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

```
let add x = let f y = x+y in f end
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)

```
let f x = 1
    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.

$$
\dfrac{\rho[x \mapsto t_x, f \mapsto t_x \rightarrow int] \vdash 1 : int \qquad \dfrac{\dfrac{\rho(f) = \forall \alpha_1 ... \alpha_n . t_x}{\rho[f \mapsto \forall \alpha_1 ... \alpha_n . t_x \rightarrow int] \vdash f : t_x \rightarrow int}\,\text{P3} \qquad \rho \mapsto f : t_x}{\rho[f \mapsto \forall \alpha_1 ... \alpha_n . t_x \rightarrow int] \vdash f f : t}\,\text{P9} \qquad \alpha_1 ... \alpha_n \mathit{not\ free\ in}\ \rho}{\rho \vdash \mathit{let}\ f\ x = 1\ \mathit{in}\ f\ f\ \mathit{end} : t}\,\text{P8}
$$

(ii)

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

One might think that the expression should be polymorphic since the right hand side of the if-expression never terminates. However the rule 7 proves that the right hand side has the same type as the left hand side of the if-then-else. Hence the if-expression has type int.

$$
\text{P8}\,\dfrac{
\text{P7}\,\dfrac{
\text{P5}\,\dfrac{\rho \mapsto x : int \qquad \text{P1}\,\dfrac{}{\rho \mapsto 10 : int}}{\rho[x \mapsto t_x, f \mapsto t_x \to t_r] \vdash x < 10 : bool}
\qquad \rho\ 42 : int \qquad
\text{P9}\,\dfrac{\rho \vdash x + 1 : int}{\rho \vdash f(x+1) : int}
}{\rho[x \mapsto t_x, f \mapsto t_x \to t_r] \vdash if\ x\ <\ 10\ then\ 42\ else\ f(x+1) : int}
\qquad
\text{P9}\,\dfrac{
\text{P3}\,\dfrac{\rho(f) = \forall \alpha_1 ... \alpha_n . int}{\rho[x \mapsto \forall \alpha_1 ... \alpha_n] \vdash f : int \to int}
\qquad \text{P1}\,\dfrac{}{\rho \vdash 20 : int}
}{\rho[x \mapsto \forall \alpha_1 ... \alpha_n] \vdash f\ 20\ end : int}
\qquad \alpha_1 ... \alpha_n\ not\ free\ in\ \rho
}{\rho \vdash let\ f\ x = if\ x <\ 10\ then\ 42\ else\ f(x+1)\ in\ f\ 20\ end : int}
$$