Homework 2

Problem 1. Find λ terms representing the logical or and not functions.

Solution. or: $\lambda xy.x\mathbf{t}y$
not: $\lambda x.x\mathbf{ft}$

Problem 2. Prove that
(a) $\mathbf{add}\,\overline{m}\,\overline{n} \twoheadrightarrow_\beta \overline{m+n}$.
(b) $\mathbf{mult}\,\overline{m}\,\overline{n} \twoheadrightarrow_\beta \overline{m \cdot n}$.

Solution. (a)

$$\begin{aligned}
\mathbf{add}\,\overline{m}\,\overline{n} &\twoheadrightarrow_\beta \lambda fx.\overline{m}f(\overline{n}fx)\\
&\equiv \lambda fx.\overline{m}f((\lambda fx.f^n x)fx)\\
&\twoheadrightarrow_\beta \lambda fx.\overline{m}f(f^n x)\\
&\equiv \lambda fx.(\lambda fx.f^m x)f(f^n x)\\
&\twoheadrightarrow_\beta \lambda fx.(f^m(f^n x))\\
&\equiv \lambda fx.f^{m+n}x\\
&\equiv \overline{m+n}
\end{aligned}$$

(b)

$$\begin{aligned}
\mathbf{mult}\,\overline{m}\,\overline{n} &\twoheadrightarrow_\beta \lambda f.\overline{m}(\overline{n}f)\\
&\equiv \lambda f.(\lambda fx.f^m x)(\overline{n}f)\\
&\rightarrow_\beta \lambda f.(\lambda x.(\overline{n}f)^m x)\\
&\equiv \lambda f.(\lambda x.((\lambda fx.f^n x)f)^m x)\\
&\rightarrow_\beta \lambda f.(\lambda x.(\lambda x.f^n x)^m x)\\
&\rightarrow_\beta \lambda f.(\lambda x.(\lambda x.f^n x)^{m-1}(f^n x))\\
&\twoheadrightarrow_\beta \lambda fx.f^{mn}x\\
&\equiv \overline{m \cdot n}
\end{aligned}$$

where the second to last line can be derived by induction/ apply beta reduction m times. It is suggested to clearly state how you get $\lambda fx.f^{mn}x$.

Problem 3. Compute the β-normal forms of the following terms. Are they strongly normalizable?

(a) $(\lambda xy.yx)((\lambda x.xx)(\lambda x.xx))(\lambda xy.y)$.

(b) $(\lambda xy.yx)(\mathbf{kk})(\lambda x.xx)$.

Solution. Computation omitted.

(a) $\lambda y.y$, It is not strongly normalizable, since it has the $\mathbf{\Omega}$ term.

(b) $\lambda xy.x$. It is strongly normalizable, which can be shown by taking all possible reduction paths.

Problem 4. Find a representation of the following functions on integers

(a) $f(n) = \begin{cases} \text{true} & n \text{ is even,} \\ \text{false} & n \text{ is odd.} \end{cases}$

(b) $\exp(n, m) = n^m$.

(c) $\text{pred}(n) = \begin{cases} 0 & \text{if } n = 0, \\ n - 1 & \text{otherwise.} \end{cases}$ (Hard)

Solution. (a) $\lambda n.n\,\mathbf{not}\,\mathbf{t}$

(b) $\lambda nm.mn$ or $\lambda nm.m(\mathbf{mult}\,n)\overline{1}$

(c) Two types of constructions:

- $\lambda nfx.n(\lambda gh.h(gf))(\lambda u.x)(\lambda u.u)$. Check the following link.
- $\lambda n.\mathbf{fst}(n(\lambda p.\mathbf{pair}(\mathbf{snd}\,p)(\mathbf{succ}\,(\mathbf{snd}\,p)))(\mathbf{pair}\,\overline{0}\,\overline{0}))$. The idea behind is similar with computing the nth Fibonacci number, you would maintain a pair of $\langle n-1, n\rangle$, and update its value.

Some of you used the recursion trick for factorial in class for problem (a) and (b). It is correct, but sometimes with a little more hard work, you can get much simpler expressions.

Problem 5. Suppose two binary relations $\rightarrow_1$ and $\rightarrow_2$ *commute*, that is, $s \rightarrow_1 t_1$ and $s \rightarrow_2 t_2$ implies that there exists t such that $t_1 \rightarrow_2 t$ and $t_2 \rightarrow_1 t$. Let $\rightarrow_{12}$ be the union of $\rightarrow_1$ and $\rightarrow_2$. Prove that if $\rightarrow_1$ and $\rightarrow_2$ satisfy the diamond property, then so is $\twoheadrightarrow_{12}$.

Solution. You can first prove that the relation $\rightarrow_{12}$ satisfies the diamond property, then by the 'drawing the grid' method from class, $\twoheadrightarrow_{12}$ also satisfies the diamond property.

Problem 6. (Optional) Write an algorithm computing the factorial function in Python without using explicit recursion. Sample codes are provided in `lambda.py`. Note that the use of parenthesis in Python for function application is different from the mathematical way. For example, the term xyz used in classes as an abbreviation for $((xy)z)$ should be written as $x(y)(z)$ in Python in order to be consistent with the Python function call convention.