1 Denotational Semantics for FL

So far the most interesting thing we have given a denotational semantics for is the while loop. However, we now have enough machinery to capture higher-order constructs such as mutually recursive functions. We show how to give a semantics for a version of the FL language.

1.1 Syntax

We will work with a simplified version of FL similar to the λ-lifted version from Assignment 3.

$$\begin{array}{rcl} p & ::= & \mathsf{letrec}\ d\ \mathsf{in}\ e \\ d & ::= & f(x_1, \ldots, x_n) = e \ \mid\ f(x_1, \ldots, x_n) = e\ \mathsf{and}\ d \\ e & ::= & n \ \mid\ x \ \mid\ \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \ \mid\ f(e_1, \ldots, e_n) \ \mid\ \mathsf{ifp}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \\ & & \mid\ e_1 + e_2 \ \mid\ \ldots \text{ (other arithmetic operators)} \end{array}$$

The syntactic constructs defined by d are mutually recursive function declarations. These occur at the outermost level only. The conditional test ifp-then-else expects a number instead of a Boolean for its first argument, and the test succeeds if that number is positive.

For example,

$$\begin{array}{l} \mathsf{letrec}\ f_1(n, m) = \mathsf{ifp}\ m^2 - n\ \mathsf{then}\ 1\ \mathsf{else}\ (n\ \mathsf{mod}\ m) \cdot f_1(n, m+1) \\ \mathsf{and}\ f_2(n) = \mathsf{ifp}\ f_1(n, 2)\ \mathsf{then}\ n\ \mathsf{else}\ f_2(n+1) \\ \mathsf{in}\ f_2(1000) \end{array}$$

In this program, $f_2(n)$ finds the first prime number $p \geq n$. The value of $n\ \mathsf{mod}\ m$ is positive iff m does not divide n.

1.2 CBV Denotational Semantics for REC

We will interpret an expression e as a function is $[\![e]\!] \in \mathit{FEnv} \to \mathit{Env} \to \mathbb{Z}_\perp$, where Env and FEnv denote the sets of *variable environments* and *function environments*, respectively.

$$\rho \in \mathit{Env} = \mathit{Var} \to \mathbb{Z} \qquad\qquad \varphi \in \mathit{FEnv} = (\mathbb{Z}^{n_1} \to \mathbb{Z}_\perp) \times \cdots \times (\mathbb{Z}^{n_k} \to \mathbb{Z}_\perp)$$

Here Var and FVar are disjoint countable sets of variables, $\mathbb{Z}$ is the set of integers, and $\mathbb{Z}^n = \underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_{n \text{ times}}$.

$$
\begin{aligned}
[\![n]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} n \\
[\![x]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} \rho(x) \\
[\![e_1 + e_2]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} [\![e_1]\!]\,\varphi\,\rho \;\; +^\dagger \;\; [\![e_2]\!]\,\varphi\,\rho \quad \text{(similarly for other arithmetic operators)} \\
[\![\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} \mathit{let}\ v \in \mathbb{Z} = [\![e_1]\!]\,\varphi\,\rho\ \mathit{in} \\
&\phantom{\stackrel{\triangle}{=}}\ [\![e_2]\!]\,\varphi\,\rho[v/x] \\
[\![\mathsf{ifp}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} \mathit{let}\ v_0 \in \mathbb{Z} = [\![e_0]\!]\,\varphi\,\rho\ \mathit{in} \\
&\phantom{\stackrel{\triangle}{=}}\ \mathit{if}\ v_0 > 0\ \mathit{then}\ [\![e_1]\!]\,\varphi\,\rho\ \mathit{else}\ [\![e_2]\!]\,\varphi\,\rho \\
[\![f_i(e_1, \ldots, e_n)]\!]\,\varphi\,\rho &\stackrel{\triangle}{=} \mathit{let}\ v_1 \in \mathbb{Z} = [\![e_1]\!]\,\varphi\,\rho\ \mathit{in} \\
&\phantom{\stackrel{\triangle}{=}}\ \vdots \\
&\phantom{\stackrel{\triangle}{=}}\ \mathit{let}\ v_n \in \mathbb{Z} = [\![e_n]\!]\,\varphi\,\rho\ \mathit{in} \\
&\phantom{\stackrel{\triangle}{=}}\ (\pi_i\,\varphi)(v_1, \ldots, v_n)
\end{aligned}
$$

In the definition of $[\![e_1 + e_2]\!]\,\varphi\,\rho$, the symbol $+^\dagger$ refers to the lifted version of addition on $\mathbb{Z}$. This function takes the value $\perp$ if either of its arguments is $\perp$, otherwise returns the sum of its arguments.

