsf{nat}^2 \to \mathsf{nat})$ and so $\mathsf{add} = Y(\Phi)$ has type $\mathsf{nat}^2 \to \mathsf{nat}$.

How good is this language?

Theorem. *The (partial) functions of type* nat $\to$ nat *definable in LCF are precisely the Turing machine computable ones.*

How good is this language?

Theorem. *The (partial) functions of type* nat $\to$ nat *definable in LCF are precisely the Turing machine computable ones.*

This says that the language is expressive but we also want to know whether it is comprehensible to the programmer.

How good is this language?

Theorem. *The (partial) functions of type* nat $\rightarrow$ nat *definable in LCF are precisely the Turing machine computable ones.*

This says that the language is expressive but we also want to know whether it is comprehensible to the programmer.

> If every expression in LCF is a combination of the basic operations, is it also the case that the meaning of every expression can be seen as a combination of the meanings of the basic operations?
>
> In other words, does this language have a compositional semantics?

A compositional semantics for LCF

We need semantic spaces for every type.

$\sigma \quad ::= \quad \mathsf{bool} \mid \mathsf{nat} \mid \sigma \to \sigma \mid \sigma \times \sigma$

Naturally, we choose $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$ and $\mathbb{N}$ for the basic types, and the set of all functions and cartesian product for the two constructions. This works for everything except the fixpoint combinator Y.

In categorical language we say that the category **Set** is cartesian closed but it contains endomorphisms without fixpoints.

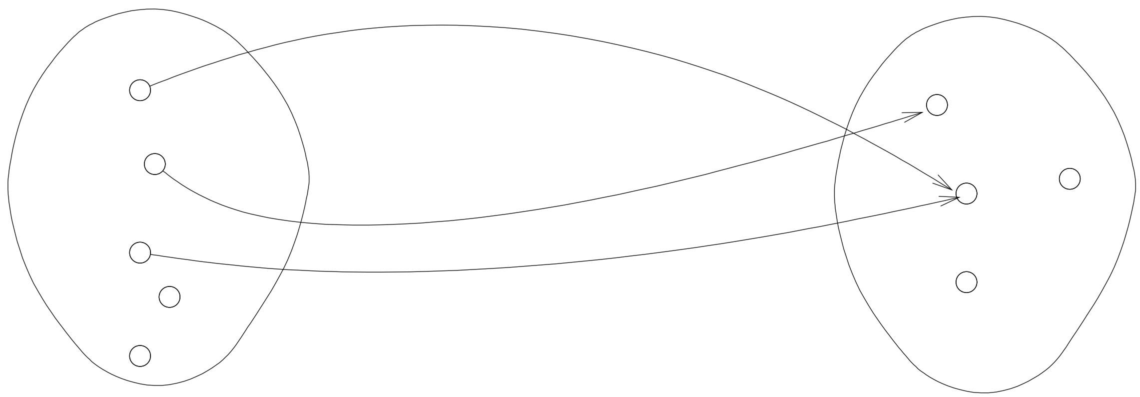
And indeed, we know that in order to capture computable functions we must allow for partial functions as well because recursion may lead to non-terminating expressions.

From partial functions to total monotone functions

Scott deals with partiality elegantly by *adding one extra element* to the mathematical sets $\mathbb{B}$ and $\mathbb{N}$:

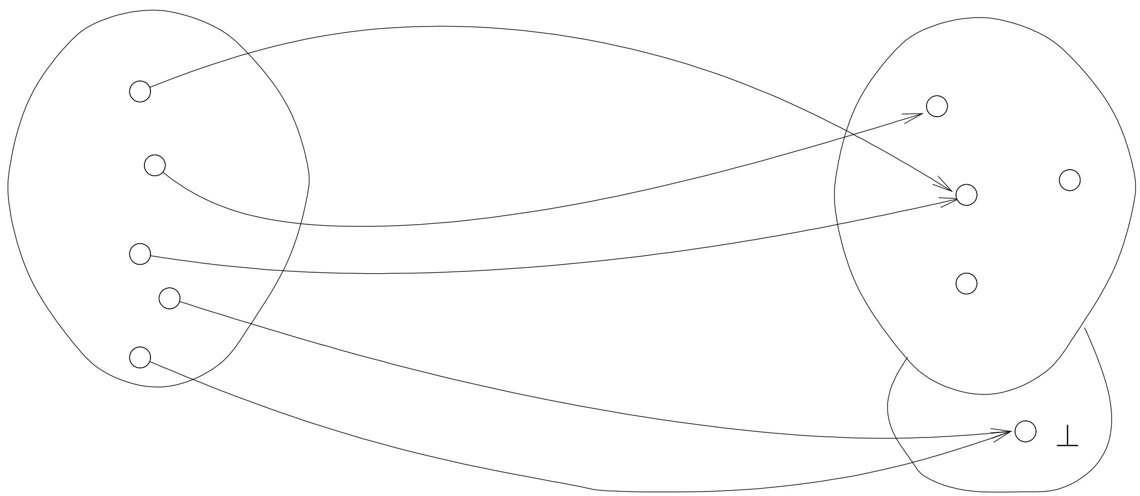
From partial functions to total monotone functions

Scott deals with partiality elegantly by *adding one extra element* to the mathematical sets $\mathbb{B}$ and $\mathbb{N}$:



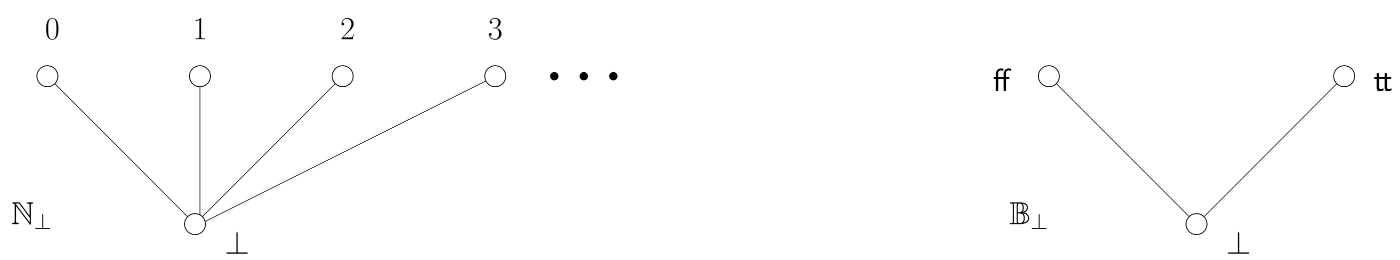
From partial functions to total monotone functions

Scott deals with partiality elegantly by *adding one extra element* to the mathematical sets $\mathbb{B}$ and $\mathbb{N}$:



What to do with $\bot$?

If all sets have a bottom element added, then this must also be mapped somewhere. Scott introduces an order relation in which $\bot$ is smaller than every other element:

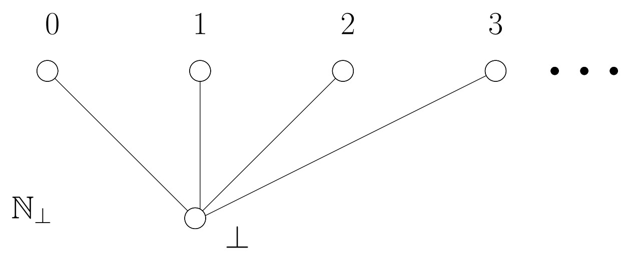


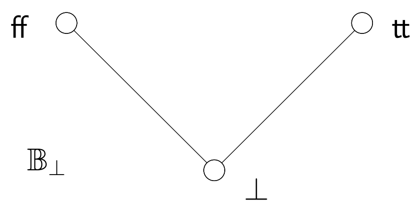
and stipulates that functions must preserve the order.

Observation. *The category* **POS** *of posets and monotone functions is cartesian closed.*

What to do with $\perp$?

If all sets have a bottom element added, then this must also be mapped somewhere. Scott introduces an order relation in which $\perp$ is smaller than every other element:





and stipulates that functions must preserve the order.

Observation. *The category* **POS** *of posets and monotone functions is cartesian closed.*

How about fixpoints?

More complicated orders

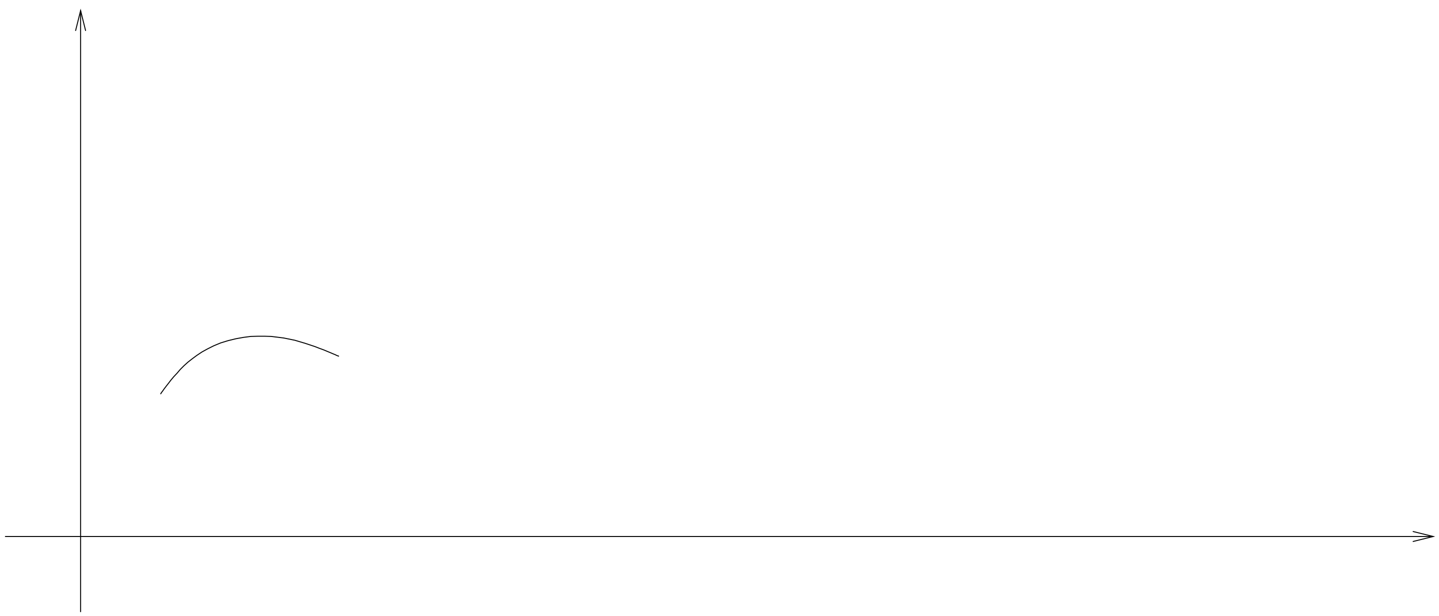
The semantics of nat $\to$ nat is the monotone function space $[\mathbb{N}_\perp \to \mathbb{N}_\perp]$. It "contains" all of $[\mathbb{N} \to \mathbb{N}]$ and $[\mathbb{N} \rightharpoonup \mathbb{N}]$, so in particular, has uncountably many elements.

