Recitation 18: The Environment Model

Substitution

$\langle env, \texttt{let } x = 5 \texttt{ in } e \rangle$

$e\{5/x\}$

Rules

$e \longrightarrow e'$

$e \longrightarrow^* e'$

$e \Longrightarrow v$

$x\{v/x\} = v$

$(\texttt{let } y = e_1 \texttt{ in } e_2)\{v/x\}$

Environment

env: partial function from variables to values

Rules

$\langle env, e \rangle \longrightarrow \langle env', e' \rangle$

$\langle env, e \rangle \longrightarrow^* \langle env', e' \rangle$

$\langle env, e \rangle \Rightarrow v$

$\langle env, x \rangle \Rightarrow env(x)$

if $\langle env, e_1 \rangle = v_1$
and $\langle env[x \rightarrow v_1], e_2 \rangle \Rightarrow v$
$\langle env, \texttt{let } x = e_1 \texttt{ in } e_2 \rangle \Rightarrow v$

Env Model Rules

Values

$\langle env, v \rangle \Rightarrow v$

Bops

if $\langle env, e_1 \rangle \Rightarrow v_1$
and $\langle env, e_2 \rangle \Rightarrow v_2$
and "v_1 bop v_2" is v
$\langle env, e_1 \text{ bop } e_2 \rangle \Rightarrow v$

Conditionals

if $\langle env, e_1 \rangle \Rightarrow$ true
and $\langle env, e_2 \rangle \Rightarrow v_2$

$\langle env, \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \rangle \Rightarrow v_2$

Pairs

if $\langle env, e_1 \rangle \Rightarrow v_1$
and $\langle env, e_2 \rangle \Rightarrow v_2$

$\langle env, (e_1, e_2) \rangle \Rightarrow (v_1, v_2)$

if $\langle env, e \rangle \Rightarrow (v_1, v_2)$
$\langle env, \text{fst } e \rangle \Rightarrow v_1$

Constructors

if $\langle env, e \rangle \Rightarrow v$
$\langle env, \text{Left } e \rangle \Rightarrow \text{Left } v$

Pattern Matching

if $\langle env, e \rangle \Rightarrow$ Left v
and $\langle env[x_1 \to v], e_1 \rangle \Rightarrow v'$
$\langle env,$ match e with Left $x_1 \to e_1$; Right $x_2 \to e_2 \rangle \Rightarrow v'$

Anonymous Functions

Wrong!

$\langle env,$ fun $x \to e \rangle \Rightarrow$ fun $x \to e$

if $\langle env, e_1 \rangle \Rightarrow$ fun $x \to e$
and $\langle env, e_2 \rangle \Rightarrow v_{arg}$
and $\langle env[x \to v_{arg}], e \rangle \Rightarrow v$
$\langle env, e_1\ e_2 \rangle \Rightarrow v$

Why it's wrong

Environments

```
let x = 1 in          ↙ FV
let f = fun y → x in
let x = 2 in
f 0
```

{}
{x : 1}
{x : 1, f : fun y → x}
{x : 2, f : fun y → x}

f 0 becomes (fun y → 2) 0 = 2

Ocaml: 1

Scoping

Lexical
- see where all vars are bound
- breaks modularity

Dynamic
Wrong!

Closures!

pair of code and environment

(fun $x \to e$, env)

Lexical scoping (Correct!) rules

$\langle env, \text{fun } x \to e \rangle \Rightarrow (|\ \text{fun } x \to e,\ env\ |)$

if $\langle env, e_1 \rangle \Rightarrow (|\ \text{fun } x \to e,\ defenv\ |)$
and $\langle env, e_2 \rangle \Rightarrow v_{arg}$
and $\langle defenv[x \to v_{arg}],\ e \rangle \Rightarrow v$
$\langle env, e_1\ e_2 \rangle \Rightarrow v$

Environment

```
let x = 1 in                {}
let f = fun y -> x in       {x : 1}
                            {x : 1, f : (| fun y -> x, {x : 1} |)}
let x = 2 in                {x : 2, f : (| " " |)}
f 0
```

$\langle env, f \rangle \Rightarrow env(f) = (|\ \text{fun } y \to x,\ \underbrace{\{x : 1\}}_{defenv}\ |)$
$\langle env, 0 \rangle \Rightarrow 0$

$\langle \{x : 1, y : 0\}, x \rangle \Rightarrow 1$