

Week 38

Sigurt Bladt Dinesen
sidi@itu.dk

Jens Egholm Pedersen
jegp@itu.dk

DRAFT of December 27, 2012

Exercise 6.1 (PLC)

Is the result of the third one as expected?

```
1 let add x = let f y = x+y in f end
2 in let addtwo = add 2
3   in let x = 77 in addtwo 5 end
4   end
5 end
```

We would expect *add x* to return a function *f of y* that in turn returns $x+y$. We would expect *addtwo* to increment its (integer) input by 2, and the entire expression to return 7. The *let x = 77* in line 3 means nothing. *x* from line 1 is not overwritten, only "shadowed".

Explain the result of the last one

```
1 let add x = let f y = x+y in f end
2 in add 2 end
```

The result is a function (described as a Closure in our object language) of type $(int \rightarrow int)$. As in the previous expression, *add x* returns a function *f of y* that returns $x + y$. The result of the entire expression is then the function of an integer, that returns the sum of 2 and that integer. In our object language, that is the closure

$$("f", "y", Prim("+", Var"x", Var"y"))$$

with an environment

$$\rho[x \mapsto 2]$$

(feel free to comment the notation.)

The environment also contains the closure that defines the *add x* function, though we will not need it to use the function returned by the expression.

Exercise 6.4 (PLC)

(i)

```
1 let f x = 1
2   in f f end
```

Since we define $f\ f$ to be of type $t_x \rightarrow int$ and we define f to be of type t_x (since it is the argument to f), we have to define the argument for f as a polymorphic type or risk infinity.

$$\begin{array}{c}
\text{p8} \frac{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow int] \vdash 1 : int}{\text{p9} \frac{\frac{\text{r3} \frac{\rho(f) = \forall \alpha_1 \dots \alpha_n. t_x}{\rho[f \mapsto \forall \alpha_1 \dots \alpha_n. t_x \rightarrow int] \vdash f : t_x \rightarrow int} \quad \rho \mapsto f : t_x}{\rho[f \mapsto \forall \alpha_1 \dots \alpha_n. t_x \rightarrow int] \vdash ff : t} \quad \alpha_1 \dots \alpha_n \text{ not free in } \rho} \\
\rho \vdash \text{let } f\ x = 1 \text{ in } f\ f \text{ end} : t
\end{array}$$

(ii)

```

1 let f x = if x < 10 then 42 else f(x+1)
2   in f 20 end

```

f should not be polymorphic since x is proven to be an int (p5) and since we know that *then* and *else* returns the same type.

$$\begin{array}{c}
\text{p5} \quad \frac{\rho \mapsto x : \text{int}}{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow t_r] \vdash x < 10 : \text{bool}} \quad \text{p1} \quad \frac{\rho \mapsto 10 : \text{int}}{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow t_r] \vdash t_x \rightarrow t_r} \\
\text{p7} \quad \frac{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow t_r] \vdash t_x \rightarrow t_r}{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow t_r] \vdash \text{if } x < 10 \text{ then } 42 \text{ else } f(x+1) : \text{int}} \quad \text{p9} \quad \frac{\rho \vdash x+1 : \text{int} \quad \rho \vdash f(x+1) : \text{int}}{\rho \vdash f(x+1) : \text{int}} \\
\text{p3} \quad \frac{\rho \vdash x+1 : \text{int} \quad \rho \vdash f(x+1) : \text{int}}{\rho[x \mapsto \forall \alpha_1 \dots \alpha_n] \vdash f : \text{int} \rightarrow \text{int}} \quad \text{p1} \quad \frac{\rho \vdash 20 : \text{int}}{\rho[x \mapsto \forall \alpha_1 \dots \alpha_n] \vdash f \ 20 \text{ end} : \text{int}} \\
\text{p8} \quad \frac{\rho \vdash \text{let } f \ x = \text{if } x < 10 \text{ then } 42 \text{ else } f(x+1) \text{ in } f \ 20 \text{ end} : t}{\rho \vdash \text{let } f \ x = \text{if } x < 10 \text{ then } 42 \text{ else } f(x+1) \text{ in } f \ 20 \text{ end} : t} \quad \alpha_1 \dots \alpha_n \text{ not free in } \rho
\end{array}$$