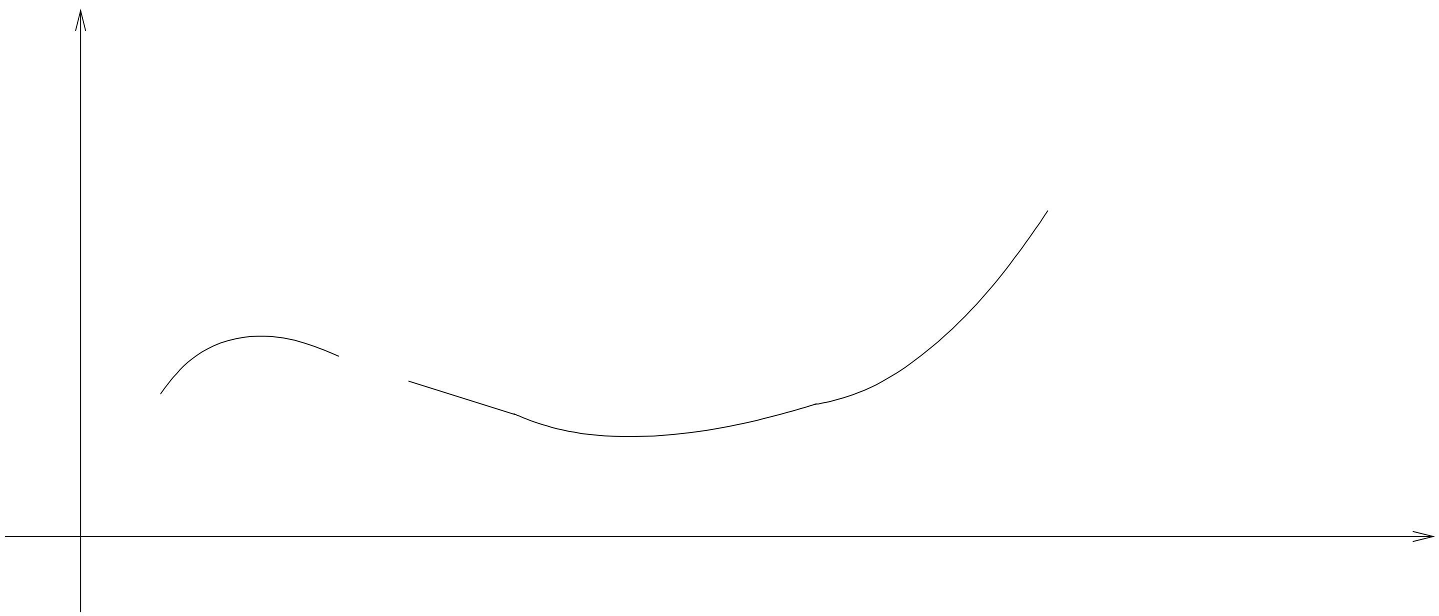
More complicated orders

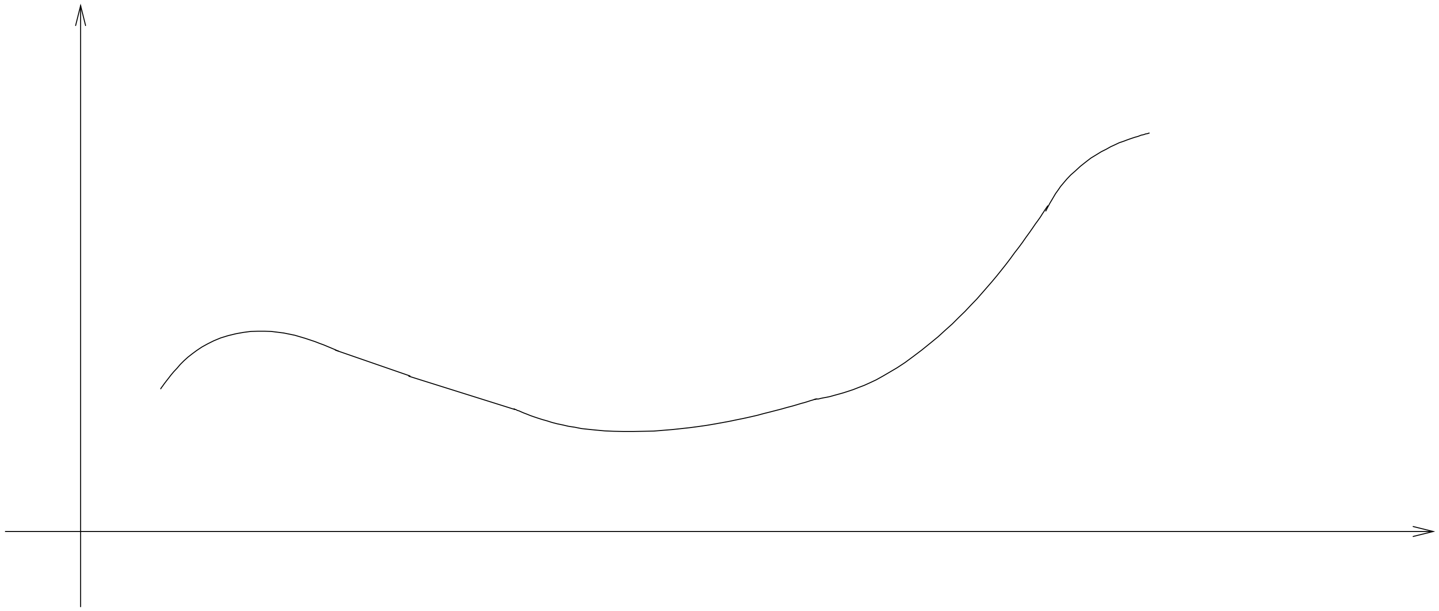
The semantics of $\mathsf{nat} \rightarrow \mathsf{nat}$ is the monotone function space $[\mathbb{N}_\perp \rightarrow \mathbb{N}_\perp]$. It "contains" all of $[\mathbb{N} \rightarrow \mathbb{N}]$ and $[\mathbb{N} \rightharpoonup \mathbb{N}]$, so in particular, has uncountably many elements.

It also contains infinite ascending chains:

$$\begin{array}{ccccccccccccccccc} 0 & \mapsto & 0 & \leq & 0 & \mapsto & 0 & \leq & 0 & \mapsto & 0 & \leq & 0 & \mapsto & 0 & \leq & \ldots \\ & & & & 1 & \mapsto & 1 & & 1 & \mapsto & 1 & & 1 & \mapsto & 1 & & \\ & & & & & & & & 2 & \mapsto & 4 & & 2 & \mapsto & 4 & & \\ & & & & & & & & & & & & 3 & \mapsto & 9 & & \end{array}$$







Scott's thesis

How big is $[[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to \mathbb{N}_\perp$?

Scott's thesis

How big is $[[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to \mathbb{N}_\perp$?

Scott observed that a terminating computation of type $(\mathsf{nat} \to \mathsf{nat}) \to \mathsf{nat}$ can query its argument only finitely often.

Thus only a finite part of the graph of the argument is needed.

This was known to recursion theorists as the Myhill-Shepardson Theorem.

Scott's thesis

How big is $[[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to \mathbb{N}_\perp$?

Scott observed that a terminating computation of type (nat $\to$ nat) $\to$ nat can query its argument only finitely often.

Thus only a finite part of the graph of the argument is needed.

This was known to recursion theorists as the Myhill-Shepardson Theorem.

Scott formulated and generalized this as follows:

> If f is computable and if the input is of the form $\bigvee^{\uparrow} x_i$ then the output $f(x)$ can be computed as $\bigvee^{\uparrow} f(x_i)$, or as a formula
>
> $f(\bigvee^{\uparrow} x_i) = \bigvee^{\uparrow} f(x_i)$

We say that functions should be Scott-continuous.

Scott domains

These considerations lead us to the category **S** of Scott domains which stands at the beginning of domain theory. We will define it formally in a moment, but to conclude this introduction to denotational semantics, let us write down that **S** can serve as a semantic universe for the language LCF:

Theorem. *The category* **S** *is cartesian closed. Every endomorphism in* **S** *has a (least) fixpoint, computed as*

$$\mathsf{lfp}(f) := \bigvee^{\uparrow}_{n\in\mathbb{N}} f^n(\bot)$$

Summary

1. The "algebraic" approach to programming uses names for basic computable functions and combinators to express arbitrary computable functions.

2. To make this meaningful, basic functions and the combinators need to be given an interpretation in a mathematical universe. Scott's category **S** is such a universe, in particular, it contains lfp as a meaningful interpretation of the recursion operator Y.

3. Many deep connections between LCF and **S** have been found.

Summary

1. The "algebraic" approach to programming uses names for basic computable functions and combinators to express arbitrary computable functions.

2. To make this meaningful, basic functions and the combinators need to be given an interpretation in a mathematical universe. Scott's category **S** is such a universe, in particular, it contains lfp as a meaningful interpretation of the recursion operator Y.

3. Many deep connections between LCF and **S** have been found.

4. However, this is not the final answer to the question "What is computation?"

Literature

D. S. Scott. *A type theoretic alternative to ISWIM, CUCH, OWHY.* Manuscript, University of Oxford, 1969. (Reprinted in *Theoretical Computer Science*, 1993, vol. 121, pp. 411–440.)

G. D. Plotkin. *LCF considered as a programming language.* *Theoretical Computer Science*, 5:223–255, 1977.

Th. Streicher. *Domain-Theoretic Foundations of Functional Programming* World Scientific, 2006.

A. Jung. *Teaching denotational semantics.* *ACM SIGLOG News*, 1(2):25–37, 2014.

Directed sets and dcpos

Definition. *A non-empty subset A of an ordered set D is called* *directed, if every pair of elements of A has an upper bound in A.*

Definition. *An ordered set D is called a directed-complete partial order or dcpo if every directed subset A of D has a supremum (which we write as $\bigvee^{\uparrow} A$).*

The dcpo is called pointed if it has a least element $\bot$.

We recall that suprema of directed sets (countable chains actually) were used to define the least fixpoint operator:

$$\mathsf{lfp}(f) := \bigvee^{\uparrow}{}_{n\in\mathbb{N}} f^n(\bot)$$

Examples

1. Every finite directed set has a largest element. Consequently, every finite poset is a dcpo.

2. Every (non-empty) totally ordered set ("chain") is directed.

3. The unit interval $[0, 1]$ is a pointed dcpo but $\mathbb{R}$ is neither pointed nor a dcpo. Every successor ordinal is a dcpo.

4. Every complete lattice is a dcpo.

5. The compact non-empty subsets of a Hausdorff topological space form a dcpo under reverse inclusion.

Finite elements

Definition. *An element* c *of a dcpo* D *is called* *compact* *or* *finite* *if whenever* $c \leq \bigvee^{\uparrow} A$ *for a directed set* $A \subseteq D$*, then* $c \in {\downarrow} A$.

