Scribe: Yee Jiun Song, Dan Stowell Lecturer: Andrew Myers

Today's notes cover:

- revisiting the fixed point theorem
- a metalanguage for denotational semantics

1 Fixed Point Theorem, Reprise

Last time, we showed that given a function $f : D \mapsto D$, with D being a pointed c.p.o. and f continuous, (i.e., $f : [D \mapsto D]$, then fix $f = \bigsqcup f^n(\perp)$ is the least fixed point of f. Perhaps we should revisit this proof to try to eliminate any confusion.

Assume y is a fixed point of f.

$$\begin{gathered}\perp \sqsubseteq y \\ f(\perp) \sqsubseteq y \\ f^2(\perp) \sqsubseteq y) \\ \vdots \\ \forall n. f^n(\perp) \sqsubseteq y \\ \bigsqcup f^n \sqsubseteq y\end{gathered}$$

Now we know that fix f is smaller than any fixed point y and must be the least fixed point.

But how does the concept of a complete partial order have anything to do with the semantics of a programming language? If we take f as the rule operator R, $\sqsubseteq$ as $\leq$, and $\perp$ as $\emptyset$, we can use what we've learned about fixed points to describe programs.

As a quick aside, what does continuity mean? All behavior of a function is explained by finite approximation. In other words, nothing funny happens at infinity.

$$C[\![\textbf{while } b \textbf{ do } c]\!] = \text{fix } \lambda d \in (\Sigma \mapsto \Sigma_\perp).(\lambda\sigma \in \Sigma. \text{ if } \neg B[\![b]\!]\sigma \text{ then } \sigma \text{ else } d^*(C[\![c]\!]\sigma))$$

We need to show that the function we are taking a fixed point of is continuous. For our purposes, continuity means $\forall \omega - \text{chains} d1, d2, \ldots, f(\bigsqcup d_n = \bigsqcup f(d_n$. We also want to show that the function is monotonic, i.e. $d \sqsubseteq d' \Rightarrow f(d) \sqsubseteq f(d')$. In this case, d is a function that we want to show $\forall\sigma. f(d)(\sigma) \sqsubseteq f(d')(\sigma)$.

$$\text{if } \neg B[\![b]\!]\sigma \text{ then } \sigma \text{ else } d^*(C[\![c]\!]\sigma) \sqsubseteq \text{ if } \neg B[\![b]\!]\sigma \text{ then } \sigma \text{ else } d'^*(C[\![c]\!]\sigma)$$

If $B[\![b]\!]$ is false, then we have $\sigma \sqsubseteq \sigma$, which is true by definition. Otherwise, we have $d^*(C[\![c]\!]\sigma) \sqsubseteq d'^*(C[\![c]\!]\sigma)$. $C[\![c]\!]$ will either return $\perp$ or σ'. We know that $\perp \sqsubseteq \perp$ and $d(\sigma') \sqsubseteq d'(\sigma')$. The function we are trying to take the fix of is monotonic.

But how do we know that it is continuous?

$$\bigsqcup \lambda\sigma \in \Sigma. \text{ if } \neg B[\![b]\!] \text{ then } \sigma \text{ else } d_n^*(C[\![c]\!]\sigma)$$

$$\lambda\sigma \in \Sigma. \bigsqcup \text{ if } \neg B[\![b]\!] \text{ then } \sigma \text{ else } d_n^*(C[\![c]\!]\sigma)$$

$$\lambda\sigma \in \Sigma. \text{ if } \neg B[\![b]\!] \text{ then } \sigma \text{ else } \bigsqcup d_n^*(C[\![c]\!]\sigma)$$

$$\lambda\sigma \in \Sigma. \text{ if } \neg B[\![b]\!] \text{ then } \sigma \text{ else } (\bigsqcup d_n)^*(C[\![c]\!]\sigma)$$

2 A Metalanguage

Proving functions to be continuous gets tedious after a while. Why not write a metalanguage in which continuity comes naturally? We can then use this metalanguage in describing denotational semantics.

In this metalanguage, we'll define "types" as domains (i.e., CPOs). We'll be dealing with discrete CPOs, such as the integers. We'll also define the unit CPO as {unit}.

2.1 Lifting

Given some complete partial order D, we know that $D_\perp$ is also a complete partial order:

Given that d ∈ D, the elements of $D_\perp$ are $\lfloor D \rfloor$ and $\perp_D$.

Ordering: $d \sqsubseteq d' \Rightarrow \lfloor d \rfloor \sqsubseteq \lfloor d' \rfloor$

Also, $\perp \sqsubseteq \lfloor d \rfloor$

Now, in order for this to be a complete partial order, we require that ever chain have a least upper bound.

$\perp \ \sqsubseteq \ \perp \ \sqsubseteq ... \Rightarrow$ LUB $= \ \perp$

$\perp \ \sqsubseteq \ \lfloor d_1 \rfloor \ \sqsubseteq \ \lfloor d_2 \rfloor \ \sqsubseteq ... \Rightarrow$ LUB $= \lfloor \bigsqcup d_n \rfloor$

2.2 Product

The product $D \times E$ is a CPO of elements $\langle d, e \rangle$ where $d \in D$, and $e \in E$. They are ordered as such:

$\langle d_1, e_1 \rangle \ \sqsubseteq \ \langle d_2, e_2 \rangle \ \Leftrightarrow \ d_1 \sqsubseteq d_2 \ \wedge \ e_1 \sqsubseteq e_2$

Is it a CPO? Consider this:

$\langle d_1, e_1 \rangle \ \sqsubseteq \ \langle d_2, e_2 \rangle \ \sqsubseteq ...$

$\bigsqcup \langle d_n, e_n \rangle = \langle \bigsqcup d_n, \bigsqcup e_m \rangle$

In addition, the CPO $D \times E$ is pointed if both D and E are pointed. $\perp_{D \times E} = \langle D_\perp, E_\perp \rangle$.

2.3 Tupling and Projection

Tupling: $\langle d_1, d_2, d_3, ..., d_n \rangle$

Projection: $\pi_i \langle d_1, d_2, d_3, ..., d_n \rangle \ = \ d_i$

2.4 Sums

Elements in $D_1 + D_2$ are either D_1 or D_2 tagged.

Elements: $in_1(d_1)$ or $in_2(d_2)$

Ordering: $in_{\mathrm{i}}(d) \sqsubseteq in_{\mathrm{i}}(d') \Leftrightarrow d \sqsubseteq_{D_i} d'$

2.5 Functions

Given CPOs D and E, $[D \to E] \subseteq \ E^D$ is also a CPO.

Ordering: $f \sqsubseteq g \Rightarrow f(d) \sqsubseteq g(d)$

In order for this to be a CPO, if we take $\bigsqcup f_n$, we should get a continuous function:

$\bigsqcup (\bigsqcup \ f_n) d_m = (\bigsqcup \ f_n)(\bigsqcup \ d_m)$

Is this true? To be continued...