CS 510 – The PROC-Language
Exercise Booklet 1

Exercise 1

Write a derivation to show that `let f = proc (x) { x - 11} in (f 77)` is a program in PROC.

Exercise 2

Write down the parse tree for the expression `let pred = proc(x) { x- 1} in (pred 5)`.

Exercise 3

Write an OCaml expression of each of the OCaml types below:

1. `expr`
2. `env`
3. `exp_val`

Exercise 4

Execute the following program using the interpreter for PROC and copy the output on paper.

```
proc (x) { x-11 }
```

Exercise 5

Write down the result of evaluating the following expression:

```
1 roc (x) { let y=2 in x }
```

Depict the full details of the closure including the environment. Use the tabular notation seen in class to depict the environment.

Exercise 6

Write down the result of evaluating the following expression:

```
1 let a=1
  in proc (x) { x }
```

Depict the full details of the closure including the environment.

Exercise 7

Write down the result of evaluating the following expression:

```
let a=1
in let b=2
in proc (x) { x }
```

Exercise 8

Write down the result of evaluating the following expression:

```
proc (x) { proc (y) { x- y } }
```

Exercise 9

Consider the following code in PROC

```
let x=2
in let y=proc (d) { x }
in let z=proc(d) { x }
in 3
```

Draw the environment used by the interpreter when it is about to evaluate line 4.

Exercise 10

Use the "higher-order" trick of self-application to implement the mutually recursive definitions of `even` and `odd` in PROC:

$$\begin{aligned} even(0) &= true \\ even(n) &= odd(n-1) \end{aligned}$$

$$\begin{aligned} odd(0) &= false \\ odd(n) &= even(n-1) \end{aligned}$$

Exercise 11

Use the "higher-order" trick of self-application to implement a function `pbt` that given a value `v` and a height `h` builds a perfect binary tree constructed out of pairs and that has `v` im the leaves and has height `v`. For example `((pbt 2) 3)` should produce

```
PairVal
 (PairVal
   (PairVal (PairVal (NumVal 2, NumVal 2),
     PairVal (NumVal 2, NumVal 2)),
   PairVal (PairVal (NumVal 2, NumVal 2),
    PairVal (NumVal 2, NumVal 2))),
 PairVal
  (PairVal (PairVal (NumVal 2, NumVal 2),
    PairVal (NumVal 2, NumVal 2)),
  PairVal (PairVal (NumVal 2, NumVal 2),
   PairVal (NumVal 2, NumVal 2))))
```