The meaning of a program $\mathsf{letrec}\ d\ \mathsf{in}\ e$ is

$$[\![\mathsf{letrec}\ d\ \mathsf{in}\ e]\!] \;\stackrel{\triangle}{=}\; [\![e]\!]\,\varphi\,\rho_0,$$

where ρ_0 is some initial environment containing default values for the variables (say 0), and if the function declarations d are

$$f_1(x_1, \ldots, x_{n_1}) = e_1\ \mathsf{and}\ \ldots\ \mathsf{and}\ f_k(x_1, \ldots, x_{n_k}) = e_k,$$

then

$$
\begin{aligned}
\varphi \quad = \quad \mathit{fix}\ \ \lambda\psi \in \mathit{FEnv}.\,(&\lambda v_1 \in \mathbb{Z}, \ldots, v_{n_1} \in \mathbb{Z}.\,[\![e_1]\!]\,\psi\,\rho_0[v_1/x_1, \ldots, v_{n_1}/x_{n_1}], \\
&\vdots \\
&\lambda v_1 \in \mathbb{Z}, \ldots, v_{n_k} \in \mathbb{Z}.\,[\![e_k]\!]\,\psi\,\rho_0[v_1/x_1, \ldots, v_{n_k}/x_{n_k}]),
\end{aligned}
$$

or more accurately,

$$
\begin{aligned}
\varphi \quad = \quad \mathit{fix}\ \ \lambda\psi \in \mathit{FEnv}.\,(&\lambda v \in \mathbb{Z}^{n_1}.\,[\![e_1]\!]\,\psi\,\rho_0[\pi_1(v)/x_1, \ldots, \pi_{n_1}(v)/x_{n_1}], \\
&\vdots \\
&\lambda v \in \mathbb{Z}^{n_k}.\,[\![e_k]\!]\,\psi\,\rho_0[\pi_1(v)/x_1, \ldots, \pi_{n_k}(v)/x_{n_k}]).
\end{aligned}
$$

For this fixpoint to exist, we need to know that FEnv is a pointed CPO and that the function $\mathit{FEnv} \to \mathit{FEnv}$ to which we are applying fix is continuous. The domain FEnv is a product, and a product is a pointed CPO when each factor is a pointed CPO. Each factor $\mathbb{Z}^{n_i} \to \mathbb{Z}_\perp$ is a pointed CPO, since a function is a pointed CPO when the codomain of that function is a pointed CPO, and $\mathbb{Z}_\perp$ is a pointed CPO. Therefore, FEnv is a pointed CPO.

The function $\tau : \mathit{FEnv} \to \mathit{FEnv}$ to which we are applying fix is continuous, because it can be written using the metalanguage. Here is the argument. We illustrate with $k = 2$ and $n_1 = n_2 = 1$ for simplicity, thus we assume the declaration d is

$$f_1(x) = e_1\ \mathsf{and}\ f_2(x) = e_2.$$

Then

$$\varphi \;=\; \mathit{fix}\ \lambda\psi \in \mathit{FEnv}\,.\,(\lambda v \in \mathbb{Z}\,.\,[\![e_1]\!]\,\psi\,\rho_0[v/x],\ \lambda v \in \mathbb{Z}\,.\,[\![e_2]\!]\,\psi\,\rho_0[v/x]).$$

