Comp 311
Principles of Programming Languages
Lecture 12
The Semantics of Recursion III & Loose Ends

Corky Cartwright

September 24, 2010

Call-by-value Fixed-Point Operators

Given a recursive definition in a call-by-value language

```
f = E_f
```

where E_f is an expression constructed from constants in the based data domain D, operations (continuous functions) on D, and `f`, what does it mean?

- Example: let D be the domain of Scheme values. Then

```
fact =
    (lambda (n) (if (zero? n) 1 (* n (fact (- n 1)))))
```

is such a definition.

In a call-by-name language, the meaning is

```
Y (lambda (f) E_f)
```

We need to define Y_v

Solutions to Recursion Equations

Key trick: use η-conversion to delay evaluation.

In the mathematical literature on the λ-calculus, η-conversion is often assumed as an axiom. In models of the λ-calculus, it is typically required to hold.

Definition: η-conversion is the following equation:

```
M = λ x . Mx
```

where `x` is not free in `M`

What Is the Code for $\mathbf{Y_v}$?

λ`F.`$($$\lambda$`x.`$\lambda$`y. F(x x)y)(`$\lambda$`x.`$\lambda$`y. F(x x)y)`

- Does this work for Scheme (or Java with an appropriate encoding of functions as anonymous inner classes? Yes!
- Let `G` be some functional `G =`λ`f.`λ`n.`M_f like **FACT** for a recursive *function* definition. `G` is a value. Then

  Y_v`G` → `(`λ`x.`λ`y. G(x x)y)(`λ`x.`λ`y. G(x x)y)` →

  λ`y. G ((`λ`x.`λ`z. G(x x)z)(`λ`x.`λ`z. G(x x)z)) y`

  is a value.
- Hence, `G(`Y_v `G)` → `(`λ`n.`M_f`) [f:=`Y_v `G]` is a value.
- Moreover,

  Y_v`G = `λ`y. G ((`λ`x.`λ`z. G(x x)z)(`λ`x.`λ`z. G(x x)z)) y =`

  λ`y. G (`Y_v`G) y`

  which is the η-conversion of `G(`Y_v`G)`

Loose Ends

- Meta-errors
- Read the notes!
- rec-let (in notes)