Journées Topologie et Informatique

Topology in Denotational Semantics

Thomas Ehrhard

Preuves, Programmes et Systèmes (PPS)
Université Paris Diderot - Paris 7
and CNRS

March 21, 2013

The usual stateful approach to programming

Most current programming languages are based on *states*.

A program is a sequence of instructions (including *while* or *goto* instructions allowing for loops).

An instruction induces a change of the state of the machine, and its behaviour depends on the state.

Abstract models behind this approach: Turing machines, automata. . .

The functional approach

There is another approach, more in the spirit of recursion theory where programs are *functions* taking values to values.

It is based on *rewriting*, does not require a machine (automaton) and has no notion of state.

Pros

- conceptually simpler
- allows to write higher order programs (taking programs as arguments)
- direct connection with mathematical proof systems (Curry Howard)
- stateful extensions are possible, in an elegant way (monads).

Cons

- computational complexity harder to evaluate
- some algorithms are more naturally expressed in a stateful model.

A functional language: PCF

The language *Programming Computational Functions* (PCF) has been introduce by Plotkin in 1977. It is Turing complete.

One ground type ι of natural numbers.

If σ and τ are types, then $\sigma \Rightarrow \tau$ is a type.

Example of types: ι, $\iota \Rightarrow \iota$, $\iota \Rightarrow (\iota \Rightarrow \iota)$, $(\iota \Rightarrow \iota) \Rightarrow \iota$.

Any partial recursive function from $\mathbb{N}$ to $\mathbb{N}$ can be represented as a PCF term of type $\iota \Rightarrow \iota$.

Syntax and typing rules for PCF

The language uses variables $x, y, \ldots$.

Typing context: finite partial function Γ from variables to types.

Typing judgement $\Gamma \vdash M : \sigma$ means that the term M is of type σ when its variables have types specified by Γ.

There is a set of deduction rules for deriving such judgements.

For each natural number $\mathbb{N}$, there is a *constant* which is a term of PCF:

$$\frac{}{\Gamma \vdash \underline{n} : \iota}$$

Each variable is a term

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}$$

Successor, predecessor and conditional constructions

$$\frac{\Gamma \vdash M : \iota}{\Gamma \vdash \mathrm{succ}(M) : \iota} \qquad \frac{\Gamma \vdash M : \iota}{\Gamma \vdash \mathrm{pred}(M) : \iota}$$

$$\frac{\Gamma \vdash M : \iota \qquad \Gamma \vdash P : \sigma \qquad \Gamma \vdash Q : \sigma}{\Gamma \vdash \mathrm{if}(M, P, Q) : \sigma}$$

PCF: the lambda-calculus constructs

To define a function:

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x^{\sigma}\, M : \sigma \Rightarrow \tau}$$

and to apply a function to an argument:

$$\frac{\Gamma \vdash M : \sigma \Rightarrow \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash (M)\, N : \tau}$$

PCF: the fixpoint operator

The last construct is essential for Turing-completeness.

$$\frac{\Gamma \vdash M : \sigma \Rightarrow \sigma}{\Gamma \vdash \mathrm{fix}(M) : \sigma}$$

PCF: operational semantics

Until now, we have said nothing about the "meaning" of these terms, ie. on their computational behaviour.

This *operational semantics* is specified by a rewriting system $\rightsquigarrow$, defined by the following rules.

$$\frac{}{\mathrm{succ}(\underline{n}) \rightsquigarrow \underline{n+1}} \qquad \frac{}{\mathrm{pred}(\underline{0}) \rightsquigarrow \underline{0}} \qquad \frac{}{\mathrm{pred}(\underline{n+1}) \rightsquigarrow \underline{n}}$$

$$\frac{M \rightsquigarrow M'}{\mathrm{succ}(M) \rightsquigarrow \mathrm{succ}(M')} \qquad \frac{M \rightsquigarrow M'}{\mathrm{pred}(M) \rightsquigarrow \mathrm{pred}(M')}$$

The rewriting rules for conditionals:

$$\frac{}{\mathrm{if}(\underline{0}, P, Q) \rightsquigarrow P} \qquad \frac{}{\mathrm{if}(\underline{n+1}, P, Q) \rightsquigarrow Q}$$

$$\frac{M \rightsquigarrow M'}{\mathrm{if}(M, P, Q) \rightsquigarrow \mathrm{if}(M', P, Q)}$$

The rewriting rules for the lambda-calculus fragment

The main rule is β-reduction:

$$\frac{}{(\lambda x^{\sigma}\, M)\, N \rightsquigarrow M\,[N/x]}$$

Then we have to specify how to reduce in an abstraction or an application.

$$\frac{M \leadsto M'}{\lambda x^\sigma\, M \leadsto \lambda x^\sigma\, M'}$$

$$\frac{P \leadsto P' \quad \text{and } P \text{ is not of the shape } \lambda x^\sigma\, M}{(P)\, N \leadsto (P')\, N}$$

This last rule specifies that we have a *call-by-name* operational semantics.

