CS611 Lecture 13: The Fixed Point Theorem September 27, 1999

Tony Faradjian

Our goal is to make command denotations *total* functions, so that divergent computations map to $\perp$ ("bottom"). In particular, we need a way to find a fixed point of the function Γ that defines the **while** command, to take care of infinite loops.

We generalize the old fixed point construction (involving **R**) so that we can define a fixed point over complete partial orders (CPO's). Recall that a partial order is a reflexive, transitive, anti-symmetric relation $(P, \sqsubseteq)$ on the set P. If $X \subset P$, then an element p is an *upper bound* of X if $\forall q \in X . q \sqsubseteq p$. An upper bound p of a subset X is a *least* upper bound ("lub") if for any upper bound p' of X, we have $p' \sqsubseteq p$. The least upper bound of X is also denoted by $\sqcup X$. If a partial order contains the lub of any chain $d_0 \sqsubseteq d_1 \sqsubseteq \cdots$ then it is called a *complete* partial order (CPO). The lub of a chain is written as $\bigsqcup_{n \in \omega} d_n$, where ω is the set of whole numbers. Here are some examples:

- $([0,1], \leq)$ is a CPO.
- $([0,1], =)$ is also a CPO, but this is in fact called a *flat* CPO (FCPO) for the simple reason that elements relate only to themselves and not to each other.

If we add a bottom element $\perp$ to a FCPO, then we call it a *pointed* FCPO (PFCPO), due to the appearance of the Hasse diagram. Here are two examples. Let ω denote the natural numbers $\{0, 1, 2, \ldots\}$.

- $(\omega \cup \{\perp\}, =)$ is a PFCPO.
- $(\Sigma \cup \{\perp\}$,"provides less information") is a PFCPO, where Σ is the set of all states (stores) in our programming languages.

We use the symbol $\Sigma_\perp$ in place of $\Sigma \cup \{\perp\}$.

We return to our goal of finding fixed points of functions. Let D denote the set of all functions $\Sigma_\perp \to \Sigma_\perp$. We render D a partial order by defining, for any $f_1, f_2 \in D$:

$$f_1 \sqsubseteq_D f_2 \overset{def}{\Leftrightarrow} \forall \sigma f_1(\sigma) \sqsubseteq_{\Sigma_\perp} f_2(\sigma).$$

We can also define the upper bound of f_0 and f_1, when it exists, as the function returning the upper bound of the two states they produce when given the same state:

$$f_1 \sqcup_D f_2 = \lambda\sigma \,.\, f_1(\sigma) \sqcup_{\Sigma_\perp} f_2(\sigma).$$

In general, we do not require that $f_1 \sqcup_D f_2$ might not exist, because there is no guarantee that an upper bound exists for $f_1(\sigma)$ and $f_2(\sigma)$. However, we only need D to be a complete partial order, so we need to be able to take the upper bound of a chain $f_0 \sqcup_D f_1 \sqcup_D \ldots$. This can be done by defining

$$\bigsqcup_D f_i = \lambda\sigma \,.\, \bigsqcup f_i(\sigma)$$

Because the f_i form a chain, the $f_i(\sigma)$ for any particular σ form a chain in $\Sigma_\perp$.

Finally, we define the bottom of D as the constant function returning the bottom of $\Sigma_\perp$:

$$\perp_D = \lambda\sigma \,.\, \perp$$

Since D has a bottom element, it is a pointed CPO.

Note also that we didn't use any properties that are particular to $\Sigma_\perp$. In fact, if $(P, \sqsubseteq)$ is *any* PFCPO and $\mathcal{F}(P)$ is the set of all functions $P \to P$, we can define the appropriate relation $\sqsubseteq_{\mathcal{F}_P}$, as shown, that makes $(\mathcal{F}(P), \sqsubseteq_{\mathcal{F}_P})$ a PFCPO too.

To see how this generalizes our earlier rule induction, make the correspondences:

$$\begin{array}{ccc} \sqsubseteq & \leftrightarrow & \subseteq \\ \sqcup & \leftrightarrow & \cup \\ \bot & \leftrightarrow & \emptyset \end{array}$$

Thus, the sets being constructed by rule induction are members of a pointed CPO, and all of the mathematical machinery being developed here for pointed CPO's will apply to rule induction as well.