Definition. *A dcpo is called* *algebraic*, *or an* *algebraic domain* *if every element is the directed sup of the compact elements below it.*

Examples

1. Every element of a finite poset is compact. Consequently, every finite poset is an algebraic domain.
2. The compact elements of a powerset lattice are exactly the finite sets. Powerset lattices are algebraic lattices.
3. The finitely generated subalgebras of any (universal) algebraic structure are the compact elements of the subalgebra lattice, which therefore is an algebraic lattice. Similarly for congruence lattices.
4. The set of ideals (the "ideal completion") of any poset is an algebraic domain.
5. Only 0 is compact in the unit interval. This is therefore not algebraic.
6. Only $\emptyset$ is compact in the open set lattice of $\mathbb{Q}$. This complete lattice is therefore not algebraic.

Scott domains

Definition. *A Scott domain is an algebraic domain in which*

1. *every bounded subset has a supremum;*
2. *there are at most countably many finite elements.*

Examples.

1. *The "flat domains"* $\mathbb{B}_\perp$ *and* $\mathbb{N}_\perp$*.*
2. *Algebraic lattices.*
3. *The partial functions* $[\mathbb{N} \rightharpoonup \mathbb{N}]$ *(ordered by graph inclusion).*

Categories of domains

Definition. *A monotone map* f *between dcpos is called* *Scott-continuous* *if it preserves suprema of directed sets:*

$$f(\bigvee^{\uparrow} A) = \bigvee^{\uparrow}{}_{a \in A} f(a)$$

NB. Even when the dcpos are pointed we do not usually require a Scott-continuous map to preserve the least element. Also, when we consider Scott-domains, we do not require the morphisms to preserve bounded suprema.

Using Scott-continuous maps as morphisms, we obtain the categories

1. **DCPO** of pointed dcpos;
2. **ALG** of pointed algebraic domains;
3. **S** of Scott-domains.

Categorical properties

1. **DCPO**, **ALG**, and **S** have finite products.

2. Endomorphisms in **DCPO**, **ALG**, and **S** have least fixpoints.

3. **DCPO** and **S** have exponentials, and are cartesian closed:

$$[A \times B \to C] \cong [A \to [B \to C]]$$

Categorical properties

1. **DCPO**, **ALG**, and **S** have finite products.

2. Endomorphisms in **DCPO**, **ALG**, and **S** have least fixpoints.

3. **DCPO** and **S** have exponentials, and are cartesian closed:

$$[A \times B \to C] \cong [A \to [B \to C]]$$

NB. **ALG** is not cartesian closed. For example, the function space $[\mathbb{Z}_- \to \mathbb{Z}_-]$ contains no compact elements at all.

Categorical properties

1. **DCPO**, **ALG**, and **S** have finite products.

2. Endomorphisms in **DCPO**, **ALG**, and **S** have least fixpoints.

3. **DCPO** and **S** have exponentials, and are cartesian closed:

$$[A \times B \to C] \cong [A \to [B \to C]]$$

   NB. **ALG** is not cartesian closed. For example, the function space $[\mathbb{Z}_{-} \to \mathbb{Z}_{-}]$ contains no compact elements at all.

4. There is an adjunction between **POS** (posets and monotone maps) and **ALG** given by ideal completion and (non-full) inclusion.

Going beyond pure functional languages

Mathematically interesting challenges arise if we want to extend Scott's approach to programming languages that allow for some choice, either nondeterministic or probabilistic:

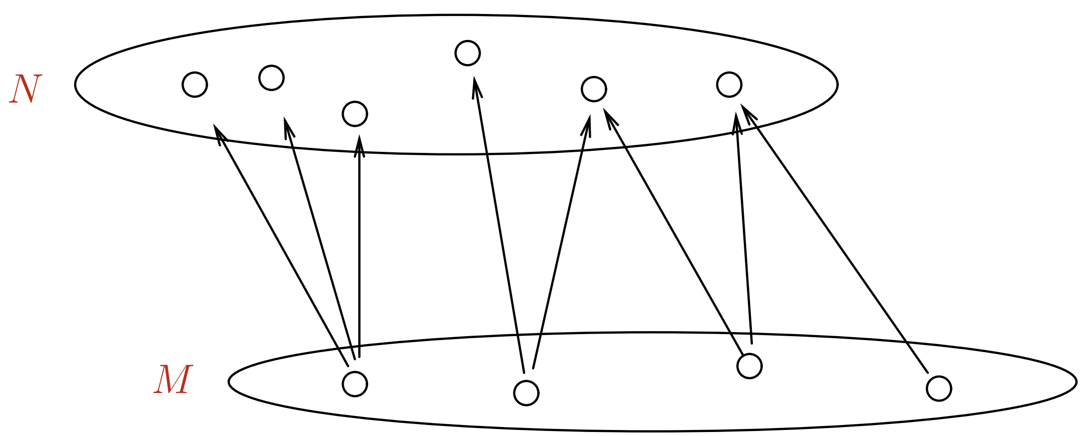
$$P \ \text{or} \ Q \qquad\qquad P \ \text{or}_p \ Q$$

To keep the Tarskian approach to semantics, we need an algebraic structure on domains that reflects these operators. Concretely, we need to consider sets of possible outcomes, or distributions over outcomes.

The Egli-Milner order

The most natural order between sets M and N of possible outcomes is given by

$$M \leq N \quad :\iff \quad \forall m \in M \exists n \in N.\, m \leq n \quad \text{and}$$
$$\forall n \in N \exists m \in M.\, m \leq n$$



The Plotkin powerdomain

Definition. *Given an algebraic domain* D, *define* $\mathcal{P}(\mathcal{D})$ *as the ideal completion of the finite (non-empty) powerset of the set of compact elements of* D, *equipped with the Egli-Milner ordering.*

This gives again an algebraic domain (since ideal completions always do) and it can be shown that it adds a semantic "choice" operation freely:

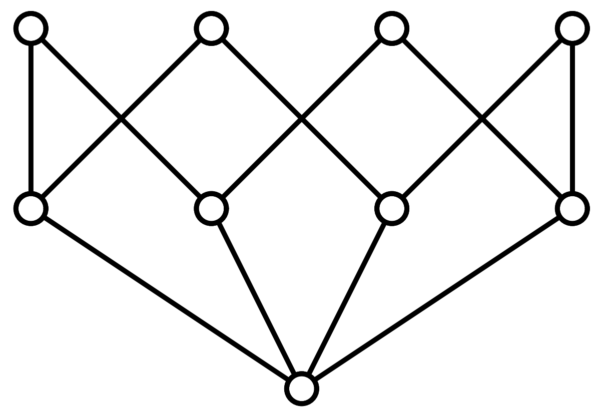
Theorem. *The Plotkin powerdomain of an algebraic domain is the free domain semilattice over that domain. In other words, it carries a Scott-continuous operation* $+$ *satisfying the equations*

1. $x + y = y + x$;
2. $x + (y + z) = (x + y) + z$;
3. $x + x = x$.

Beyond Scott domains

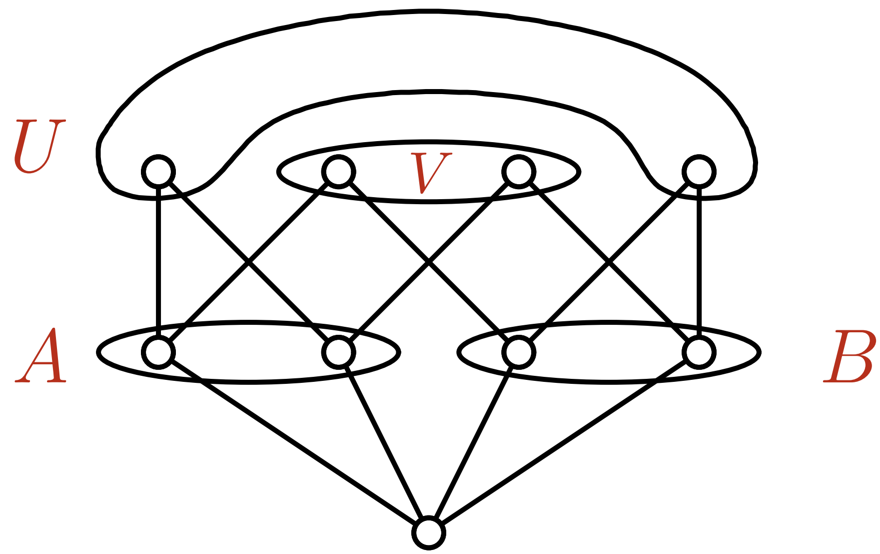
There is a problem with the Plotkin powerdomain construction: applied to a Scott domain we may not get a Scott domain again. Specifically, the resulting domain may not have suprema for all bounded subsets.

$\mathbb{B}_\perp \times \mathbb{B}_\perp$



Beyond Scott domains

There is a problem with the Plotkin powerdomain construction: applied to a Scott domain we may not get a Scott domain again. Specifically, the resulting domain may not have suprema for all bounded subsets.



We see that $A, B \leq_{EM} U, V$ but U and V are incomparable.

Bifinite domains

So we find that the Plotkin powerdomain works for **ALG** but **ALG** is not cartesian closed. On the other hand, **S** is cartesian closed but the Plotkin powerdomain construction is not closed on it.

The solution lies in the observation that the Plotkin powerdomain of a finite poset is again a finite poset. So we define bifinite domains as limits of ω-chains of finite posets, where the connecting morphisms have lower adjoints.

$$P_0 \longleftarrow P_1 \longleftarrow P_2 \cdots$$

We get a cartesian closed category **B** of algebraic domains that is also closed under the Plotkin powerdomain construction.

$$\mathbf{S} \subset \mathbf{B} \subset \mathbf{ALG} \subset \mathbf{DCPO}$$

The category **B** is canonical

Theorem. [Michael B. Smyth, 1983] **B** *is the largest cartesian closed category of algebraic domains which have at most countably many compact elements.*

A similar theorem was shown in 1988 for general algebraic domains, removing the size limitation.

I. Capturing computability: calculators

II. Capturing computability: algebra

III. Algebraic Scott domains

IV. Power domains

V. Summary

What have we achieved?

We have exhibited a category of structures, which

1. are familiar: not that different from algebraic lattices;
2. are useful: they do capture computable functions;
3. are remarkable: cartesian closed and fixpoints for endofunctions (and fixpoints for endofunctors);
4. accommodate free constructions in the sense of universal algebra;
5. have several more interesting properties, as we will see in the remainder of this tutorial.

Literature

B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 2nd edition, 2002.

S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Semantic Structures*, volume 3 of *Handbook of Logic in Computer Science*, pages 1–168. Clarendon Press, 1994.

G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003.

J. Goubault-Larrecq. *Non-Hausdorff Topology and Domain Theory*, volume 22 of *New Mathematical Monographs*. Cambridge University Press, 2013.