The last rule concerns fixpoints:

$$\frac{}{\mathrm{fix}(M) \rightsquigarrow (M)\,\mathrm{fix}(M)}$$

Fact

For any term M, there is at most one rule which can apply for reducing it to a term M': the rewriting relation $\rightsquigarrow$ is a deterministic evaluation strategy on PCF terms.

Easy to see by induction on M. The strategy can be implemented by means of an *abstract machine* (a stateful device).

Subject reduction

Reduction is compatible with typing.

Fact

If $\Gamma \vdash M : \sigma$ *and* $M \leadsto M'$ *then* $\Gamma \vdash M' : \sigma$.

Programming in PCF: an example

We want to represent in PCF a total function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ which computes the inf of two natural numbers: $f(n, 0) = n$, $f(0, m) = m$ and $f(n+1, m+1) = 1 + f(n, m)$.

One defines a closed term F of type $\sigma \Rightarrow \sigma$ where $\sigma = (\iota \Rightarrow (\iota \Rightarrow \iota))$ is the expected type for our function inf:

$$F = \lambda f^{\sigma}\, \lambda x^{\iota}\, \lambda y^{\iota}\, \mathrm{if}(x, y, \mathrm{if}(y, x, ((f)\,\mathrm{pred}(x))\,\mathrm{pred}(y)))$$

and then $\mathrm{inf} = \mathrm{fix}(F)$ is a closed term of type σ which has the expected behaviour.

Computations at ground type ι

A term M is *normal* if $M \rightsquigarrow M'$ for no M' (M cannot be reduced).

And M is closed if it has no free variables (the notion of free variable is the same as in first order logic, considering λ as a binder).

Fact

If M is closed and normal and $\vdash M : \iota$, then $M = \underline{n}$ for some $n \in \mathbb{N}$.

Turing completeness of PCF

So if $\vdash M : \iota$ (M is closed), there are two possibilities:

- either $M \rightsquigarrow^* \underline{n}$ (where $n \in \mathbb{N}$ is uniquely defined), notation $M \downarrow \underline{n}$
- or there is a uniquely defined sequence of terms $(M_i)_{i \in \mathbb{N}}$ such that $M_0 = M$ and $M_i \rightsquigarrow M_{i+1}$ for each i, notation $M \uparrow$.

Theorem

Let $f : \mathbb{N} \to \mathbb{N}$ be a partial recursive function. There is a term M with $\vdash M : \iota \Rightarrow \iota$ such that, for any $n, m \in \mathbb{N}$

- *if $f(n) = m$ then $(M)\,\underline{n} \downarrow \underline{m}$*
- *and if $f(n)$ is undefined then $(M)\,\underline{n} \uparrow$.*

The converse is obviously true (Church Thesis).

Observational equivalence

Let M and M' be closed with $\vdash M : \sigma$ and $\vdash M' : \sigma$.

We can say that M and M' are equivalent if they produce the same results when used in the same environment.

Definition

$M \sim M'$ if, for any closed term C with $\vdash C : \sigma \Rightarrow \iota$ one has

$$\forall n \in \mathbb{N} \quad (C)\,M \downarrow \underline{n} \Leftrightarrow (C)\,M' \downarrow \underline{n}\,.$$

This is a nice definition, but it is usually quite difficult to prove that two terms are equivalent.

Denotational semantics is a convenient way of proving such equivalences.

The Myhill Shepherdson Theorem

Let $\mathcal{P}$ be the set of partial recursive functions $\mathbb{N} \to \mathbb{N}$. An *effective operation* is a partial function $\Phi : \mathcal{P} \to \mathbb{N}$ such that there is $e \in \mathbb{N}$ such that, for any $n \in \mathbb{N}$

$$\Phi(\varphi_n) = \varphi_e(n)\,.$$

The MS Theorem implies:

Theorem

Let Φ be an effective operation.

- *If $f \subseteq g \in \mathcal{P}$, then $\Phi(f) = n \Rightarrow \Phi(g) = n$.*
- *If $\Phi(f) = n$ then there is a finite function $f_0 \subseteq f$ such that $\Phi(f_0) = n$.*

Let $\mathcal{P}'$ be the set of all partial functions $\mathbb{N} \to \mathbb{N}$, so that $\mathcal{P} \subseteq \mathcal{P}'$.

Any effective operation $\Phi : \mathcal{P} \to \mathbb{N}$ can be extended to a partial function $\Phi' : \mathcal{P}' \to \mathbb{N}$ setting

$$\Phi'(f) = \begin{cases} n & \text{if there is } f_0 \subseteq f,\ f_0 \text{ finite} \\ & \qquad \text{such that } \Phi(f_0) = n \\ \text{undefined} & \text{otherwise} \end{cases}$$

Complete partial orders (cpos)

A subset D of a partially ordered set $(X, \leq)$ is directed if

- D is non empty
- and $\forall u_1, u_2 \in D \, \exists u \in D \;\; u_1 \leq u$ and $u_2 \leq u$.

