DM3 – Le comité contre les chats

simon.castellan@ens-lyon.fr

We recall that a ω-cpo is a (partial) order $(A, \leq)$ such that every increasing sequence $x_1 \leq \ldots \leq x_n \leq \ldots$ has a least upper bound that we write $\sup(x_n)_n$ and A has a least element written $\bot_A$. A Scott-continuous function f between ω-CPOS $(A, \leq)$ and $(B, \leq)$ is a monotonic function such that for every increasing sequence we have:

$$f(\sup(x_n)_n) = \sup(f(x_n))_n$$

This makes sense because $(f(x_n))$ is an increasing sequence of B. We recall from TD9 that if A, B are ω-CPOs the set of continuous functions from A to B is an ω-CPO written $[A \Rightarrow B]$ ordered by the pointwise ordering. **NB:** It does not need to preserve the least element!

In this assignment, all maps between ω-CPOs are assumed Scott-continuous unless explicitly stated otherwise. We recall the syntax of IMP with variables ranging in the set $\mathscr{V}$.

$a, a' ::=$	(*Arithmetic expressions*)
$\mid \underline{n}$	Integer litterals
$\mid a + a' \mid a \times a' \mid a - a'$	Addition, Soustraction and Multiplication
$\mid r$	Variable dereferenciation
$b, b' ::=$	(*Boolean expressions*)
$\mid a = a'$	Integer equality
$\mid b \wedge b' \mid b \vee b' \mid \neg b$	Conjunction, Disjunction and Negation
$C, C' ::=$	(IMP*commands*)
$\mid r := a$	Affectation
$\mid$ `if` b `then` C `else` C'	Conditionals
$\mid$ `while` b `do` C `done`	Unbounded loops
$\mid$ `skip`	No-op
$\mid C; C'$	Sequencing

We recall that memory states are partial functions from $\mathscr{V}$ to $\mathbb{N}$ and we write this set $\mathscr{M}$.

1 Warmup on CPOs

Question 1

Show that if A and B are two ω-CPOS, the product order $A \times B$ is an ω-CPO. Build an ω-CPO 1, such that for all other ω-CPO A there exists a **unique** map $f : A \to 1$.

Question 2

If X and Y are sets, remember from TD9 that $[X \to Y_\bot]$ is the set of partial functions from X to Y with graph inclusion. Define an operation of composition $[Y \to Z_\bot] \times [X \to Y_\bot] \to [X \to Z_\bot]$ and show it is continuous.

Question 3
Show that there is a Scott-continuous function F from $[A \to A]$ to A such that for all Scott-continuous function $f \in [A \to A]$ we have

$$f(F(f)) = F(f).$$

Question 4
By induction on C, build a partial function $[\![C]\!] \in [\mathscr{M} \to \mathscr{M}_\perp]$ such that $\langle C, \sigma \rangle \Downarrow \sigma'$ if and only if $[\![C]\!](\sigma) = \sigma'$ where $\Downarrow$ is the operational semantics seen in TD8, *using the constructions introduced in this section*.

2 Interpreting IMP inside CPOs

In this section we define an interpretation of the language IMP inside any ω-CPO with some extra-structure.

Given a ω-CPO A we define

$$T(A) = [\mathscr{M}_\perp \to A \times \mathscr{M}_\perp].$$

Note that the interpretation seen during the lecture of a Imp commands lives in $T(1) \cong [\mathscr{M}_\perp \to \mathscr{M}_\perp]$.

Question 5
Given a Scott-continuous $f : A \to B$ build a map $T(f) : T(A) \to T(B)$ such that $T(\mathrm{id}_A) = \mathrm{id}_{T(A)}$ and $T(g \circ f) = T(g) \circ T(f)$.

Question 6
Given a ω-CPO A, show that there exists two maps $\eta_A : A \to T(A)$ and $\mu_A : T(T(A)) \to T(A)$ such that $\mu_A \circ T(\eta_A) = \mu_A \circ \eta_{T(A)} = \mathrm{id}_{T(A)}$ and $\mu_A \circ \mu_{T(A)} = \mu_A \circ T(\mu_A)$.

Question 7
Given ω-CPOs A, B, show that there exists a map $s : T(A) \times B \to T(A \times B)$ satisfying the following equation:

$$s(\eta(a), b) = \eta(a, b)$$

Deduce that there are two "natural" maps $T(A) \times T(B) \to T(A \times B)$. Are they identical?

Show that one of them, specialized with $A = B = 1$ was (implicitly) used to interpret the composition of commands in the lectures. We write $\mathrm{sq} : T(A) \times T(B) \to T(A \times B)$ for this map.

Question 8
Let A be an ω-CPO along with

- An object $w_x : \mathbb{N} \to T(A)$ used to interpret affectation where $\mathbb{N}$ is the ω-CPO of natural numbers with equality and $x \in \mathscr{V}$.
- A map $\mathrm{comp} : [A \times A \to A]$ used to interpret sequencing

Show how to build a function

$$C \in \text{IMP} \mapsto [\![C]\!]_A \in T(A)$$

using this data and such that $[\![C]\!]_A(\sigma)$ is defined and equal to (a, σ') if and only if $\langle C, \sigma \rangle \Downarrow \sigma'$ (and no constraints on a).

Question 9
Using the previous question, give an interpretation of IMP commands $[\![C]\!] \in T(\mathbb{N}_\perp)$ such that if $[\![C]\!](\sigma) = (n, \sigma')$ then the corresponding execution of C on σ performs n memory writes.

Question 10
We want to add `abort` to IMP. Show how to modify T to take this into account. What does the new μ morphism look like?

Question 11
(*Optional*) Explain how this translation $C \mapsto [\![C]\!]$ correspond to a syntactic translation from IMP to the (simply-typed) λ-calculus and how IMP can be seen as a purely functional programming language through this encoding.