Part II: Continuous domains

I. Introduction

II. Scott topology

III. Categories of continuous domains

IV. Continuous domain algebras

V. Valuations

VI. Summary

Recap of Part I

1. Domains provide the semantic universe of mathematical objects which programs are "measured against". For example, the space of Scott-continuous functions $[\mathbb{N}_\perp \to \mathbb{N}_\perp]$ contains the computable first-order functions which are the subject of classical computability theory.
2. Scott identified the category **S** of Scott domains as a suitable semantic universe for higher-order functional programming, and showed how the paradigmatic language LCF (based on the λ-calculus) can be interpreted in **S**. Crucial ingredients for his compositional semantics are the following:
   (a) **S** is cartesian closed (ie, it has exponentials and products, and these are adjoint to each other);
   (b) every map $f\colon D \to D$ in **S** has a least fixpoint, calculated as $\bigvee^{\uparrow}_{n\in\mathbb{N}} f^n(\perp)$;
   (c) every object of **S** is algebraic and this allows us to analyse the relationship between language (LCF) and semantics (**S**) (but we did not have time to go into this).

3. Extending the language with a choice operator requires domains that carry a semantic analogue of this, ie, a binary operation. Plotkin showed that this is not possible in **S** and defined the category **B** of bifinite domains, inverse limits of finite posets. He also showed that **B** still has all the desirable properties listed above, and allows one to add a choice operator freely. His construction is called the Plotkin powerdomain, with connections to the Vietoris construction in topology and modal logic.
4. Smyth showed that **B** is the maximal cartesian closed subcategory consisting of algebraic domains.
5. In the hierarchy **aLat** $\subset$ **S** $\subset$ **B** $\subset$ **ALG** $\subset$ **DCPO**, only **S** and **B** are serious contenders for a "semantic universe".

I. Introduction

II. Scott topology

III. Categories of continuous domains

IV. Continuous domain algebras

V. Valuations

VI. Summary

Retracts

One can observe that while **DCPO** is closed under (Scott-continuous) retracts, **ALG** is not. We'll see some examples in a moment. On the positive side, we have the following:

Theorem. *If the category* **C** *is cartesian closed, then so is its Karoubi envelope* **rC**, *the category of (formal) retracts.*

This means that we don't have to worry about losing our interpretation of the λ-calculus part, and it suggests that the categories **raLat**, **rS**, **rB** and **rALG** should be of interest to us.

Example: The unit interval

We consider $P = [0, 1] \cap \mathbb{Q}$, the chain of rational numbers in the unit interval. We already know that its ideal completion is an algebraic Scott domain, whereas $[0, 1]$ is not.

We define a Scott-continuous retraction on the set of ideals:

$$r : I \mapsto \{a \in I \mid \exists b \in I.\, a < b\} \cup \{0\}$$

The image of r is the set of round ideals, ie, those ideals which do not have a largest element (unless the ideal consists of 0 alone).

The set of round ideals is easily seen to be isomorphic to $[0, 1]$.

Example: The unit interval

We consider $P = [0,1] \cap \mathbb{Q}$, the chain of rational numbers in the unit interval. We already know that its ideal completion is an algebraic Scott domain, whereas $[0,1]$ is not.

We define a Scott-continuous retraction on the set of ideals:

$$r : I \mapsto \{a \in I \mid \exists b \in I.\, a < b\} \cup \{0\}$$

The image of r is the set of round ideals, ie, those ideals which do not have a largest element (unless the ideal consists of 0 alone).

The set of round ideals is easily seen to be isomorphic to $[0,1]$.

We see that **rS** contains the unit interval (which is a nice space to have).

Continuous domains

It turns out that the retracts of algebraic domains can be characterised intrinsically. We begin with a slight generalisation of the notion of "compact element":

Definition. *For x, y elements of a dcpo D, say that x is way-below y (and write $x \ll y$) if whenever $y \leq \bigvee^{\uparrow} A$ for some directed set A, then $x \in {\downarrow} A$.*

We see that an element x of a dcpo is compact exactly if $x \ll x$.

Definition. *A dcpo D is called continuous, or a continuous domain if every element of D is the supremum of a directed family of elements way-below it.*

Equivalently, we could request that the set $\twoheaddownarrow y = \{x \in D \mid x \ll y\}$ be directed with supremum y.

CONT and ALG

Theorem. *Continuous domains are exactly the retracts of algebraic domains.*

Proof. [Sketch] Assume $e\colon D \leftrightarrows E\colon r$ is an embedding-retraction pair and E is algebraic. For $y \in D$ we know that $e(y)$ is the supremum of a directed family of compact elements in E. Now show that the images of these compact elements under r are way-below y in D and that y is their supremum.

Conversely, if D is continuous, then its ideal completion is algebraic and D is easily shown to be a retract of its ideal completion. The embedding is given by $x \mapsto \Downarrow x$. ■

So from now on we may write **CONT** instead of **rALG**.

Examples

1. In a finite poset we have $x \ll y$ iff $x \leq y$, so all finite posets are continuous domains.
2. All algebraic domains are also continuous (use the identity retraction).
3. The set of closed subintervals of $[0, 1]$, ordered by reverse inclusion is a continuous domain. We have $[a, b] \ll [a', b']$ iff $[a', b']$ is contained in the interior of $[a, b]$.
4. The lattice of open sets of a locally compact topological space.
5. The lattice of compact subsets of a locally compact Hausdorff space, ordered by reverse inclusion.

Enter topology

Definition. *A subset O of a dcpo D is called Scott-open if it is an upper set, $\uparrow O = O$, and if it is inaccessible by directed joints, $\bigvee^{\uparrow} A \in O \;\Rightarrow\; A \cap O \neq \emptyset$. The set of Scott-open subsets is denoted σ_D.*

Observation. *1. The Scott topology satisfies the T_0 separation axiom, but not T_1 or T_2.*

2. If D is an algebraic domain, then for every $y \in O$ there is a compact element $c \leq y$ contained in O. Likewise, if D is a continuous domain, then for every $y \in O$ there is an element $x \ll y$ contained in O.

3. if D is a continuous domain and O a Scott-open subset, then $O = \bigcup_{x \in O} \twoheaduparrow x$, and each $\twoheaduparrow x$ is itself a Scott-open set.

Theorem. *A function between dcpos is Scott-continuous if and only if it is continuous with respect to the Scott topologies.*

Enter topology

Definition. *A subset O of a dcpo D is called Scott-open if it is an upper set, $\uparrow O = O$, and if it is inaccessible by directed joints, $\bigvee^{\uparrow} A \in O \;\Rightarrow\; A \cap O \neq \emptyset$. The set of Scott-open subsets is denoted σ_D.*

Observation. *1. The Scott topology satisfies the T_0 separation axiom, but not T_1 or T_2.*

2. If D is an algebraic domain, then for every $y \in O$ there is a compact element $c \leq y$ contained in O. Likewise, if D is a continuous domain, then for every $y \in O$ there is an element $x \ll y$ contained in O.

3. if D is a continuous domain and O a Scott-open subset, then $O = \bigcup_{x \in O} \Uparrow x$, and each $\Uparrow x$ is itself a Scott-open set.

Theorem. *A function between dcpos is Scott-continuous if and only if it is continuous with respect to the Scott topologies.*

Corollary. *The categories* **rS** *and* **rB** *are cartesian closed full subcategories of* **TOP***. Ditto for the category of continuous lattices.*

More on the Scott topology

Theorem. *Continuous lattices equipped with the Scott topology are exactly the injective* T_0 *spaces.*

Continuous Scott domains equipped with the Scott topology are exactly the densely injective T_0 *spaces.*

Proof. [Sketch.] Let $Y \subseteq X$ be topological spaces and $f\colon Y \to D$ continuous where D is a continuous lattice with the Scott topology. Define $\bar{f}\colon X \to D$ by $\bar{f}(x) := \bigvee^{\uparrow}_{O \in \mathcal{N}(x)} \bigwedge\{f(y) \mid y \in O \cap Y\}$. The function $\bar{f}$ is always Scott continuous and the continuity of D ensures that it extends f.

Conversely, it is easy to see that Sierpiński space $\mathbf{2}$ is injective, and that products and retracts of injective spaces are again injective. Every T_0 space $(X, \Omega X)$ can be topologically embedded into the algebraic lattice $\mathbf{2}^{\Omega X}$, equipped with the Scott topology. If X is assumed to be injective, then this embedding has a right inverse, so X is a continuous retract of an algebraic lattice, ie, a continuous lattice. ■

I. Introduction

II. Scott topology

III. Categories of continuous domains

IV. Continuous domain algebras

V. Valuations

VI. Summary

Cartesian closed subcategories of **CONT**

We saw in the first lecture that the category of algebraic domains is **not** cartesian closed, so it is not surprising that **CONT** is neither.

On the other hand, we already mentioned that the Karoubi envelope construction preserves cartesian closure. Thus we have cartesian closed subcategories **cLat**, **rS** and **rB** of **CONT**.

Since **B** is the maximal ccc inside **ALG**, it is natural to conjecture that **rB** is the maximal ccc inside **CONT**.

Cartesian closed subcategories of CONT

We saw in the first lecture that the category of algebraic domains is **not** cartesian closed, so it is not surprising that **CONT** is neither.

On the other hand, we already mentioned that the Karoubi envelope construction preserves cartesian closure. Thus we have cartesian closed subcategories **cLat**, **rS** and **rB** of **CONT**.

Since **B** is the maximal ccc inside **ALG**, it is natural to conjecture that **rB** is the maximal ccc inside **CONT**.

This is a surprisingly hard problem which is as yet unsolved!

The classification of continuous domains

The open problem from the previous slide is even more astonishing, as we now know the largest ccc inside **CONT** (but we don't know whether it is the same as **rB**).

Definition. *Let D be a dcpo. A Scott-continuous function $f \leq \mathrm{id}_D$ is called*

- *an idempotent deflation if f has finite image and $f \circ f = f$;*
- *a deflation if f has finite image;*
- *finitely separated if there exists a finite set $M \subseteq D$ such that $\forall x \in D\, \exists m \in M.\ f(x) \leq m \leq x$.*

Definition. *A sequence of functions $f_n \colon D \to D$ is called an approximate identity if $\bigvee^{\uparrow}_{n \in \mathbb{N}} f_n = \mathrm{id}_D$.*

FS-domains

Theorem. *A dcpo is*

1. *bifinite, exactly if it carries an approximate identity consisting of idempotent deflations;*
2. *a retract of a bifinite domain, exactly if it carries an approximate identity consisting of deflations.*

Definition. *We call a dcpo an FS-domain if it carries an approximate identity consisting of finitely separated functions. We denote the corresponding category* **FS**.