Intuition: natural generalization of increasing sequences.

NB: a finite directed set has a maximal element.

$(X, \leq)$ is a cpo if all directed subsets D of X have a least upper bound, denoted as $\bigvee D$.

$(\mathcal{P}', \subseteq)$ is a cpo, but $(\mathcal{P}, \subseteq)$ is not.

Add to $\mathbb{N}$ a new element $\perp$ corresponding to the undefined computation: $\mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}$ endowed with the following order relation:

$$u \leq v \quad \text{if} \quad u = \perp \ \text{or} \ u = v\,.$$

$\mathbb{N}_\perp$ is (trivially) a cpo: directed sets in $\mathbb{N}_\perp$ have at most 2 elements!

Then Φ' is a total function $\mathcal{P}' \to \mathbb{N}_\perp$ which satisfies

- $f \subseteq g \Rightarrow \Phi'(f) \leq \Phi'(g)$
- and if $\mathcal{D} \subseteq \mathcal{P}'$ is directed then $\Phi'(\cup\mathcal{D}) = \bigvee_{f\in\mathcal{D}} \Phi'(f)$.

This observation is the starting point of denotational semantics (Dana Scott).

Towards a Mathematical Semantics of Programming Languages,
Dana Scott and Christopher Strachey, 1971.

Even if they considered stateful programming languages, this paper settled the bases of the denotational semantics of functional languages.

Stateful languages can be translated in functional languages (this syntactic transformation is also known as denotational semantics).

Idea: interpret

- PCF types as cpos
- PCF terms as monotone and directed lub preserving functions.

This does not work, because the category of ω-algebraic cpos and monotone and directed lubs preserving functions is not cartesian closed.

However there are suitable subcategories which are cartesian closed. For instance, the category of Scott domains.

A *Scott domain* is a cpo $(X, \leq)$

- which has a least element $\perp$
- where each bounded subset has a lub
- where any element is the lub of its compact lower bounds (algebraicity)
- and which has at most countably many compact elements (ω-algebraicity).

$u \in X$ compact means that if $D \subseteq X$ is directed and $u \leq \bigvee D$, then $\exists v \in D \; u \leq v$. In some sense, a compact element is “finite”.

Scott Topology

Any cpo $(X, \leq)$ has a natural topology for which $U \subseteq X$ is open if:

- $\forall u, v \in X \ (u \leq v \text{ and } u \in U) \Rightarrow v \in U$
- and for any $D \subseteq X$ directed, if $\bigvee D \in U$ then $D \cap U \neq \emptyset$.

Equivalently $F \subseteq X$ is closed if

- $\forall u, v \in X \ (u \leq v \text{ and } v \in F) \Rightarrow u \in F$
- and for any $D \subseteq X$ directed, if $D \subseteq F$ then $\bigvee D \in F$.

Scott domains are topological spaces

Fact

If $(X, \leq)$ is a Scott domain, the Scott topology on X is T_0 and $\leq$ is the specialization order of the Scott topology.

T_0 is the weakest form of separation: if $u \neq v$ there is an open set such that $u \in U$ and $v \notin U$, or such that $v \in U$ and $u \notin U$.

The specialization order of a T_0 space is defined by: $u \leq v$ if any open set which contains u also contains v.

Fact

If $(X, \leq)$ and $(Y, \leq)$ are Scott domains and $f : X \to Y$, the two following conditions are equivalent

- *f is continuous for the Scott topologies of X and Y*
- *f is monotone and preserves the lubs of all directed sets.*

This shows that the category of Scott domains and monotone and directed lubs preserving maps is a full subcategory of the category of topological spaces and continuous maps.

Fact

The category of Scott domains and Scott continuous functions is cartesian closed.

If $(X, \leq)$ and $(Y, \leq)$ are Scott domains, then

- $X \times Y$ equiped with the product order is a Scott domain and the projections are continuous
- the set $X \Rightarrow Y$ of Scott continuous function $X \rightarrow Y$, equiped with the pointwise order ($f \leq g$ if $\forall a \in X\ f(a) \leq g(a)$) is a Scott domain
- the evaluation map $(X \Rightarrow Y) \times X \rightarrow Y$ is Scott continuous.

This fact is surprising because the category of topological spaces and continuous functions is not cartesian closed.

It is probably related to the following very specific feature of Scott topology.

Fact

If $(X, \leq)$, $(Y, \leq)$ *and* $(Z, \leq)$ *are Scott domains, a function* $f : X \times Y \to Z$ *is continuous as soon as it is separately continuous.*

NB: the Scott topology of $X \times Y$ is the product topology of the Scott topologies of X and Y.

Any Scott continuous function $f : X \to X$ has a least fixpoint, namely $\bigvee_{n\in\mathbb{N}} f^n(\bot)$. (Indeed $\bot \leq f(\bot) \leq f^2(\bot) \leq \cdots$).