To find the least fixed point of some function over a PFCPO, the function must have two properties:

- *Monotonicity.* A function $\phi : D_1 \to D_2$ between two CPO's D_1 and D_2 is *monotonic* if

$$\forall \delta, \delta' \in D_1 \,.\, \delta \sqsubseteq_{D_1} \delta' \Rightarrow \phi(\delta) \sqsubseteq_{D_2} \phi(\delta')$$

- *Continuity.* $\phi : D_1 \to D_2$ is *continuous* if it is monotonic and

$$\phi(\bigsqcup_i x_i) = \bigsqcup_i \phi(x_i)$$

This brings us to our main result:

Fixed Point Theorem. Let Γ be a continuous function from a pointed CPO to a pointed CPO, and define $\mathit{fix}(\Gamma) = \bigsqcup_{n\in\omega} \Gamma^n(\bot)$. That is, $\mathit{fix}(\Gamma)$ is the least upper bound of the set $\{\Gamma^n \mid n \in \omega\}$. Because of the monotonicity of Γ, this upper bound is well defined. Then $\mathit{fix}(\Gamma)$ is the least fixed point of Γ.

Proof. First we show $\Gamma(\mathit{fix}(\Gamma)) = \mathit{fix}(\Gamma)$:

$$\begin{aligned}
\Gamma(\mathit{fix}(\Gamma)) &= \Gamma(\bigsqcup_{n\in\omega} \Gamma^n(\bot)) \\
&= \bigsqcup_{n\in\omega} \Gamma(\Gamma^n(\bot)) \qquad \text{(continuity)} \\
&= \bigsqcup_{n\in\omega} \Gamma^{n+1}(\bot)) \\
&= \{\bot\} \sqcup \{\bigsqcup_{n\in\omega} \Gamma^{n+1}(\bot)\} \qquad \text{(defn. of } \bot) \\
&= \{\Gamma^0(\bot)\} \sqcup \{\bigsqcup_{n\in\omega} \Gamma^{n+1}(\bot)\} \\
&= \bigsqcup_{n\in\omega} \Gamma^n(\bot) \\
&= \mathit{fix}(\Gamma)
\end{aligned}$$

So $\mathit{fix}(\Gamma)$ is indeed a fixed point of Γ. To show that it is the *least* fixed point, suppose that d is a fixed point, i.e., $\Gamma(d) = d$. Since $\bot \sqsubseteq d$, we have, by monotonicity, $\Gamma(\bot) \sqsubseteq \Gamma(d) = d$. By induction we can show $\forall n \in \omega \,.\, \Gamma^n(\bot) \sqsubseteq d$. That is, d is an upper bound of $\{\Gamma^n \mid n \in \omega\}$. But $\mathit{fix}(\Gamma)$ is the least upper bound of this set, so $\mathit{fix}(\Gamma) \sqsubseteq d$. Because this is true for all fixed points d, we see that $\mathit{fix}(\Gamma)$ is the least fixed point of Γ.

Our denotational semantics are now on firm ground, and with the help of $\bot$, we can express our denotation for IMP and other languages as total functions.

Addendum

We expressed the denotation for **while** as the fixed point of the following function:

$$\Gamma = \lambda f : \Sigma_\bot \to \Sigma_\bot \,.\, \lambda\sigma : \Sigma_\bot \,.\, \textbf{if } \mathcal{B}[\![b]\!]\sigma \textbf{ then } f(\mathcal{C}[\![c]\!]\sigma) \textbf{ else } \sigma$$

To get an intuition for how the fixed point operator works, consider the following sequence:

$$
\begin{array}{rcl}
\Gamma(\bot) & = & \lambda\sigma : \Sigma_\bot \,.\, \textbf{if}\ \mathcal{B}[\![b]\!]\sigma\ \textbf{then}\ \bot\ \textbf{else}\ \sigma \\
\Gamma(\Gamma(\bot)) & = & \lambda\sigma : \Sigma_\bot \,.\, \textbf{if}\ \mathcal{B}[\![b]\!]\sigma\ \textbf{then}\ (\textbf{let}\ \sigma' = \mathcal{C}[\![c]\!]\sigma\ \textbf{in if}\ \mathcal{B}[\![b]\!]\sigma'\ \textbf{then}\ \bot\ \textbf{else}\ \sigma')\ \textbf{else}\ \sigma
\end{array}
$$

Note that $\Gamma(\bot)$ works as a denotation for while loops whose condition immediately fails: loops that run 0 times. $\Gamma(\Gamma(\bot))$ works as a denotation for while loops that run 0 or 1 times. $\Gamma^n(\bot)$ works as a denotation for while loops that run up to n times. The LUB of all such functions is the denotation for while loops that run an unbounded number of times.