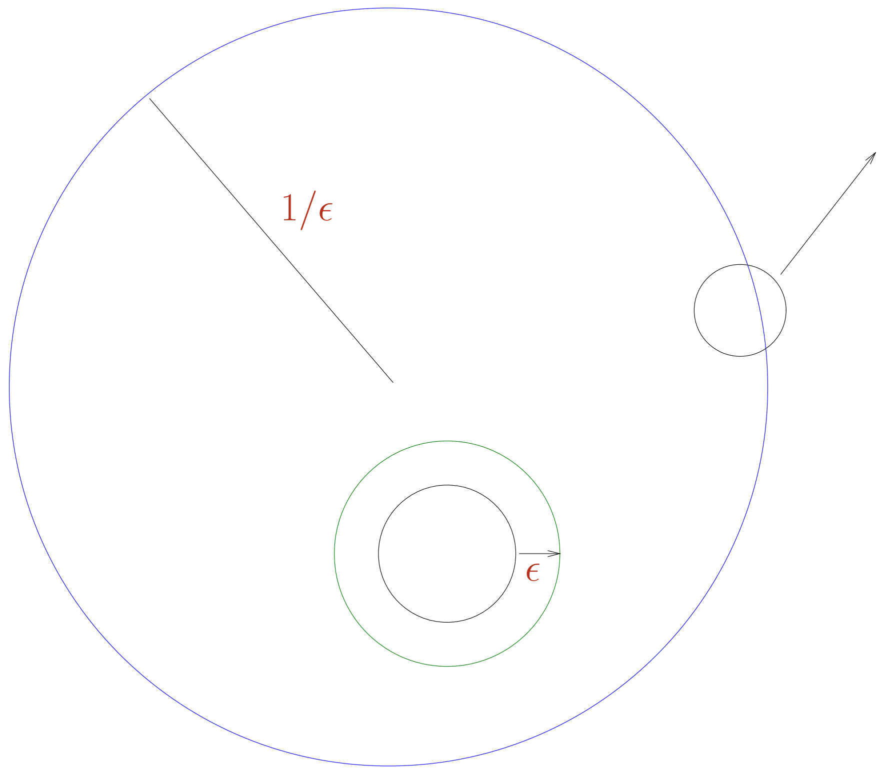
Clearly, **B** $\subset$ **rB** $\subset$ **FS**

Theorem. **FS** *is a maximal cartesian closed full subcategory of* **CONT**.

So, the open problem from above boils down to **rB** $\stackrel{?}{=}$ **FS**.

Jimmie Lawson's example

The set of closed discs in the plane, ordered by reversed inclusion.



This dcpo belongs to **FS** but we don't know whether or not it belongs to **rB**.

Another approach

Theorem. *If D is an rB-domain and E is an arbitrary continuous pointed domain then $[D \to E]$ is continuous.*

Whether the same is true for D being an FS-domain is not known.

We will meet yet another potential source for discriminating examples in a moment.

Fundamentally, the problem is that there is no characterisation of FS-domains other than the one given in the definition.

Abstract bases

Definition. *A set* B *carrying a binary relation* $\prec$ *is called an abstract basis if*

1. $\prec \circ \prec \;\subseteq\; \prec$ *(transitivity);*
2. $M \prec a \quad \Rightarrow \quad \exists b.\, M \prec b \prec a$ *(strong interpolation).*

A subset A *of an abstract basis* $(B;\prec)$ *is called a round ideal if*

1. $\{b \in B \mid b \prec a \in A\} \subseteq A$ *(downward closure);*
2. A *is directed wrt* $\prec$*.*

Proposition. *For* $(B;\prec)$ *an abstract basis, the set of round ideals is a continuous domain, and every continuous domain arises in this way.*

Free continuous domain algebras

Theorem. *Let D be a continuous domain, Σ a signature (in the sense of universal algebra), and $\mathcal{E}$ a set of (in)equations. Then the free continuous $(\Sigma, \mathcal{E})$-algebra exists.*

Proof. [Sketch.] Pick an abstract basis $(B; \prec)$ for D and construct the term algebra $T_\Sigma(B)$ with respect to Σ. On $T_\Sigma(B)$ consider the smallest transitive binary relation that contains $\prec$ extended to terms, and the order relation resulting from $\mathcal{E}$. Then show that an abstract basis is obtained. The round ideal completion is the desired free domain algebra. ∎

Note the use of *abstract bases* in the proof, analogous to the corresponding result for free algebras over algebraic domains.

However...

We have to remember that while **CONT** has many nice properties, we are really interested in cartesian closed categories of domains, and **CONT** is not.

This raises the question of when the free domain algebra of an rB-domain or an FS-domain is again in **rB** or **FS**, respectively.

Embarrassingly, no general answer to this question exists (in contrast to the situation for bifinite domains).

On the positive side, the Plotkin powerdomain construction can be shown to be closed on both **rB** and **FS**.

I. Introduction

II. Scott topology

III. Categories of continuous domains

IV. Continuous domain algebras

V. Valuations

VI. Summary

Valuations

Definition. *A function* v *from* σ_D *to* $[0,1]$ *is called a valuation if*

- $v(\emptyset) = 0$
- $v(\bigcup^{\uparrow} U_i) = \sup v(U_i)$ *(Scott continuity)*
- $v(U \cup V) = v(U) + v(V) - v(U \cap V)$ *(modularity)*

A probability valuation furthermore satisfies $v(D) = 1$*.*

Definition. *The set of all probability valuations on a dcpo* D*, ordered pointwise, is called the probabilistic powerdomain of* D*, and denoted by* $\mathcal{V}(D)$*.*

Fact. *If* D *is a continuous domain then the probability valuations on* D *are in one-to-one correspondence with probability Radon measures on the Borel algebra generated by the Scott-topology* σ_D*.*

Closure properties of $\mathcal{V}$

Proposition. *The probabilistic powerdomain of a dcpo is again a dcpo.*

Theorem. [Jones 1990] *The probabilistic powerdomain of a continuous domain is again a continuous domain.*

Definition. *A continuous domain is called coherent if it is stably compact as a topological space.*

Fact. *rB- and FS-domains are coherent.*

Theorem. [Jung & Tix 1998] *The probabilistic powerdomain of a coherent domain is again coherent.*

Theorem. [Jean Goubault-Larrecq 2021] *The probabilistic powerdomain of a quasi-continuous domain is again quasi-continuous.*

The Jung-Tix problem

No cartesian closed category of continuous domains is known for which the probabilistic powerdomain construction is closed. In particular:

> Is the probabilistic powerdomain of an FS-domain again an FS-domain?

What we know

Theorem. [Jung & Tix 1998] *The probabilistic powerdomain of a finite tree belongs to* **rB**.

Theorem. [Jung & Tix 1998] *The probabilistic powerdomain of a finite upside-down tree belongs to* **FS**.

What we know

Theorem. [Jung & Tix 1998] *The probabilistic powerdomain of a finite tree belongs to* **rB**.

Theorem. [Jung & Tix 1998] *The probabilistic powerdomain of a finite upside-down tree belongs to* **FS**.

So any domain arising as the probabilistic powerdomain of an upside-down finite tree is a candidate for showing $\mathbf{rB} \neq \mathbf{FS}$.

Another open problem

If the probabilistic powerdomain of every finite poset belongs to **FS**, does it follow that **FS** is closed under $\mathcal{V}$?

I. Introduction

II. Scott topology

III. Categories of continuous domains

IV. Continuous domain algebras

V. Valuations

VI. Summary

Summary

1. Continuous domains are a natural generalisation of algebraic ones and share many of the nice properties of the latter.

2. It is continuous domains that appear naturally in other mathematical contexts.

3. Many questions of interest are in fact unsolved for continuous domains. At the heart of these is our lack of understanding of how to exploit the rB- and FS-conditions.

Bibliography

S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Semantic Structures*, volume 3 of *Handbook of Logic in Computer Science*, pages 1–168. Clarendon Press, 1994

A. Jung. The classification of continuous domains. In *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 35–40. IEEE Computer Society Press, 1990

G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003

A. Jung and R. Tix. The troublesome probabilistic powerdomain. In A. Edalat, A. Jung, K. Keimel, and M. Kwiatkowska, editors, *Proceedings of the Third Workshop on Computation and Approximation*, volume 13 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers B.V., 1998. 23 pages

X. Jia, A. Jung, H. Kou, Q. Li, and H. Zhao. All cartesian closed categories of quasicontinuous domains consist of domains. *Theoretical Computer Science*, 594:143–150, 2015

Part III: Duality

Recap of Part II

1. Retracts of algebraic domains give us continuous domains, a wider class and one with more pleasing closure properties.
2. Among the domains now available, we have the continuous Scott domain $[\mathbf{I} \to \mathbf{I}]$, the function space of the interval domain. This plays the central role in Martín Escardó's work on *real-number computation* with results paralleling those of Scott and Plotkin for computation on natural numbers.
3. We also have the continuous domain $[\mathcal{V}(\mathbb{N}_\perp) \to \mathcal{V}(\mathbb{N}_\perp)]$ which seems the natural setting to study *machine learning* algorithms. This is an active area of investigation.
4. We have the hierarchy $\textbf{cLat} \subset \textbf{cS} \subset \textbf{rB} \subset \textbf{FS} \subset \textbf{CONT}$.
5. We saw two of the most prominent open problems in this area: $\textbf{rB} \stackrel{?}{=} \textbf{FS}$ and $\mathcal{V}(\textbf{FS}) \stackrel{?}{\subset} \textbf{FS}$.

I. Stone duality for algebraic domains

II. Domain theory in logical form

III. Stone duality for continuous domains

IV. DTLF for continuous domains?

V. Final thoughts

Stone 1937

M. H. Stone. Topological representation of distributive lattices and Brouwerian logics. *Časopis pro pěstování matematiky a fysiky*, 67:1–25, 1937/38.

Definition. *Let L be a bounded distributive lattice. A subset A of L is called a prime filter if it is*

1. *nonempty:* $1 \in A$;
2. *an upper set:* $\uparrow A = A$;
3. *filtered ("downward directed"):* $\forall a, b \in A\, \exists c \in A.\, c \leq a, b$;
4. *inaccessible by finite joins:* $\bigvee M \in A \;\; \Rightarrow \;\; M \cap A \neq \emptyset$ *for all finite* $M \subseteq L$.

We call the set of all prime filters the spectrum of L, *denoted* $\mathrm{spec}(L)$.

Order and topology on the spectrum

Proposition. *A directed union of prime filters is again a prime filter.*

Corollary. *Equipped with the inclusion order,* $\mathrm{spec}(L)$ *is a dcpo.*

Order and topology on the spectrum

Proposition. *A directed union of prime filters is again a prime filter.*

Corollary. *Equipped with the inclusion order,* $\mathsf{spec}(L)$ *is a dcpo.*

Proposition. *Equipped with the usual spectral topology, generated by the sets* $\Phi(x) := \{A \in \mathsf{spec}(L) \mid x \in A\}$, *the spectrum is a spectral space, to wit, it is*

1. *compact;*
2. *locally compact;*
3. *stably compact: finite intersections of compact upper sets are compact;*
4. *well-filtered:* $\bigcap_{\downarrow} K_i \subseteq O \quad \Rightarrow \quad \exists i_0 .\, K_{i_0} \subseteq O$ *for filtered families of compact upper sets and an open set* O*;*
5. *zero-dimensional: there is a basis of compact open subsets.*

Furthermore, the topology is contained in the Scott topology derived from the order.