This gives the least fixpoint of the operator

$$\tau \;=\; \lambda\psi \in \mathit{FEnv}\,.\,(\lambda v \in \mathbb{Z}\,.\,[\![e_1]\!]\,\psi\,\rho_0[v/x],\ \lambda v \in \mathbb{Z}\,.\,[\![e_2]\!]\,\psi\,\rho_0[v/x]),$$

provided we can show that τ is continuous. We can write

$$\begin{aligned}
\tau \;&=\; \lambda\psi \in \mathit{FEnv}\,.\,(\lambda v \in \mathbb{Z}\,.\,[\![e_1]\!]\,\psi\,\rho_0[v/x],\ \lambda v \in \mathbb{Z}\,.\,[\![e_2]\!]\,\psi\,\rho_0[v/x]) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,(\tau_1(\psi),\ \tau_2(\psi)) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\langle\tau_1,\ \tau_2\rangle\,(\psi) \\
&=\; \langle\tau_1,\ \tau_2\rangle,
\end{aligned}$$

where $\tau_i : \mathit{FEnv} \to \mathit{FEnv}$ is

$$\tau_i \;=\; \lambda\psi \in \mathit{FEnv}\,.\,\lambda v \in \mathbb{Z}\,.\,[\![e_i]\!]\,\psi\,\rho_0[v/x].$$

Because $\langle\tau_1,\ \tau_2\rangle$ is continuous iff τ_1 and τ_2 are, it suffices to show that each τ_i is continuous. Now we can write τ_i in our metalanguage.

$$\begin{aligned}
\tau_i \;&=\; \lambda\psi \in \mathit{FEnv}\,.\,\lambda v \in \mathbb{Z}\,.\,[\![e_i]\!]\,\psi\,\rho_0[v/x] \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\lambda v \in \mathbb{Z}\,.\,[\![e_i]\!]\,\psi\,(\mathit{subst}\,\rho_0\,x\,v) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\lambda v \in \mathbb{Z}\,.\,([\![e_i]\!]\,\psi)\,((\mathit{subst}\,\rho_0\,x)\,v) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\lambda v \in \mathbb{Z}\,.\,(([\![e_i]\!]\,\psi)\circ(\mathit{subst}\,\rho_0\,x))\,v \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,(([\![e_i]\!]\,\psi)\circ(\mathit{subst}\,\rho_0\,x)) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\mathit{compose}\,([\![e_i]\!]\,\psi,\ \mathit{subst}\,\rho_0\,x) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\mathit{compose}\,([\![e_i]\!]\,\psi,\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\,\psi) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,\mathit{compose}\,(\langle[\![e_i]\!],\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\rangle\,\psi) \\
&=\; \lambda\psi \in \mathit{FEnv}\,.\,(\mathit{compose}\circ\langle[\![e_i]\!],\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\rangle)\,\psi \\
&=\; \mathit{compose}\circ\langle[\![e_i]\!],\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\rangle.
\end{aligned}$$

Now we can argue that τ_i is continuous. The composition of two continuous functions is continuous, so it suffices to know that *compose* and $\langle[\![e_i]\!],\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\rangle$ are continuous. We argued last time that *compose* is continuous. To show $\langle[\![e_i]\!],\ \mathit{const}\,(\mathit{subst}\,\rho_0\,x)\rangle$ is continuous as a function, it suffices to show that both $[\![e_i]\!]$ and $\mathit{const}\,(\mathit{subst}\,\rho_0\,x)$ are continuous as functions. The former is continuous by the induction hypothesis (structural induction on e). The latter is a constant function on a discrete domain and is therefore continuous.

1.3 CBN Denotational Semantics

The denotational semantics for CBN is the same as for CBV with two exceptions:

$$\begin{aligned}
[\![\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2]\!]\,\varphi\,\rho \;&\triangleq\; [\![e_2]\!]\,\varphi\,\rho[[\![e_1]\!]\,\varphi\,\rho/x] \\
[\![f_i(e_1,\ldots,e_n)]\!]\,\varphi\,\rho \;&\triangleq\; (\pi_i\,\varphi)([\![e_1]\!]\,\varphi\,\rho,\ldots,[\![e_n]\!]\,\varphi\,\rho).
\end{aligned}$$

We must extend environments and function environments:

$$\mathit{Env} = \mathit{Var} \to \mathbb{Z}_\perp \qquad\qquad \mathit{FEnv} = (\mathbb{Z}_\perp^{n_1} \to \mathbb{Z}_\perp) \times \cdots \times (\mathbb{Z}_\perp^{n_k} \to \mathbb{Z}_\perp).$$