TRIVIA: WHAT ARE:

52 2.30
37 3.35
19 11.15
6 7.30

PS #2 DUE 9/22 11:59 PM
QUIZ # ON 9/22 10:10 AM

Q1 BACK MONDAY
9/27 GUEST LECTURE WALKER WHITE
- NO RD2 O.H.

SUBSTITUTION MODEL
PRECISE DESCRIPTION OF OCAML

let rec evil (f1, f2, n) =
let f(x) = 10 + n in
if n=1 then f(0) + f1(0) + f2(0)
else evil (f, f1, n-1)
and dummy(x) = 1000
in evil (dummy, dummy, 3)

Model of Evaluation

Type at OCaml: term or declaration
Evaluate a term to produce a value

Value is a subset of terms needing no further evaluation
Values: constants, tuples of values, functions, etc

\[(1 + 2) * (3 + 4)\] \[\frac{((1 + 2) * 3) \times 4 * 5}{3 * 7}\]
\[3 + 5\] \[a1\]

Rewrite expression to another one continue until value OR [exception, inf/loop, etc]

Which reduction does occur and do?
Leftmost reduction first

Rules:

**IF**
- if true then e1 else e2 \(\rightarrow\) e1
- if false then e1 else e2 \(\rightarrow\) e2

**IF**
- if e0 then e1 else e2

**IF**
- if 2 = 3 then "hello" else "good" "bye"
- false "good" "bye"

**LET**
- let id = e1 in e2
- Go through e2 and replace id by e1 substitute
  - Evaluate e1 \(\rightarrow\) v1
  - "Substitute" v1 for id in e2
  - Evaluate that
let \( x = 1 + 4 \) in \( x \times 3 \) \( \Rightarrow \)
let \( x = 5 \) in \( x \times 3 \) \( \Rightarrow \)
\[ 5 \times 3 \Rightarrow 15 \]
only substitute once we have a value
EAGER Ocaml eagerly evaluates the expression binding for a variable

\((\text{Fun} \ x \rightarrow x \times 3) \ (1+4)\) LAZY, Haskell

Ocaml is EAGER
Java, C, etc Do lazy in ocaml later on
let \( X = \nu \) in \( e \) with \( x \) replaced by \( \nu \)

\( e \) expr (arbitrary term)

\( \nu \) value

\( \nu ::= e \ | \ \text{let} \ f(x) = x \ \text{in} \ e \ | \ (\nu_1, \ldots, \nu_n) \ | \ \text{fun} \ p \rightarrow e \)

\( \text{Cons}(1, \text{Nil}) \ | \ \text{Nil} \)

\( \text{Cons}(2+3, \text{Nil}) \)

\( X(\nu) \) is a value when

* \( X \) is a type constructor
* \( \nu \) a value

\( (f(3), 24, 17) \)

\( (12, 24, 17) \)

fun \( y \rightarrow \text{fun} \ z \rightarrow \)

fun \( x \rightarrow 2x + 3 \)

2 + 3 \times 3

Values self-evaluate

Substitution

\[
\begin{align*}
\text{let} \ x &= 11 \ \text{in} & \text{bound occurrence} \\
\text{let} \ f(x) &= x \ \text{in} & \text{binding} \\
\text{let} \ y &= x + 1 \ \text{in} & \text{fun} \ (a \ : \ \text{string}) \rightarrow "x \ marks \ the \ spot" \\
\text{let} \ x &= \nu \ \text{in} \ e & \text{replace unbound occurrences of} \ x \ \text{by} \ \nu \ \text{in} \ e \\
\end{align*}
\]
X 2/x3 => 2
X 2/y3 => X
(Fun y -> X) "hi" / x3 => Fun y ~> "hi"

Let x = "hi"
In Fun y -> x
& 'a' -> string

E \{ V / id3
* Expression e with
unbound occurrences of
the variable id replaced
by the value V

f(x) & Fun y -> y/ F3 => (Fun y = y) (x)

Unbound? "Free"

1. let x = e in e1
X is bound in e1
2. Fun (Z) -> e
Z is bound in e