Spectral domains

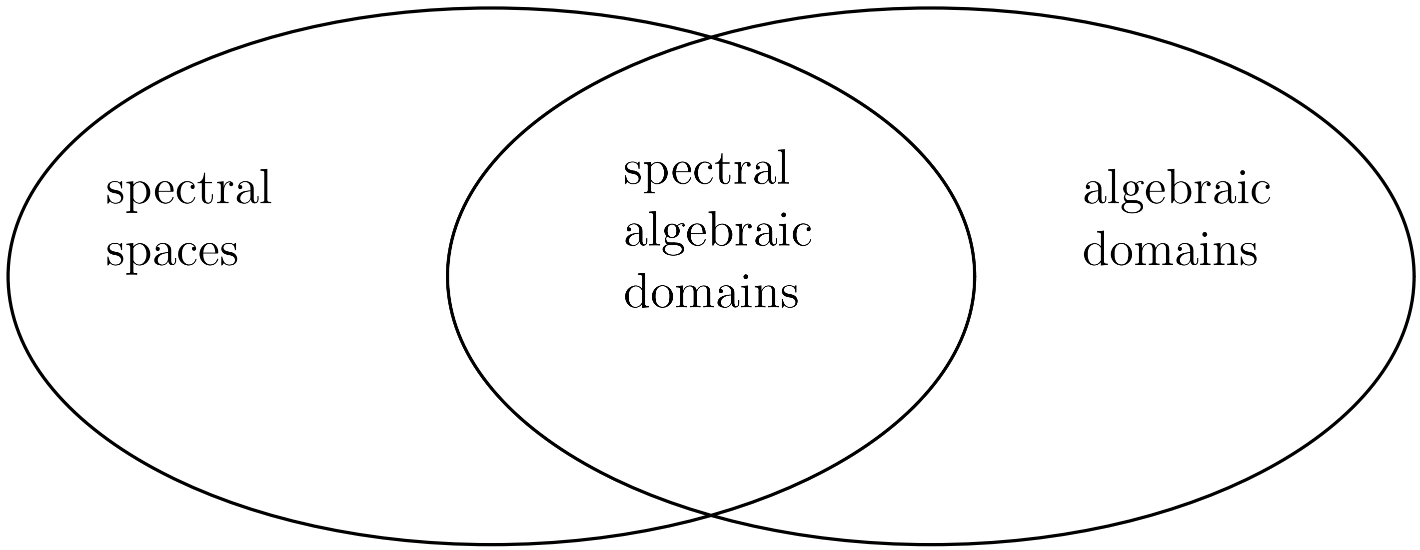
The basic opens $\Phi(x) := \{A \in \text{spec}(L) \mid x \in A\}$ arising from Stone duality are always compact (a consequence of the prime filter theorem).

Among dcpos, algebraic domains are such that there is an abundance of compact open subsets, namely, all sets of the form $\uparrow M$ with M a finite set of compact elements. In general, however, the set of compact open sets may fail to be closed under intersection. An algebraic domain is called spectral exactly if the intersection of compact open sets is again compact.

Returning to the other side again, we see that the compact open set lattices of algebraic domains are special in that each such set is the finite join of principal upsets $\uparrow c$ where c is compact. Such sets are clearly join prime. So we see that we get a lattice in which every element is the join of finitely many join primes.

Theorem. *Stone's duality between bounded distributive lattices and spectral spaces restricts to a duality between join-prime generated lattices and spectral algebraic domains.*

Domain theory meets Stone duality



Spectral domains

A pleasing fact:

Theorem. *Algebraic lattices, Scott domains, and bifinite domains are all spectral.*

Proposition. *The spectral topology equals the Scott topology for spectral algebraic domains.*

As an aside, there does exist a cartesian closed category of algebraic domains whose objects are not spectral spaces.

I. Stone duality for algebraic domains

II. Domain theory in logical form

III. Stone duality for continuous domains

IV. DTLF for continuous domains?

V. Final thoughts

Abramsky's programme

S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

The connection given by Stone duality suggests that we can study domains via lattices.

Abramsky's programme

S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

The connection given by Stone duality suggests that we can study domains via lattices.

While mathematically intriguing, we would also hope for some "payback" for our enterprise of analysing computability.

Abramsky's programme

S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

The connection given by Stone duality suggests that we can study domains via lattices.

While mathematically intriguing, we would also hope for some "payback" for our enterprise of analysing computability.

The key insight (first put forward by Mike Smyth in 1983) is that topologically open sets can be coded as continuous maps into $\mathbf{2}$ and hence as "semi-decidable properties" (exploiting the idea that continuity is an approximation of computability).

Example

We look at $D = [\mathbb{N} \rightharpoonup \mathbb{N}]$, the archetypical semantic domain.

The set of partial functions that are defined for input $x = 5$ form a Scott-open subset (remember, the order is graph extension).

Example

We look at $D = [\mathbb{N} \rightharpoonup \mathbb{N}]$, the archetypical semantic domain.

The set of partial functions that are defined for input $x = 5$ form a Scott-open subset (remember, the order is graph extension).

Whether a program belongs to this open set is semi-decidable: We run the program on input $x = 5$ and wait to see whether it terminates. If it does, the result of the test is affirmative. If it doesn't, we may never find out, but this is the essence of semi-decidability.

Different approaches to semantics

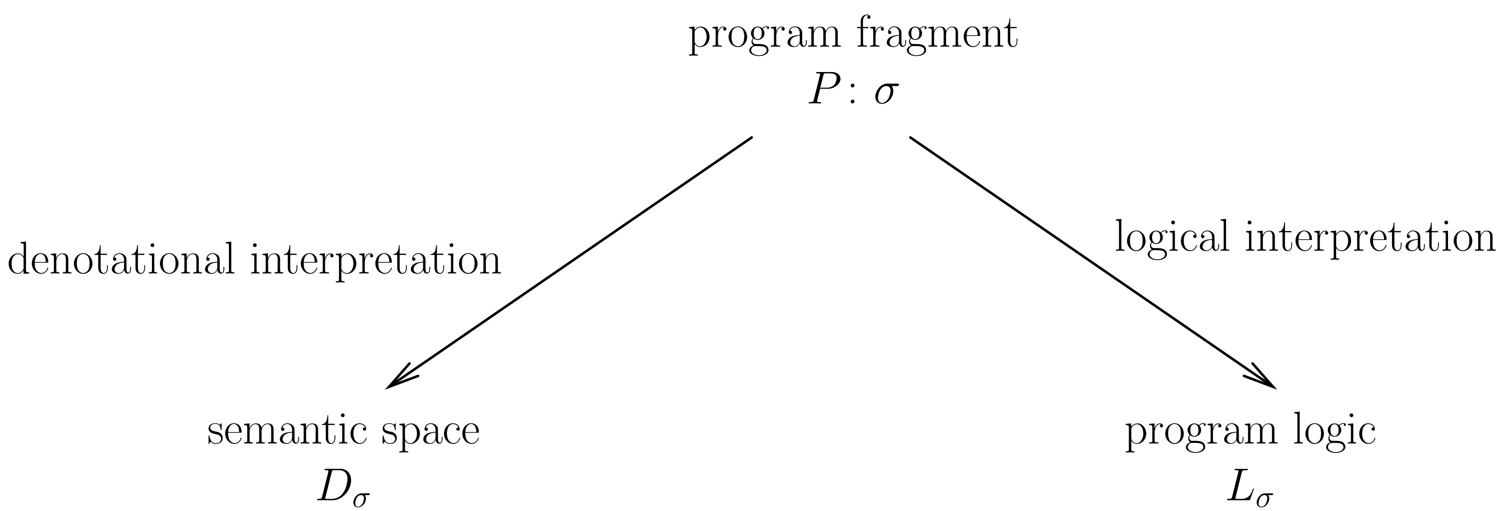
P : a program (fragment)

1. operational: How to evaluate P?

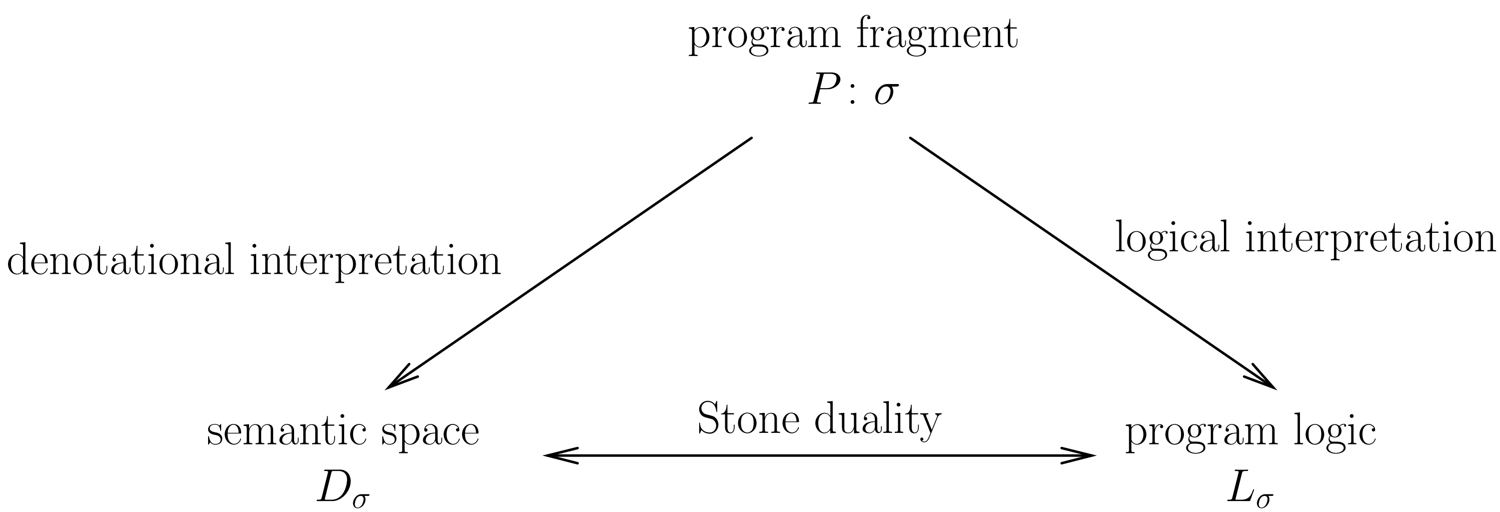
2. denotational: What (mathematical entity) does P denote?

3. logical: What properties does P have?

