Computability and Complexity
Jacobs University Bremen
Dragi Kamov and Dushan Terzikj

Course: CO21-320352
April 2, 2019

Homework 7

Exercise 1

Solution:
To show that **and true true** $=$ **true**, we need to define the following λ-terms (taken from section 7.3 in the LN):

- **true** $\equiv \lambda xy.x$ (=K!)
- **false** $\equiv \lambda xy.y$
- **if** $\equiv \lambda pxy.pxy$
- **and** $\equiv \lambda pq.$ **if** $p\ q$ **false**

From these, we could further define **and**:

and $\equiv \lambda pq.(\lambda pxy.pxy)pq(\lambda xy.y)$

Then we have **and true true** = $(\lambda pq.(\lambda pxy.pxy)pq(\lambda xy.y))(\lambda xy.x)(\lambda xy.x)$

$$\begin{aligned}
&(\lambda pq.(\lambda pxy.pxy)pq(\lambda xy.y))(\lambda xy.x)(\lambda xy.x)\\
\rightarrow^* &(\lambda pq.(\lambda uvw.uvw)pq(\lambda xy.y))(\lambda xy.x)(\lambda xy.x)\\
\rightarrow^* &(\lambda uvw.uvw)(\lambda xy.x)(\lambda xy.x)(\lambda xy.y)\\
\rightarrow^* &(\lambda xy.x)(\lambda xy.x)(\lambda xy.y)\\
\rightarrow^* &(\lambda xy.x)\\
\rightarrow^* &x
\end{aligned}$$

Which in our case means true!

Second solution:
There is another solution which looks more intuitive and it's easier to understand by a mortal human being:

We define **and** $\equiv \lambda xy.xyF$, where F means false and T means true. **if** is omitted, because it is trivial and it does not change anything in our current case.

$$\begin{aligned}
&\textbf{and}\ TT\\
\rightarrow^* &(\lambda xy.xyF)TT\\
\rightarrow^* &(\lambda y.TyF)T\\
\rightarrow^* &TTF\\
\rightarrow^* &(\lambda xy.x)TF\\
\rightarrow^* &(\lambda y.T)F\\
\rightarrow^* &T
\end{aligned}$$

Reference for the second solution: http://blog.suspended-chord.info/2012/06/26/csmm—lesson-13-boolean-logic-in-the-lambda-calculus/.

Exercise 2

Solution:
From the problem sheet ***Laa = Lbb = Lcc = Lba = Lca = Lcb = false***, and ***Lab = Lac = Lbc = true***. From that we can see that **L** is true only when $a < b$, $b < c$ and $a < c$. From that we can see that **L** acts as "properly less than" ordering *a*, *b* and *c* in alphabetical order.
Define:

$$\text{SUB} \equiv \lambda ab.\ b\ -\ a$$
$$\text{ISZERO} \equiv \lambda a.\ a(b\ False)\ True$$
$$\text{LEQ} \equiv \lambda ab.\ ISZERO(SUB\ a\ b)$$
$$\text{NOT} \equiv \lambda a.\ a\ False\ True$$
$$\text{L} \equiv \lambda ab.\ NOT(LEQ\ b\ a)$$

(Referenced from https://jwodder.freeshell.org/lambda.html)