By cartesian closeness, the function

$$\begin{aligned}(X \Rightarrow X) &\to X \\ f &\mapsto \bigvee_{n\in\mathbb{N}} f^n(\bot)\end{aligned}$$

is itself Scott continuous.

This provides a direct interpretation of the $\mathrm{fix}(M)$ construction of PCF.

With each type σ of PCF we can associate a Scott domain $[\sigma]$:

- $[\iota] = \mathbb{N}_\perp$
- $[\sigma \Rightarrow \tau] = [\sigma] \Rightarrow [\tau]$.

and with any term M such that $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash M : \tau$ we can associate a continuous function

$$[M] : [\sigma_1] \times \cdots \times [\sigma_n] \to [\tau]\,.$$

Theorem

Soundness: if $M \rightsquigarrow M'$ then $[M] = [M']$.

This interpretation is an invariant of the reduction.

So in particular, if $\vdash M : \iota$ and $M \rightsquigarrow^* \underline{n}$, then $[M] = n \in \mathbb{N}_\perp = [\iota]$.

The converse is also true.

Theorem

Adequacy: if $\vdash M : \iota$ *and* $[M] = n \neq \perp$, *then* $M \rightsquigarrow^* \underline{n}$.

The proof uses a very powerful tool in the semantics of lambda-calculus: *logical relations*, introduced by Tait (*reducibility*).

A logical relation

The following idea is due to Plotkin.

By induction on types one defines for each type σ a relation $\mathcal{R}^\sigma$ between elements of $[\sigma]$ and closed terms of type σ.

- $n \,\mathcal{R}^\iota\, M$ if $n \neq \perp \Rightarrow M \rightsquigarrow^* \underline{n}$.
- $f \,\mathcal{R}^{\sigma\Rightarrow\tau}\, M$ if, for all $u \in [\sigma]$ and P such that $\vdash P : \sigma$, if $u \,\mathcal{R}^\sigma\, P$ then $f(u) \,\mathcal{R}^\tau\, (M)\,P$.

By induction on terms one proves that if $\vdash M : \sigma$, then $[M] \,\mathcal{R}^\sigma\, M$. (One needs actually to prove a more complicated statement involving non closed terms). Adequacy follows.

Main lemma: $\{u \in [\sigma] \mid u \,\mathcal{R}^\sigma\, M\}$ is Scott closed.

A consequence of adequacy

Theorem

If M and M' are closed of type σ, then

$$[M] = [M'] \Rightarrow M \sim M' .$$

Indeed, let $\vdash C : \sigma \Rightarrow \iota$.

If $(C)\, M \downarrow \underline{n}$ then $[(C)\, M] = n$ by soundness.

But $[(C)\, M] = [C]([M]) = [C]([M']) = [(C)\, M']$.

So $[(C)\, M'] = n$ and hence $(C)\, M' \downarrow \underline{n}$ by adequacy.

Full abstraction is the converse property:

$$M \sim M' \Rightarrow [M] = [M'] .$$

This property does not hold in Scott domains but is highly desirable because it means that the model provide an abstract and complete way of reasoning on programs.

The reason is that it contains “non deterministic” morphisms such as

$$\begin{aligned}\mathrm{por} : \mathbb{N}_\perp \times \mathbb{N}_\perp &\to \mathbb{N}_\perp \\ (n, m) &\mapsto \begin{cases} 0 & \text{if } n = 0 \text{ or } m = 0 \\ \perp & \text{otherwise} \end{cases}\end{aligned}$$

This function is Scott continuous.

Let $\Omega = \mathrm{fix}(\lambda x^\iota\, x)$, it is the everlooping program of type $\mathbb{N}_\perp$ $(\Omega \rightsquigarrow (\lambda x^\iota\, x)\,\Omega \rightsquigarrow \Omega \rightsquigarrow \ldots)$. One has $[\Omega] = \perp$.

Consider $\lambda h^{\iota \Rightarrow (\iota \Rightarrow \iota)}\,\Omega$, also denoted as Ω, and

$$\mathrm{port} = \lambda h^{\iota \Rightarrow (\iota \Rightarrow \iota)}\, \mathrm{if}(((h)\,\underline{0})\,\Omega, \mathrm{if}(((h)\,\Omega)\,\underline{0}, \underline{0}, \Omega), \Omega)\,.$$

then we have $[\Omega] \neq [\mathrm{port}]$ since $[\Omega](\mathrm{por}) = \perp$ and $[\mathrm{port}](\mathrm{por}) = 0$.

But $\mathrm{port} \sim \Omega$ as we shall see.

port: por taster.

The trouble with por:

- $\mathrm{por}(\bot,\bot) = \bot$
- $\mathrm{por}(0,0) = 0$ so that por needs its arguments to produce the result 0 from $(0,0)$
- but since $\mathrm{por}(0,\bot) = 0$ and $\mathrm{por}(\bot,0) = 0$ it is impossible to say which are the arguments really used by the function to compute the result.