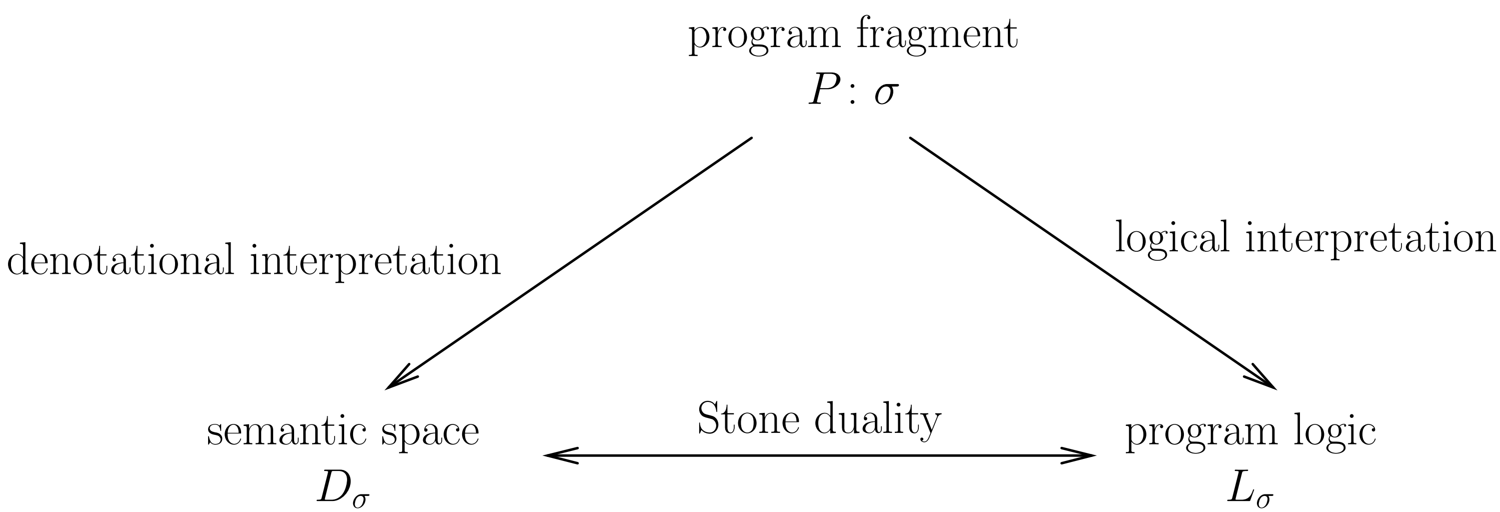
The semantic triangle



The semantic triangle



The semantic triangle



So if you are an algebraist or topologist, you take the right-hand side as the lattice of compact open subsets, while if you are a logician, you take it to be a positive propositional theory.

Capturing equality

For any distributive lattice L we obviously have for prime filters F and G:

$$F = G \quad \text{iff} \quad (\forall a \in L.\, a \in F \;\Leftrightarrow\; a \in G)$$

Under Stone duality, this turns to the following for any two elements x and y of a spectral space $(X;\tau)$

$$x = y \quad \text{iff} \quad (\forall O \in \tau.\, x \in O \;\Leftrightarrow\; y \in O)$$

Reading open sets as propositions, this becomes Leibniz equality:

$$x = y \quad \text{iff} \quad (\forall \varphi.\, x \models \varphi \;\Leftrightarrow\; y \models \varphi)$$

Samson Abramsky adds to this:

> Two programs are considered equal if they satisfy the same effective tests.

Constructing domains logically

We start with the one-point domain $\mathbf{1}$ for which the Stone dual is the two-element lattice and the corresponding propositional theory is all positive propositional expressions formed from true and false.

More elaborate domains (including $\mathbb{B}_\perp$ and $\mathbb{N}_\perp$) can be built from $\mathbf{1}$ by the "type constructors" $\times$, $\perp$, $\rightarrow$, $\mathcal{P}$, $\oplus$, and bilim.

$$\begin{array}{ccccc}
\textbf{B} & & \textbf{dLat} & & \textbf{PL} \\
D & \longleftrightarrow & L & \longleftrightarrow & \mathcal{L} \\
\downarrow & & \downarrow & & \downarrow \\
\nabla D & \longleftrightarrow & \nabla L & \longleftrightarrow & \nabla \mathcal{L}
\end{array}$$

Example: Plotkin powerdomain

From a positive propositional theory $\mathcal{L}$ construct its *Plotkin power theory* $\mathcal{P}(\mathcal{L})$ by

generators $\{\Box\varphi \mid \varphi \in \mathcal{L}\} \cup \{\Diamond\varphi \mid \varphi \in \mathcal{L}\}$

axioms

$$\begin{array}{rcl@{\qquad}rcl}
\mathrm{tt} & \leftrightarrow & \Box\, \mathrm{tt} & \mathrm{ff} & \leftrightarrow & \Diamond\, \mathrm{ff} \\
\Box(\varphi \wedge \psi) & \leftrightarrow & \Box\,\varphi \wedge \Box\,\psi & \Diamond(\varphi \vee \psi) & \leftrightarrow & \Diamond\,\varphi \vee \Diamond\,\psi \\
\Box(\varphi \vee \psi) & \rightarrow & \Box\,\varphi \vee \Diamond\,\psi & \Box\varphi \wedge \Diamond\psi & \rightarrow & \Diamond(\varphi \wedge \psi)
\end{array}$$

rules $$\frac{\varphi \rightarrow \psi}{\Box\,\varphi \rightarrow \Box\,\psi} \qquad\qquad \frac{\varphi \rightarrow \psi}{\Diamond\,\varphi \rightarrow \Diamond\,\psi}$$

Example: Plotkin powerdomain

Theorem. *If D is the Stone dual of the theory $\mathcal{L}$, then $\mathcal{P}(\mathcal{D})$ is the Stone dual of the theory $\mathcal{P}(\mathcal{L})$.*

Example: Plotkin powerdomain

Theorem. *If D is the Stone dual of the theory $\mathcal{L}$, then $\mathcal{P}(\mathcal{D})$ is the Stone dual of the theory $\mathcal{P}(\mathcal{L})$.*

So now we have three characterisations of the Plotkin powerdomain:

1. the ideal completion of the finite powerset of the set of compact elements;
2. the free algebraic theory of semilattices;
3. a modal logic like propositional theory.

The difficulties

1. The category of spectral algebraic domains is not closed under the function space construction, so not cartesian closed. We identified previously the category **B** of bifinite domains as a cartesian closed category also closed under the Plotkin powerdomain construction. The duality must be restricted to **B** and a suitable category of theories/lattices.

The difficulties

1. The category of spectral algebraic domains is not closed under the function space construction, so not cartesian closed. We identified previously the category **B** of bifinite domains as a cartesian closed category also closed under the Plotkin powerdomain construction. The duality must be restricted to **B** and a suitable category of theories/lattices.

2. Stone's duality links lattice homomorphisms with "proper" continuous maps between the spectra. Scott-continuous maps are not proper.

The difficulties

1. The category of spectral algebraic domains is not closed under the function space construction, so not cartesian closed. We identified previously the category **B** of bifinite domains as a cartesian closed category also closed under the Plotkin powerdomain construction. The duality must be restricted to **B** and a suitable category of theories/lattices.
2. Stone's duality links lattice homomorphisms with "proper" continuous maps between the spectra. Scott-continuous maps are not proper.
3. The function space construction is crucial for semantics, yet this is the one that is most unsatisfactory in Domain Theory in Logical Form as it requires an encoding of the domains inside the propositional theories.

The difficulties

1. The category of spectral algebraic domains is not closed under the function space construction, so not cartesian closed. We identified previously the category **B** of bifinite domains as a cartesian closed category also closed under the Plotkin powerdomain construction. The duality must be restricted to **B** and a suitable category of theories/lattices.
2. Stone's duality links lattice homomorphisms with "proper" continuous maps between the spectra. Scott-continuous maps are not proper.
3. The function space construction is crucial for semantics, yet this is the one that is most unsatisfactory in Domain Theory in Logical Form as it requires an encoding of the domains inside the propositional theories.
4. What about continuous domains?

I. Stone duality for algebraic domains

II. Domain theory in logical form

III. Stone duality for continuous domains

IV. DTLF for continuous domains?

V. Final thoughts

Frames

Spectral spaces are zero-dimensional but continuous domains are not. We move to frame duality to capture them.

On the side of the algebras, we use frames, complete lattices satisfying the frame distributivity law:

$$b \wedge \bigvee A = \bigvee_{a \in A} b \wedge a$$

Frame homomorphisms preserve finite meets and arbitrary joins.

Prime filters are replaced with completely prime filters, which in addition to being prime, are also Scott-open. Alternatively, they are filters inaccessible by arbitrary joins.

NB. There is no “completely prime filter theorem”!

Frame duality

Theorem. *There is a dual adjunction between the category of frames and frame homomorphisms, and the category of topological spaces and continuous maps. It assigns to a topological space* $(X;\tau)$ *the frame* $\Omega(X) := (\tau; \bigcup, \cap)$, *and to a frame* L *the set* $\mathsf{spec}(L)$ *of completely prime filters topologized with opens* $\Phi(a) := \{F \in \mathsf{spec}(L) \mid a \in F\}$, $a \in L$.

Definition. *A frame is called spatial if it is isomorphic to* $\Omega(X)$ *for some topological space* $(X;\tau)$.

A topological space is called sober if it is homeomorphic to $\mathsf{spec}(L)$ *for some frame* L.

Theorem. *The dual adjunction between frames and spaces restricts to a duality between spatial frames and sober spaces.*

Domain continuity in frame duality

As with Stone duality for distributive lattices, we easily get that sober spaces carry a dcpo-order and that the spectral topology is contained in the Scott topology with respect to that order.

Theorem. *Continuous domains with their Scott topology are sober spaces. Continuous distributive lattices are spatial frames.*

Domain continuity in frame duality

As with Stone duality for distributive lattices, we easily get that sober spaces carry a dcpo-order and that the spectral topology is contained in the Scott topology with respect to that order.

Theorem. *Continuous domains with their Scott topology are sober spaces. Continuous distributive lattices are spatial frames.*

Even better:

Theorem. [Jimmie Lawson 1979, Rudolf-Eberhard Hoffmann 1981] *The frame duals of continuous domains with their Scott topology are precisely the completely distributive lattices.*

Proof. [Sketch.] Both directions make use of George Raney's characterisation of completely distributive lattices, which says that a complete lattice is completely distributive if every element is the supremum of elements "far below" it, where $x \lll y$ iff $y \leq \bigvee A \;\;\Rightarrow\;\; x \in {\downarrow}A$.

If D is a continuous domain then every open set O is the union of sets of the form $\twoheaduparrow a$ for $a \in O$. Clearly, $\twoheaduparrow a \lll O$ in the frame σ_D.

If L is a completely distributive lattice then one can show that there is an abundance of completely prime filters of the form $\bigcup_{n\in\mathbb{N}} {\uparrow} a_n$ where $\ldots \lll a_2 \lll a_1 \lll a_0$: If F is completely prime and $a \in F$, then there is $b \in F$ with $b \lll a$. By interpolation one gets a chain $b \lll \ldots a_2 \lll a_1 \lll a_0 = a$ and the filter $\bigcup_{n\in\mathbb{N}} {\uparrow} a_n$ is clearly way-below F in $\mathsf{spec}(L)$.

