Assignment 4
Semantics, WS 2009/10

Prof. Dr. Gert Smolka, Dr. Jan Schwinghammer, Christian Doczkal
www.ps.uni-sb.de/courses/sem-ws09/

Hand in by 11.59am, Tuesday, November 17

Send your solutions to Exercises 4.2 and 4.3 in a file named `lastname.sml` to `doczkal@ps.uni-sb.de`, and make sure that the entire file compiles without errors. Use the deep implementation of PCF$^-$ in Standard ML from Exercise 3.3:

```
datatype ty = Nat | P of ty * ty
type var = string
datatype ter = V of var | A of ter * ter | L of var * ty * ter
             | O | S of ter | C of ter * ter * ter | F of ter
```

Exercise 4.1 (Contexts and Redexes) Consider the following PCF terms.

a) $\mathit{pred}(\mathit{succ}\ 0)$

b) $\lambda x{:}T.((\lambda y{:}T.y)\ x)$

c) $(\lambda x{:}T.((\lambda y{:}T.y)\ x))\ (\mathit{iszero}\ 0)$

d) $(0\ 0)\ (\mathit{iszero}\ 0)$

For each of these terms t,

- determine if t is reducible, and find all redexes s in t;
- determine all pairs of evaluation contexts C and terms s such that $t = C[s]$;
- determine the reduction context C and reduction redex s of t, if they exist.

Exercise 4.2 (RedContext and split) In Standard ML we can be represent PCF$^-$ contexts as procedures $c : \mathit{ter} \to \mathit{ter}$ such that the procedure application $c\ t$ yields the context applied to the term t.

a) Write a procedure $\mathit{redContext} : \mathit{ter} \to (\mathit{ter} \to \mathit{ter})$ that yields the reduction context of a reducible term. For example, if t is the term $(\lambda x{:}\mathit{Nat}.((\lambda y{:}\mathit{Nat}.y)\ x))\ (\mathit{natcase}\ 0\ 0\ \lambda z{:}\mathit{Nat}.0)$ then:

```
> val c = redContext t
val c : ter -> ter = _fn
> val t' = c O
val t' : ter = A (L ("x", Nat, A (L ("y", Nat, V "y"), V "x")), O)
```

b) Now extend this to a procedure $\mathit{split} : \mathit{ter} \to (\mathit{ter} \to \mathit{ter}) * \mathit{ter}$ that yields both the reduction context and the redex of a reducible term. For instance,

```
> val (c,r) = split t
val c : ter -> ter = _fn
val t : ter = C (O, O, L ("z", Nat, O))
```

The procedure *isVal* from the last assignment may be useful.

Exercise 4.3 (Closure semantics) To implement the closure semantics for PCF^-, we represent semantic values as follows:

```
datatype value = N of int
               | Cl of var * ter * env
               | RCl of var * var * ter * env
withtype env = var -> value
```

Write a procedure $clEval : ter \to value$ that yields the semantic value of a term, if it exists. Raise an exception if a term is encountered that does not satisfy the syntactic restrictions, i.e., that is of the form *fix* t where t is not a (double) abstraction or *natcase* $t\ t_1\ t_2$ where t_2 is not an abstraction.

Exercise 4.4 (*fix* terms) Find a closed PCF^- term $t : (nat \to nat) \to nat \to nat$ such that

- *fix* t does not converge, but
- $(\lambda p.\mathit{fix}(\lambda f.\lambda x.p\,f\,x))t$ converges.

(Here the type annotations are omitted for notational convenience.)