Computation Theory

Supervision 3

9. Lambda calculus

1. Given a set $\mathcal{V} = \{x, y, \ldots\}$ of variables, define the set $\mathcal{T}$ of λ-terms
   a) as an inductively defined set (see IA Formal Languages course).
   b) using Backus–Naur form (see IB Semantics course).
   c) using a recursive set comprehension (see Lecture 7 of the IB Logic course).
2. a) Simplify the following λ-terms (as much as possible, but without evaluating them) using the notational conventions described on Slide 105:

   $$(\lambda x.\,((ux)y)) \qquad (((\lambda u.\,(\lambda v.\,(vu)u))z)y) \qquad ((((\lambda x.\,(\lambda y.\,(\lambda z.\,((xz)(yz)))))u)v)w)$$

   b) Expand the following simplified λ-terms (as much as possible) using the notational conventions described on Slide 105, inserting all parentheses and λ's:

   $$xyz(yx) \qquad \lambda u.\,u(\lambda x.\,y) \qquad \lambda xy.\,ux(yz)(\lambda v.\,vy)$$

3. Give a recursive definition of the function $len(M)$ denoting the *length* of the λ-term M given by the total number of variables in M. For example, $len(x(\lambda y.\,yux)) = 5$.
4. a) Define the *subterm relation* $M \sqsubseteq N$ by recursion on N. For example,

   $$x \sqsubseteq \lambda y.\,ux \qquad \lambda x.\,y \sqsubseteq \lambda x.\,y \qquad xy \sqsubseteq (\lambda x.\,xy)z \qquad z \sqsubseteq x(\lambda z.\,y)$$

   but $uv \not\sqsubseteq \lambda x.\,xu(vy)$.

   b) We say there is *an occurrence of* M *in* N if $M \sqsubseteq N$.
      (i) Mark all occurrences of xy in $(xy)(\lambda x.\,xy)$.
      (ii) Mark all occurrences of x in $(xy)(\lambda x.\,xy)$.
      (iii) Mark all occurrences of xy in $\lambda xy.\,xy$.
      (iv) Mark all occurrences of uv in $x(uv)(\lambda u.\,v(uv))uv$.
      (v) Does $\lambda u.\,u$ occur in $\lambda u.\,uv$?
5. Let M be the λ-term $\lambda xy.\,x(\lambda z.\,zu)y$.
   a) What is the β-normal form of the term $N = M(\lambda vw.\,v(wb))(\lambda xy.\,yaz)$?
   b) Apply the simultaneous substitution $\sigma = [x/y, (\lambda xy.\,zy)/u]$ to M and N, and find the β-normal form of $N[\sigma]$.
   c) Give 2 terms α-equivalent to M. Give 2 other terms β-equivalent to M.
   h. We define η-equivalence as: $M =_\eta \lambda x.\,Mx$ for any λ-term M. Give a shorter and a longer term η-equivalent to M. What is the use of η-equivalence in functional programming?

6. What are some differences between the lambda calculus as defined in this course, and the functional subset of L2 from the IB Semantics course?

10. Lambda-definable functions

1. Give a complete proof of the correctness of Church addition from Slide 119. *Hint:* a formal justification of one of the steps will require mathematical induction.

$$\mathbf{Plus}\,\underline{m}\,\underline{n} =_\beta \underline{m+n}$$

2. Define the λ-terms **Times** and **Exp** representing multiplication and exponentiation of Church numerals respectively. Prove the correctness of your definitions.

$$\mathbf{Times}\,\underline{m}\,\underline{n} =_\beta \underline{m \times n} \qquad \mathbf{Exp}\,\underline{m}\,\underline{n} =_\beta \underline{m^n}$$

3. Show that the λ-term $\mathbf{Ack} \triangleq \lambda x.\, x\, T\, \mathbf{Succ}$, where $T \triangleq (\lambda f\, y.\, y f (f\, \underline{1}))$ represents Ackermann's function $ack \in \mathbb{N}^2 \to \mathbb{N}$. *Hint*: you will need to use nested induction; consider deriving a simplified form for the outer inductive case before starting the nested proof.

4. Consider the following λ-terms:

$$\mathbf{I} \triangleq \lambda x.\, x \qquad \mathbf{B} \triangleq \lambda g f x.\, f\, x\, \mathbf{I} (g (f\, x))$$

   a) Show that $\underline{n}\,\mathbf{I} =_\beta \mathbf{I}$ for every $n \in \mathbb{N}$.

   b) Assuming the fact about normal order reduction mentioned on Slide 115, show that if partial functions $f, g : \mathbb{N} \rightharpoonup \mathbb{N}$ are represented by closed λ-terms F and G respectively, then their composition $(g \circ f)(x) \triangleq g(f(x))$ is represented by $\mathbf{B}\, G\, F$. Explain how this avoids the discrepancy with partial functions mentioned in Slide 125.

5. In the following questions you may use all of the λ-definable functions presented in the notes, as well as the terms you define as part of this exercise. You should explain your answers (possibly using some examples), but don't need to prove their correctness.

   a) Give a λ-term **Not** representing Boolean negation.

   b) Give λ-terms **And** and **Or** representing Boolean conjunction and disjunction.

   c) Give a λ-term **Minus** representing truncated subtraction (i.e. $\mathbf{Minus}\,\underline{m}\,\underline{n} = 0$ if $m < n$).

   d) Give λ-terms **Eq, NEq, LT, LEq, GT, GEq,** representing the numeric comparison operations $=, \neq, <, \leq, >, \geq$ respectively. You can define them in any order you find most convenient.

   e) Define the λ-term **UCr** that represents the uncurrying higher-order function: if $\langle M, N \rangle$ denotes $\mathbf{Pair}\, M\, N$, then $\mathbf{UCr}\, F\, \langle M, N \rangle =_\beta F\, M\, N$.

   f) Give a λ-term **MapPair** that applies a function to both elements of a pair: that is, $\mathbf{MapPair}\, F\, \langle M, N \rangle =_\beta \langle F\, M, F\, N \rangle$.

   g) Give a λ-term **SqSum** which represents the function $\langle m, n \rangle \mapsto m^2 + n^2$.

6. a) Explain why Curry's $\mathbf{Y}$ combinator is needed and how it works.

   b) Give a λ-term which is β-equivalent to the $\mathbf{Y}$ combinator, but only uses it's f argument once. *Hint*: see if you can exploit the symmetry of the $\mathbf{Y}$ combinator.

   c) Define the λ-term $\mathbf{Fact}$ that computes the factorial of a Church numeral.

   d) Define the λ-term $\mathbf{Fib}$ such that $\mathbf{Fib}\,\underline{n} =_\beta \underline{F_n}$ where F_n is the n^{th} Fibonacci number defined recursively as $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$.