This also shows that if O is Scott-open and $F \in O$ in $\mathsf{spec}(L)$ then there is some $\bigcup_{n\in\mathbb{N}} {\uparrow} a_n \subseteq {\uparrow} b \subseteq F$ which also belongs to O. We get that $F \in \Phi(b) \subseteq O$ which shows that the Scott topology is contained in the spectral topology. ■

The Hofmann-Mislove theorem

Theorem. [Karl Heinrich Hofmann & Michael Mislove 1981] *For* $(X;\tau)$ *a sober topological space there is a bijection between Scott-open filters of the lattice* τ *and the compact upper sets of* X*.*

Proof. [Sketch.] One immediately sees that the neighbourhood filter of a compact set is Scott-open.

For the converse one shows — using the Axiom of Choice — that a Scott-open filter in a frame is the intersection of the completely prime filters it is contained in. ∎

I. Stone duality for algebraic domains

II. Domain theory in logical form

III. Stone duality for continuous domains

IV. DTLF for continuous domains?

V. Final thoughts

Using frame duality

We see that frame duality for continuous domains is particularly pleasing. In addition, we have

1. a characterisation of the frame of opens of the probabilistic powerdomain (given by Reinhold Heckmann in 1994);
2. a description of the frame of opens for the function space construction (given by Martin Hyland in 1981);
3. many years of experience of working with the infinitary ("geometric") logic of frames through the work of Peter Johnstone, Steve Vickers, and others.

On the other hand...

1. geometric logic is infinitary, so quite likely less useful to a computer scientist;

2. we don't have a frame-theoretic characterisation of FS-domains, the canonical candidates for continuous semantic domains;

3. we don't know whether the probabilistic powerdomain construction is closed on **FS**, so cannot be sure we stay within exponentiable frames.

(A good starting point for further research on these problems is the proof that **FS** is cartesian closed and to attempt a translation into geometric logic.)

Returning to Stone duality

We remember Stone's characterisation of spectral spaces, to with, spaces which are

1. compact;
2. locally compact;
3. stably compact: finite intersections of compact upper sets are compact;
4. well-filtered: $\bigcap_{\downarrow} K_i \subseteq O \quad \Rightarrow \quad \exists i_0. K_{i_0} \subseteq O$ for filtered families of compact upper sets and an open set O;
5. zero-dimensional: there is a basis of compact open subsets.

All of these properties agree with the objects in our target category **FS**, except for the last one. The first four properties define precisely stably compact spaces. One can show:

Theorem. *FS-domains equipped with the Scott topology are stably compact.*

Stably compact spaces are well-behaved wrt $\mathcal{V}$

Theorem. [Jung 2004] *The category* **SCS** *of stably compact spaces is closed under the probabilistic powerspace construction.*

For this to make sense, we need to define a topology on the set of valuations (before we focused on the order). Subbasic opens are $\{\nu \in \mathcal{V}(X) \mid \nu(O) > r\}$ for O an open set of the space $(X;\tau)$ and $r \in [0,1]$.

A finitistic Stone dual for stably compact spaces

Definition. *A strong proximity lattice is a distributive lattice* $(L; \wedge, \vee, 1, 0)$ *equipped with a binary relation* $\prec$ *which satisfies the logical axioms*

$$\begin{array}{lcll} (\prec\!-\!1) & x \prec 1 & & \\ (0\!-\!\prec) & 0 \prec x & & \\ (\prec\!-\!\wedge) & x \prec y,\; x \prec y' & \Longleftrightarrow & x \prec y \wedge y' \\ (\vee\!-\!\prec) & x \prec y,\; x' \prec y & \Longleftrightarrow & x \vee x' \prec y \end{array}$$

and the strong interpolation axioms

$$\begin{array}{lcll} (\wedge\!-\!\prec) & a \wedge x \prec y & \Longrightarrow & \exists a' \in X.\; a \prec a' \;\textit{ and }\; a' \wedge x \prec y \\ (\prec\!-\!\vee) & x \prec y \vee a & \Longrightarrow & \exists a' \in X.\; a' \prec a \;\textit{ and }\; x \prec y \vee a' \end{array}$$

(Note that choosing $\leq$ for $\prec$ turns any distributive lattice into a strong proximity lattice.)

Duality

Theorem. [Jung & Sünderhauf 1995] *The Stone duals of strong proximity lattices are precisely the stably compact spaces.*

Lattices to spaces: set of round prime filters, $F = \uparrow F$

Spaces to lattices: pairs of an open and a compact upper set $O \subseteq K$ *with the operations*

$$\begin{array}{rclr}
(U,K) \vee (U',K') & := & (U \cup U', K \cup K') & 0 := (\emptyset,\emptyset) \\
(U,L) \wedge (U',K') & := & (U \cap U', K \cap K') & 1 := (X,X) \\
(U,K) \prec (U',K') & :\Longleftrightarrow & K \subseteq U' &
\end{array}$$

Morphisms on **SCS***: adjoint pairs of approximable relations.*

The logic of the probabilistic powerdomain

Given a strong proximity lattice $\mathcal{L}$, construct $\mathcal{VL}$ by

generators $\langle \varphi, r \rangle$ for all $\varphi \in \mathcal{L}$ and all $r \in (0,1) \cap \mathbb{Q}$
with the intended reading: probability of φ is greater than r.

axioms

$$\frac{}{\langle 0, p \rangle \prec 0}$$

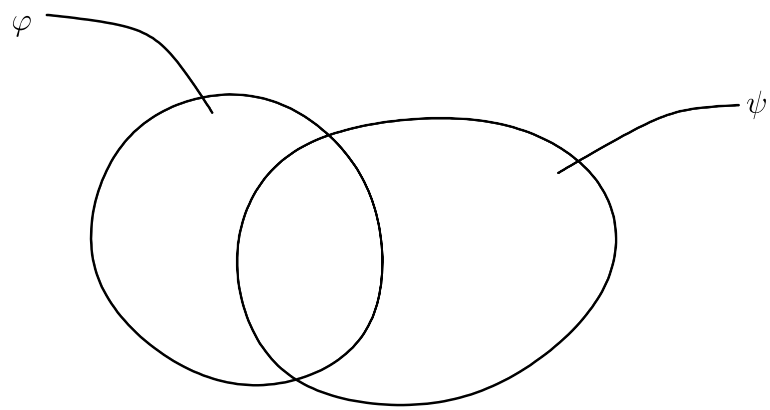
$$\frac{\varphi \vee \psi \prec \rho \quad \varphi \wedge \psi \prec \sigma \quad p + q > r + s}{\langle \varphi, p \rangle \wedge \langle \psi, q \rangle \prec \langle \rho, r \rangle \vee \langle \sigma, s \rangle}$$

$$\frac{\varphi \prec \rho \wedge \sigma \quad \psi \prec \rho \vee \sigma \quad p + q > r + s}{\langle \varphi, p \rangle \wedge \langle \psi, q \rangle \prec \langle \rho, r \rangle \vee \langle \sigma, s \rangle}$$

Theorem. [Heckmann 1994; J & Moshier 2002]

If $\mathcal{L}$ is a domain logic that is sound and complete for the stably compact space X, then logic $\mathcal{V}\mathcal{L}$ is sound and complete for the probabilistic power space $\mathcal{V}(X)$.

Illustrating the soundness of the first modularity law (over-simplified)



$$\frac{p + q > r + s}{\langle \varphi, p \rangle \wedge \langle \psi, q \rangle \prec \langle \varphi \vee \psi, r \rangle \vee \langle \varphi \wedge \psi, s \rangle}$$

I. Stone duality for algebraic domains

II. Domain theory in logical form

III. Stone duality for continuous domains

IV. DTLF for continuous domains?

V. Final thoughts

The vexed issue of the function space

The success with the probabilistic power space construction on stably compact spaces should not blind us and make us forget that it is still an open problem to characterise those strong proximity lattices which arise as Stone duals of FS-domains, and consequently, that to date there is no satisfactory logical treatment of the function space construction.

In the meantime, there have been numerous attempts to “circumvent” the problem of combining higher-order computation and probability:

The vexed issue of the function space

The success with the probabilistic power space construction on stably compact spaces should not blind us and make us forget that it is still an open problem to characterise those strong proximity lattices which arise as Stone duals of FS-domains, and consequently, that to date there is no satisfactory logical treatment of the function space construction.

In the meantime, there have been numerous attempts to “circumvent” the problem of combining higher-order computation and probability:

Ehrhard, Pagani & Tasson (2015)

The vexed issue of the function space

The success with the probabilistic power space construction on stably compact spaces should not blind us and make us forget that it is still an open problem to characterise those strong proximity lattices which arise as Stone duals of FS-domains, and consequently, that to date there is no satisfactory logical treatment of the function space construction.

In the meantime, there have been numerous attempts to "circumvent" the problem of combining higher-order computation and probability:

Ehrhard, Pagani & Tasson (2015)
Heunen, Kammar, Staton & Yang (2017)

The vexed issue of the function space

The success with the probabilistic power space construction on stably compact spaces should not blind us and make us forget that it is still an open problem to characterise those strong proximity lattices which arise as Stone duals of FS-domains, and consequently, that to date there is no satisfactory logical treatment of the function space construction.

In the meantime, there have been numerous attempts to "circumvent" the problem of combining higher-order computation and probability:

Ehrhard, Pagani & Tasson (2015)
Heunen, Kammar, Staton & Yang (2017)
Goubault-Larrecq (2019)

The vexed issue of the function space

The success with the probabilistic power space construction on stably compact spaces should not blind us and make us forget that it is still an open problem to characterise those strong proximity lattices which arise as Stone duals of FS-domains, and consequently, that to date there is no satisfactory logical treatment of the function space construction.

In the meantime, there have been numerous attempts to "circumvent" the problem of combining higher-order computation and probability:

Ehrhard, Pagani & Tasson (2015)
Heunen, Kammar, Staton & Yang (2017)
Goubault-Larrecq (2019)
Jia, Lindenhovius, Mislove & Zamdzhiev (2021)

The bitopological nature of Stone duality

Strong proximity lattices give rise to two topologies, an “upper” topology given by round ideals and a “lower” topology given by round filters. This insight is the starting point for a bitopological reading of much of Stone duality:

A. Jung and M. A. Moshier. On the bitopological nature of Stone duality. Technical Report CSR-06-13, School of Computer Science, The University of Birmingham, 2006. 110 pages.

... but this was the topic for a BLAST tutorial previously.

Papers

S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Semantic Structures*, volume 3 of *Handbook of Logic in Computer Science*, pages 1–168. Clarendon Press, 1994

A. Jung and Ph. Sünderhauf. On the duality of compact vs. open. In S. Andima, R. C. Flagg, G. Itzkowitz, P. Misra, Y. Kong, and R. Kopperman, editors, *Papers on General Topology and Applications: Eleventh Summer Conference at the University of Southern Maine*, volume 806 of *Annals of the New York Academy of Sciences*, pages 214–230, 1996

M. A. Moshier and A. Jung. A logic for probabilities in semantics. In Julian Bradfield, editor, *Computer Science Logic*, volume 2471 of *Lecture Notes in Computer Science*, pages 216–231. Springer Verlag, 2002

G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003