PLC : End-semester examination

April 28, 2020. 2.00 pm – 5.00 pm
Marks: 100
Weightage: 30

General instructions:

1. Submit your solutions as a PDF scan of your handwritten answers or as a plain text file on Moodle. The file should be named <un>-endsem.txt (or <un>-endsem.pdf, if you are submitting a PDF), where un is your username. For example, I would submit a file named spsuresh-endsem.txt.

2. If you are submitting a text file, use the Haskell notation \x -> M for the $\lambda x.M$ when writing expressions in the lambda calculus. (For $\lambda x y z.M$, write \x y z -> M.) Use the notation := for syntactic equality, and --> and = for (many-step) beta-reduction and beta-equality, respectively. Use M[x <- N] for substitution. Also use <m> for the Church encoding of m, and x^{y} for x^y.

3. Properly parenthesize your lambda expressions and use spacing to keep it readable.

4. Recall that the Church encoding of n, denoted $[n]$, is the expression $\lambda f x.f^n x$, where $f^0 x$ is defined to be just x, and $f^{i+1} x := f(f^i x)$.

5. You can assume the following standard encodings introduced in the lecture slides, with the appropriate behaviour:

$$
\begin{aligned}
[n] &= \lambda f x.f^n x \\
\textbf{succ} &= \lambda p f x.f(p f x) \\
\textbf{plus} &= \lambda p q f x.p f(q f x) \\
\textbf{mult} &= \lambda p q f.p(q f) \\
\textbf{true} &= (\lambda x y.x) \\
\textbf{false} &= (\lambda x y.y) \\
\textbf{pair} &= (\lambda x y w.w x y) \\
\textbf{fst} &= (\lambda p.p\ \textbf{true}) \\
\textbf{snd} &= (\lambda p.p\ \textbf{false}) \\
\textbf{ite} &= (\lambda b x y.b x y) \\
\textbf{iszero} &= (\lambda x.(x(\lambda z.\textbf{false}))\ \textbf{true})
\end{aligned}
$$

---

1. Reduce the following terms to β-normal form. (Note the need to rename bound variables appropriately.)

   (a) $(\lambda x y.x y)(\lambda v.v u)(\lambda x.x)$

   (b) $(\lambda x y.x y)((\lambda v.v u)(\lambda x.x))$

   (c) $(\lambda x.x(x(y z))x)(\lambda u.u v)$

   (d) $(\lambda x.x x y)(\lambda y.y z)$

   (e) $(\lambda x y.x y y)(\lambda u.u y x)$

(15 marks)

2. Find a lambda-calculus encoding for the function $S(n)$ defined by $S(n) = 0 + 1 + \cdots + n$ for $n \geq 0$. Prove that your encoding is correct. Do not use any fixed-point combinator. **(20 marks)**

3. Consider the following encoding of the empty list and **cons** operator.

$$[\,] = \lambda x y.y$$
$$H : T = \lambda x y.x H T$$

   Find λ-expressions with the following behaviour:

   (a) **null**: if $M = [\,]$ then **null** $M \xrightarrow{*}_{\beta}$ **true**, and if M is of the form $H : T$ then **null** $M \xrightarrow{*}_{\beta}$ **false**.

   (b) **head**, which extracts the head of a list and is defined only on inputs of the form $H : T$.

   (c) **tail**, which extracts the tail of a list and is defined only on inputs of the form $H : T$. **(20 marks)**

4. Give an example of a *closed λ-expression* (an expression without free variables) that belongs to each of the following types:

   (a) $a \to (a \to b) \to b$

   (b) $(a \to b \to c) \to (a \to b) \to (a \to c)$

   (c) $(a \to b \to c) \to (b \to a \to c)$

   (d) $((b \to c) \to a \to b) \to (b \to c) \to a \to c$

   (e) $a \to ((b \to a) \to c) \to c$ **(25 marks)**

5. Consider the following two threads running concurrently, with x and y being shared variables, with initial values 7 and 3 respectively. Assume that each assignment statement is atomic.

```
Thread A                        Thread B
--------                        --------
x := y + 1                      y := x + 1
x := 2*x*y                      y := 2*y*y
x := x - y                      y := y - x
```

Show how to generate the following combinations of (x,y) values at the end of the execution of both threads.

(a) (-10, 60)

(b) (30,10)

(c) (21,947)

(d) (4480, -2176)

Argue that at the completion of both threads, the value of x+y is always even. **(20 marks)**