Exercises for Computability and Complexity, Spring 2019, Sheet 6 – Solutions

Please return your solutions in the Tuesday lecture on March 26

Exercise 1. Evaluate by call-by-value the following terms as far as possible (if you run into a dead end or loop, stop). Then do the same with call-by-name evaluation.

$K(KI(KI))y$
$K(KI(KIy))\, y$
$S\, K\, (KK(KIy))\, y$

Solution:

call-by-value:
$K(KI(KI))y$ immediately stops dead, because the innermost KI cannot be evaluated.
$K(KI(KIy))\, y \to K(KII)y \to KIy \to I$
$S\, K\, (KK(KIy))\, y \to S\, K\, (KKI)\, y \to S\, K\, K\, y \to K\, y\, (K\, y) \to$ dead end

call-by-name
$K(KI(KI))y \to KI(KI) \to I$
$K(KI(KIy))\, y \to KI(KIy) \to I$
$S\, K\, (KK(KIy))\, y \to K\, y\, ((KK(KIy))\, y) \to y$

Exercise 2 (medium) Express the following combinators in terms of S, K, and I. That is, give an expression for C in each case which consists solely of S, K, and I. *Remark:* we will soon learn a way of how to solve such problems systematically. The solutions returned by the automatic procedure are however typically far from optimal – they consist in unnecessarily long and complex expressions. If you want, you may read ahead in the lecture notes and use that procedure; here however I only want you to play around with combinators and find (shorter) solutions by intuition-guided experimentation.

(a) $Cxy = y$ *(A solution with 2 instances from {S, K, I} exists)*
(b) $Cxyz = y$ *(A solution with 2 instances from { S, K, I} exists)*
(c) $Cxyz = x$ *(A solution with 4 instances from { S, K, I} exists)*

A bit more challenging (optional)

(d) $C_n x_1 \ldots x_n = x_n$ *(A solution with $n+1$ instances from { S, K, I} exists)*

Even more challenging and optional (not easy):

(e) $C_n^i x_1 \ldots x_n = x_i$ This is analog to the projection operator known from recursive functions.

Solution.

(a) One solution is $C = KI$: $KI\, xy = Iy = y$.
(b) A solution is $C = KK$
(c) A solution is $C = S(KK)K$

(d) A solution is $C_n = K^1(\ldots K^{n\text{-}1}(K^n I)\ldots)$: $K^1(\ldots K^{n\text{-}1}(K^n I)\, x_1 \ldots x_n = K^2(\ldots K^{n\text{-}1}(K^n I)\ldots)\, x_2 \ldots x_n = \ldots = K^n I\, x_{n\text{-}1} x_n = x_n$.

Comment. A halfway "divide and conquer" systematic experimentation procedure runs as follows. We consider task (d) as an example. Wanted: an expression in terms of *S, K, I* for *C* such that $Cxyz = x$. "Divide and conquer": *C* must satisfy $(Cxy)z = x$. This is achieved when $Cxy = Kx$. Thus we search for *C* such that $(Cx)y = Kx$. This is achieved if $Cx = K(Kx)$. A little experimentation finds $C = S(KK)K$.

Exercise 3 Construct the combinator *C* defined by $Cxy = y$ from *S* and *K* and *I*, by meticulously applying the construction method from the proof of Prop. 7.3. (This is the same *C* as in Exercise 2a). This is a rather short construction (3 lines), but hairy – it takes a lot of care to really get the construction right according to the method in Prop. 7.3. Document all the detailed steps that this construction requires from you. Please use exactly the notation from Prop. 7.3 – starting with setting $t_2[x,y] = y$.

Solution. Start with $t_2[x,y] = y$. This is non-composite, and the case $t_n = x_n$ applies. Hence, $t_1[x] = I$. Again, *I* is non-composite, therefore the cases applies and we get $t_0[] = KI$. Assembling: $t_2[x,y] = t_1[x]y = t_0[]xy = KIxy$, hence $C = KI$.