Say that $f : X \to \mathbb{N}_\perp$ is stable if

- f is Scott continuous
- if $f(u) = n \neq \perp$ there is $u_0 \leq u$ such that $f(u_0) = n$ and for any $v \leq u$, if $f(v) = n$ then $u_0 \leq v$.

u_0 is the part of u that f has used to produce the result n.

por is not stable: take $u = (0, 0)$, there is no u_0 with the required property.

Coherence spaces

One can build nice categories with stable functions as morphisms.

However there is no topology for which continuity would be equivalent to stability.

The simplest of these categories is that of coherence spaces and stable functions.

A coherence space is a structure $X = (|X|, \frown_X)$ where $|X|$ is a countable set and $\frown_X$ is a binary reflexive and symmetric relation on $|X|$.

A clique of X is a subset u of $|X|$ such that $\forall a, a' \in x \; a \frown_X a'$.

Ordered by $\subseteq$, the cliques of X form a (very well-behaved) Scott domain. Let $\mathrm{Cl}(X)$ be this Scott domain.

A function $f : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ is stable if it is Scott continuous and moreover

$$\forall u_1, u_2 \in \mathrm{Cl}(X) \quad u_1 \cup u_2 \in \mathrm{Cl}(X) \Rightarrow f(u_1 \cap u_2) = f(u_1) \cap f(u_2) \,.$$

Fact

A function $f : Cl(X) \to Cl(Y)$ is stable iff for all $u \in Cl(X)$ and all $b \in f(u)$ there exists $u_0 \subseteq u$ finite such that $b \in f(u_0)$ and, for all $u_1 \subseteq u$, if $b \in f(u_1)$ then $u_0 \subseteq u_1$.

Theorem

The category of coherence spaces and stable functions is cartesian closed and is a model of PCF.

ι interpreted by the coherence space N such that $|N| = \mathbb{N}$ and $n \frown_N m$ if $n = m$. $\mathrm{Cl}(N)$ is clearly isomorphic to $\mathbb{N}_\perp$.

Moreover, this model satisfies the adequacy property.

In this model $[\mathrm{port}] = \emptyset = [\Omega]$.

This shows that $\mathrm{port} \sim \Omega$.

The stable model is not fully abstract (there is a stable function $\mathbb{N}_\perp^3 \to \mathbb{N}_\perp$ which is not definable in PCF and one can define a taster for this function).

Linear decomposition of stable functions

A function $f : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ is linear if it is stable and commutes with all unions:

$$\forall u_1, u_2 \in \mathrm{Cl}(X) \quad u_1 \cup u_2 \in \mathrm{Cl}(X) \Rightarrow f(u_1 \cup u_2) = f(u_1) \cup f(u_2)\,.$$

Fact

There is a canonical bijection between linear functions $Cl(X) \to Cl(Y)$ *and cliques of the coherence space* $X \multimap Y$ *where* $|X \multimap Y| = |X| \times |Y|$ *and* $(a, b) \frown_{X \multimap Y} (a', b')$ *if*

$$a \frown_X a' \Rightarrow b \frown_Y b'$$
$$b \asymp_Y b' \Rightarrow a \asymp_X a'$$

where $a \asymp_X a'$ if $a = a'$, or $a \frown_X a'$ does not hold.

If $t \in \mathrm{Cl}(X \multimap Y)$, the associated linear map $f : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ is defined by $f(u) = \{b \in |Y| \mid a \in u \text{ and } (a, b) \in t\}$.

This suggests to define

- $X^\perp$ by $|X^\perp| = |X|$ and $a_1 \frown_{X^\perp} a_2$ if $a_1 \asymp_X a_2$ so that $X^{\perp\perp} = X$. Then $X^\perp \simeq (X \multimap \perp)$ where $\perp$ is the one point coherence space. If one think of $\perp$ as the "field" (1-dimensional vector space), then $X^\perp$ is the analogue of a linear dual of X.
- $X \otimes Y$ by $|X \otimes Y| = |X| \times |Y|$ and $(a_1, b_1) \frown_{X \otimes Y} (a_2, b_2)$ if $a_1 \frown_X a_2$ and $b_1 \frown_Y b_2$. Then $X \multimap Y = (X \otimes Y^\perp)^\perp$.

We have a structure completely similar to that of the category of finite dimensional vector space and linear map, but less “degenerated” because here, it is not true that $(X \otimes Y)^{\perp} \simeq X^{\perp} \otimes Y^{\perp}$.

Coherence spaces and stable functions form a cartesian closed category: we have a coherence space $X \Rightarrow Y$ with a canonical bijection between $\mathrm{Cl}(X \Rightarrow Y)$ and stable functions $\mathrm{Cl}(X) \to \mathrm{Cl}(Y)$.

