Assignment 3
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 10

There is a Standard ML file for this assignment. Send your solutions to Exercise 3.3 in a file named `lastname.sml` to doczkal@ps.uni-sb.de, and make sure that the entire file compiles without errors.

Exercise 3.1 (System T Shallow) Here is a shallow implementation of System T in Standard ML:

```
datatype nat = O | S of nat
fun natrec O     x _ = x
  | natrec (S n) x f = f n (natrec n x f)
```

Write the following procedures using only *natrec* for recursion.

a) *iszero* : *nat* → *bool*

b) *pred* : *nat* → *nat*

c) *add* : *nat* → *nat* → *nat*

d) *mul* : *nat* → *nat* → *nat*

e) *fac* : *nat* → *nat*

Exercise 3.2 (PCF$^-$ Shallow) Here is a shallow implementation of PCF$^-$ in Standard ML:

```
datatype nat = O | S of nat
fun natcase O x _ = x
  | natcase (S n) x f = f n
fun fix f x = f (fix f) x
```

A PCF$^-$ procedure for addition looks as follows:

```
val add = fix(fn f => fn x => fn y => natcase x y (fn x’ => f x’ (S y)))
```

Write and test PCF$^-$ procedures for multiplication and factorial.

Exercise 3.3 (PCF$^-$ Deep) We implement the abstract syntax of PCF$^-$ in Standard ML as follows:

```
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
```

a) Write a procedure $\mathit{isVal} : \mathit{ter} \to \mathit{bool}$ that tests whether a term is a value.

b) Write a procedure $\mathit{elab} : (\mathit{var} \to \mathit{ty}) \to \mathit{ter} \to \mathit{ty}$ that yields the type of a well typed term. Raise the exception *Error* if the term is not well-typed. Implement type environments as follows:

```
exception Error
fun empty x = raise Error
fun update f x a y = if y=x then a else f y
```

c) Write a procedure $\mathit{subst} : \mathit{var} \to \mathit{ter} \to \mathit{ter} \to \mathit{ter}$ that computes $[x := s]t$ if s is closed.

d) Write a procedure $\mathit{eval} : \mathit{ter} \to \mathit{ter}$ that yields the value of a closed term if it exists. Raise the exception *Error* it eval must quit because of a type inconsistency or a free variable occurrence.