Advanced Progamming Languages	Lecture 23
CS 6110 Spring 2015	Wed. March 18, 2015

Lecture 23

Topics

1. If you can attend Voevodsky's lecture at 4:30 pm in Malott 406 it will be relevant to discussion in Friday's lecture.
2. To reason about partial functions we need a new equality relation, $t_1 \simeq t_2$, and a new induction principle, *fixed point induction*. We first discuss the fixed points of recursive functionals.
3. We will study Kleene's Recursion Theorem as a basis for the induction.
4. We will study a generalization of Kleene's theorem in a classical setting.

Kleene Equality $\varphi(x) \simeq \psi(x)$ – converge or diverge together and if they converge, then they converge to the same value.

Functionals and fixed points

Consider the recursive function on $\mathbb{N}$:
$f(x, y) = \text{if } x = y \text{ then } y + 1 \text{ else } f(x, f(x - 1, y + 1))$.

Write this in terms of the functional F:
$\lambda f.\lambda x, y.$ if $x = y$ then $y + 1$ else $f(x, f(x - 1, y + 1))$.
$F(f) = \lambda x.\lambda y.$

$$\text{Let}\quad \begin{array}{llll} f_1(x,y) = & \text{if } x = y & \text{then } y+1 & \text{else } x+1 \\ f_2(x,y) = & \text{if } x \geq y & \text{then } x+1 & \text{else } y-1 \\ f_3(x,y) = & \text{if } x \geq y \;\&\; even(x-y) & \text{then } x+1 & \text{else } \perp \text{ (where } \perp \text{ is the diverging element)} \end{array}$$

Notice that for $i = 1, 2, 3$

$F(f_i)(x, y) \simeq$ if $x = y$ then $y + 1$ else $f_i(x, f_i(x - 1, y + 1))$

Notice that f_3 is a fixed point of F, i.e.

$$
\begin{array}{rcl}
F(f_3) & \simeq & f_3 \\
F(f_3) & = & \lambda x, y.\ \text{if } x = y \text{ then } y+1 \text{ else } f_3(x, f_3(x-1, y+1)) \\
& = & \lambda x, y.\ \text{if } x = y \text{ then } y+1 \text{ else if } x \geq y \ \&\ even(x-y) \text{ then } x+1 \text{ else } \perp \\
& = & \text{if } x \geq y \ \&\ even(x-y) \text{ then } x+1 \text{ else } \perp \\
& & \text{if } x = y \text{ then } even(x-y) \\
& & \qquad\qquad \text{hence } x+1 \text{ (same value as } f_3) \\
& & \text{if } x \neq y \text{ then if } x > y \\
& & \qquad\qquad \text{so if } even(x-y) \text{ then } x+1 \text{ (same value as } f_3) \\
& & \qquad\qquad \text{if } x < y \text{ then } \perp \text{ (same value as } f_3)
\end{array}
$$

But also notice:

$$
\begin{array}{rcl}
F(f_1) & \simeq & f_1 \text{ since} \\
F(f_1) & = & \lambda x, y.\ \text{if } x = y \text{ then } y+1 \\
& & \text{so if } x = y \text{ this is the same value as } f_1 \\
& & \text{if } x \neq y \text{ then } f_1(x, f_1(x-1, y+1)) \text{ and } f_1(x, ...) \text{ is } x+1 \text{, this is the same value as } f_1
\end{array}
$$

Notice $f_3(x, y) \sqsubseteq f_1(x, y)$.

Kleene's Recursion Theorem *For all recursive functionals $F(\varphi) \simeq \varphi'$ there is a partial recursive function φ such that $F(\varphi) \simeq \varphi$, and for all φ' such that $F(\varphi') \simeq \varphi'$, $\varphi \sqsubseteq \varphi'$.*

Proof sketch:

Let $\varphi_0 =$ the totally undefined patial function on $\mathbb{N}$. Define the sequence $\varphi_{i+1} \simeq F(\varphi_i)$, note $\varphi_i \sqsubseteq \varphi_j,\ i < j$.

Define φ_ω as the *limit* of this sequence. $\varphi_\omega(x)$ is defined if there is an i such that $\varphi_i(x) \downarrow$. To compute $\varphi_\omega(x)$ we compute the sequence $\varphi_1(x), \varphi_2(x), ..., \varphi_n(x)$ and give a value if one of the $\varphi_j(x)$ is defined.

We need to establish two claims:

(a) $F(\varphi_\omega)(x) \simeq \varphi_\omega(x)$ for all x

(b) If $F(\varphi)(x) \simeq \varphi(x)$ for all x, then $\varphi_\omega(x) \sqsubseteq \varphi(x)$

Why are these intuitively true?

(a) $F(\varphi_\omega)(x) \simeq \varphi_\omega(x)$ because if $F(\varphi_\omega)(x) = k$ for some number k then at some least i, $F(\varphi_i(x)) = k$. $F(\varphi_\omega)(x)$ will give the same value because it accesses the same data.

(b) If $F(\varphi)(x) \simeq \varphi(x)$, then $\varphi_\omega(x) = \varphi(x)$ because $\varphi_0 \sqsubseteq \varphi,\ \varphi_1 \sqsubseteq \varphi, ..., \varphi_i \sqsubseteq \varphi$, and this sequence has the least amount of data needed to compute the fixed point.

∎