Decomposition of implication

Girard's observation:

$$X \Rightarrow Y \;=\; !X \multimap Y$$

where $|!X|$ is the set of all finite (multi)cliques of X and $u_1 \frown_{!X} u_2$ if $u_1 \cup u_2 \in \mathrm{Cl}(X)$.

This led him to the introduction of a refinement of intuitionistic and classical logic: linear logic.

Since LL has a strong intuitive background of linear algebra, it was tempting to define models where formulae are interpreted as vector spaces.

However, the construction $!X$ will create infinite dimensional vector spaces, and therefore topology will come in naturally.

It will be quite different from Scott topology.

A starting point: observe that if $u \in \mathrm{Cl}(X)$ and $u' \in \mathrm{Cl}(X^\perp)$ then $u \cap u'$ has at most one element.

Interpret u and u' as $\{0,1\}$-valued vectors: the sum $\Sigma_{a \in |X|} u_a u'_a$ has at most one non-zero term.

Let us relax this condition by requiring $u \cap u'$ to be finite so that the corresponding sum will only have a finite number of non-zero terms.

A typical LL definition

Given a set I and a subset $\mathcal{F}$ or $\mathcal{P}(I)$, set

$$\mathcal{F}^{\perp} = \{u' \subseteq I \mid \forall u \in \mathcal{F}\ u \cap u' \text{ finite}\}\,.$$

Observe that

- $\mathcal{F} \subseteq \mathcal{G} \Rightarrow \mathcal{G}^{\perp} \subseteq \mathcal{F}^{\perp}$
- $\mathcal{F} \subseteq \mathcal{F}^{\perp\perp}$.

hence $\mathcal{F}^{\perp\perp\perp} = \mathcal{F}^{\perp}$.

This means that any set $\mathcal{G}$ which has been defined as $\mathcal{F}^{\perp}$ for some $\mathcal{F}$ will satisfy $\mathcal{G}^{\perp\perp} = \mathcal{G}$.

A *finiteness space* is a structure $X = (|X|, \mathrm{F}(X))$ where $|X|$ is a countable set and $\mathrm{F}(X) \subseteq \mathcal{P}(|X|)$ satisfies $\mathrm{F}(X)^{\perp\perp} = \mathrm{F}(X)$.

Similarity with coherence spaces: $\mathrm{F}(X)$ looks like a set of cliques. For instance: $u_1 \subseteq u_2 \in \mathrm{F}(X) \Rightarrow u_1 \in \mathrm{F}(X)$.

But there are big differences:

- $u_1, u_2 \in \mathrm{F}(X) \Rightarrow u_1 \cup u_2 \in \mathrm{F}(X)$, whereas the union of two cliques is not a clique in general
- any finite subset of $|X|$ belongs to $\mathrm{F}(X)$
- $\mathrm{F}(X)$ is not closed under directed unions, whereas cliques are.

Let $\mathbf{k}$ be a field (no topological assumptions).

Let $\mathbf{k}\langle X\rangle$ be the set of all functions $x : |X| \to \mathbf{k}$ such that $\mathrm{supp}(x) = \{a \in |X| \mid x(a) \neq 0\}$ is an element of $\mathrm{F}(X)$.

Since $\mathrm{supp}\,(x_1 + x_2) \subseteq \mathrm{supp}(x_1) \cup \mathrm{supp}(x_2)$ and $\mathrm{supp}\,(\alpha x) \subseteq \mathrm{supp}(x)$ (for $\alpha \in \mathbf{k}$),

Fact

$\mathbf{k}\langle X\rangle$ *is a* $\mathbf{k}$*-vector space.*

Topology

Let $u' \in \mathrm{F}(X)^{\perp}$. Let

$$\mathrm{V}(u') = \{x \in \mathbf{k}\langle X\rangle \mid \mathrm{supp}(x) \cap u' = \emptyset\}\,.$$

This is a linear subspace of $\mathbf{k}\langle X\rangle$. Observe that $\mathrm{V}(u'_1 \cup u'_2) = \mathrm{V}(u'_1) \cap \mathrm{V}(u'_2)$.

Say that $U \subseteq \mathbf{k}\langle X\rangle$ is open if $\forall x \in U\, \exists u' \in \mathrm{F}(X)^{\perp}\ x + \mathrm{V}(u') \subseteq U$.

This defines a topology on $\mathbf{k}\langle X\rangle$ which is T_2.

Fact

For each $u' \in \mathrm{F}(X)^{\perp}$, the space $\mathrm{V}(u')$ is both open and closed. So $\mathbf{k}\langle X\rangle$ is zero-dimensional (the $x + \mathrm{V}(u')$ are a basis of clopen sets).

Fact

This topology is compatible with the linear structure (addition and scalar multiplication are continuous). We give $\mathbf{k}$ the discrete topology.

Remark: when $|X|$ is finite, $\mathbf{k}\langle X\rangle = \mathbf{k}^{|X|}$ is finite dimensional and its topology is discrete.

Linearly topologized vector spaces

The corresponding general notion of topological vector space has been introduced by S. Lefschetz in 1942: *linearly topologized vector spaces* (ltvs).

They are a “purely algebraic” analogue of locally convex topological vector spaces (based on the topology of $\mathbb{R}$ or $\mathbb{C}$).

Cauchy completeness

A *net* is a family $(x_i)_{i \in I}$ of elements of $\mathbf{k}\langle X \rangle$ where I is a directed ordered set.

It converges to $x \in \mathbf{k}\langle X \rangle$ if for all neighborhood U of 0 there exists $i \in I$ such that $x_j - x \in U$ for all $j \geq i$.

It is *Cauchy* if for all neighborhood U of 0 there exists $i \in I$ such that $x_j - x_{j'} \in U$ for all $j, j' \geq i$.

Theorem

Any Cauchy net in $\mathbf{k}\langle X\rangle$ converges: $\mathbf{k}\langle X\rangle$ is Cauchy complete.

The proof uses the fact that $\mathrm{F}(X) = \mathrm{F}(X)^{\perp\perp}$ in an essential way. Basic examples:

- $N = (\mathbb{N}, \mathcal{P}_{\mathrm{fin}}(\mathbb{N}))$; indeed $\mathcal{P}_{\mathrm{fin}}(\mathbb{N})^\perp = \mathcal{P}(\mathbb{N})$ and $\mathcal{P}(\mathbb{N})^\perp = \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$. Then $\mathbf{k}\langle N\rangle = \mathbf{k}^{(\mathbb{N})}$, with the discrete topology.
- And $\mathbf{k}\langle N^\perp\rangle = \mathbf{k}^{\mathbb{N}}$ with the product topology (remember that $\mathbf{k}$ is discrete).

Matrix of a linear map

Given $a \in |X|$ define $e_a \in \mathbf{k}\langle X\rangle$ by $(e_a)_{a'} = \begin{cases} 1 & \text{if } a = a' \\ 0 & \text{otherwise} \end{cases}$.

Let $f : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$ be linear and continuous.

Define the *matrix* of f as $M(f) \in \mathbf{k}^{|X|\times|Y|}$ such that $M(f)_{a,b} = f(e_a)_b$.

Define $X \multimap Y$ by $|X \multimap Y| = |X| \times |Y|$ and

$$\mathrm{F}(X \multimap Y) = \{u \times v' \mid u \in \mathrm{F}(X) \text{ and } v' \in \mathrm{F}(Y)^{\perp}\}^{\perp}$$

Theorem

The operation $f \mapsto M(f)$ is a linear isomorphism between the space of linear and continuous maps $\mathbf{k}\langle X \rangle \to \mathbf{k}\langle Y \rangle$ and $\mathbf{k}\langle X \multimap Y \rangle$.

Linear boundedness

A linear subspace B of $\mathbf{k}\langle X\rangle$ is *linearly bounded* if, for any open (and therefore closed) linear subspace U of $\mathbf{k}\langle X\rangle$, the image of B by the canonical projection $\mathbf{k}\langle X\rangle \to \mathbf{k}\langle X\rangle/U$ is finite dimensional.

Fact

B is linearly bounded iff

$$\bigcup_{x\in B} \mathrm{supp}(x) \in \mathrm{F}(X)\,.$$

Theorem

Let $\mathcal{U}$ be a set of linear and continuous maps $\mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$. The two following conditions are equivalent:

- $\{M(f) \mid f \in \mathcal{U}\}$ *is open in* $\mathbf{k}\langle X \multimap Y\rangle$
- *for each* $f \in \mathcal{U}$ *there exists a linearly bounded subspace* B *of* $\mathbf{k}\langle X\rangle$ *and an open subspace* V *of* $\mathbf{k}\langle Y\rangle$ *such that, for any linear and continuous* $g : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$*, if* $g(B) \subseteq V$ *then* $f + g \in \mathcal{U}$*.*

This is the bounded-open topology.

Hypocontinuous bilinear functions

A bilinear function $\mathbf{k}\langle X_1\rangle \times \mathbf{k}\langle X_2\rangle \to \mathbf{k}\langle Y\rangle$ is hypocontinuous if:

- for any linearly bounded subspace B_1 of $\mathbf{k}\langle X_1\rangle$ and any neighborhood V of 0 in $\mathbf{k}\langle Y\rangle$, there is a neighborhood U_2 of 0 in $\mathbf{k}\langle X_2\rangle$ such that $f(B_1 \times U_2) \subseteq V$
- and symmetrically.

Warning: such a map is not necessarily continuous.

Tensor product

$X_1 \otimes X_2$ is defined as $|X_1 \otimes X_2| = |X_1| \times |X_2|$ and

$$\mathrm{F}(X_1 \otimes X_2) = \{u_1 \times u_2 \mid u_i \in \mathrm{F}(X_i) \text{ for } i = 1, 2\}^{\perp\perp}$$

Given $x_i \in \mathbf{k}\langle X_i \rangle$ for $i = 1, 2$, $x_1 \otimes x_2$ defined by $(x_1 \otimes x_2)_{(a_1, a_2)} = (x_1)_{a_1}(x_2)_{a_2}$ belongs to $\mathbf{k}\langle X_1 \otimes X_2 \rangle$.

Theorem

The map $(x_1, x_2) \mapsto x_1 \otimes x_2$ is bilinear and hypocontinuous and is universal: for any Y and any hypocontinuous bilinear $f : \mathbf{k}\langle X_1 \rangle \times \mathbf{k}\langle X_2 \rangle \to \mathbf{k}\langle Y \rangle$, there is exactly one linear and continuous $h : \mathbf{k}\langle X_1 \otimes X_2 \rangle \to \mathbf{k}\langle Y \rangle$ such that $f(x_1, x_2) = h(x_1 \otimes x_2)$.

Remark: tensor and co-tensor do not coincide, $(X \otimes Y)^{\perp}$ and $X^{\perp} \otimes Y^{\perp}$ are not isomorphic in general.

Polynomial and entire functions

A function $f : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$ is polynomial if there are hypocontinuous multilinear maps $f_0, f_1, \ldots, f_k$ with $f_i : \mathbf{k}\langle X\rangle^i \to \mathbf{k}\langle Y\rangle$ such that

$$f(x) = \sum_{i=0}^{k} f_i(\underbrace{x, \ldots, x}_{i\,\times})$$

They form a $\mathbf{k}$-vector space that we can complete under the bounded-open topology to get a space (an ltvs actually) of “entire functions” $\mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$.

Theorem

Assume $\mathbf{k}$ *infinite. The ltvs of entire functions* $\mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$ *equiped with the bounded-open topology is linearly homeomorphic to* $\mathbf{k}\langle (!X) \multimap Y\rangle$ *where* $|!X| = \mathcal{M}_{\mathrm{fin}}(|X|)$ *and*

$$\mathrm{F}(!X) = \{\mathcal{M}_{\mathrm{fin}}(u) \mid u \in \mathrm{F}(X)\}^{\perp\perp}.$$

Each $p \in \mathcal{M}_{\mathrm{fin}}(|X|)$ is a muti-exponent: given $x \in \mathbf{k}\langle X\rangle$, we define $x^p = \prod_{a\in|X|} x_a^{p(a)} \in \mathbf{k}$. Given $M \in \mathbf{k}\langle(!X) \multimap Y\rangle$ we define an entire map $\mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$ by

$$f(x)_b = \sum_{p\in\mathcal{M}_{\mathrm{fin}}(|X|)} M_{p,b} x^p$$

for each $b \in |Y|$.

For each b, this sum is finite, but its number of non-zero terms is not bounded (when b varies).

Differentiation of entire functions

An entire function $f : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle Y\rangle$ is not continuous in general.

It can nevertheless be differentiated, one can define $f' : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle (X \multimap Y)\rangle$ as an entire function.

Iterating we have a nth derivative $f^{(n)} : \mathbf{k}\langle X\rangle \to \mathbf{k}\langle X^{\otimes n} \multimap Y\rangle$.

The Taylor formula holds:

$$f(x+y) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(x)(\underbrace{y,\ldots,y}_{n\times}).$$

This sum converges in the topology of $\mathbf{k}\langle Y\rangle$.

The "!" construction introduces complicated topologies

Theorem

$\mathbf{k}\langle X\rangle$ is metrizable iff there is a countable subset $\mathcal{F}'$ of $\mathrm{F}(X)^{\perp}$ such that $\forall u' \in \mathrm{F}(X)^{\perp}\ \exists v' \in \mathcal{F}'\ u' \subseteq v'$.

Fact

Let 1 be the finiteness space with $|1| = \{*\}$, so that $\mathbf{k}\langle 1\rangle = \mathbf{k}$. The space $\mathbf{k}\langle(!(!1 \multimap 1) \multimap 1)^{\perp}\rangle$ is not metrizable.

$\mathbf{k}\langle !1 \multimap 1\rangle$ is the space of polynomials, with the discrete topology (isomorphic to $\mathbf{k}^{(\mathbb{N})}$).

$\mathbf{k}\langle !(!1 \multimap 1) \multimap 1\rangle$ contains infinite objects such as the operations which maps a polynomial P to $P(P(0))$:

$$\sum_{i=0}^{n} \alpha_i X^i \quad \mapsto \quad \sum_{i=0}^{n} \alpha_i \alpha_0^i$$

Conclusion

The category of entire functions is cartesian closed and has some basic differential operations, providing solid foundations to *calculi with resources* introduced earlier (Boudol), and recasting them in a wider LL setting.

Contrarily to Scott or stable semantics, it does not accomodate general fixpoints.

It has suggested to introduce a new way of approximating lambda-terms by Taylor expansion: Tasson's talk this afternoon.

One of the main applications of denotational semantics is to suggest sound extensions of syntaxes.