Higher-Order Computability
4. Exercise Sheet

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Mathematics
Dr. Thomas Powell

Summer Semester 2019
18 June

Homework

Key to exercises: (P) = programming component (+) = more difficult or open ended.

Exercise H1 (warm-up)
Give a standard reduction sequence for the following term of System T:

$$R_N(\mathbf{2})(\lambda n, k.s(k))(\mathbf{1}).$$

What is the *derivational complexity* of the term (i.e. the number of standard reductions needed to reduce it to normal form) and its value (i.e. the normal form it evaluates to)?

Exercise H2
For each of the following terms $t : (N \to N) \to N$ give standard *oracle* reduction sequences for tx where x is an oracle variable with contraction rule $x\mathbf{n} \rhd \alpha(\mathbf{n})$.

(a) $t := \lambda y.y((\lambda z.yz)(\mathbf{1}))$

(b) $t := \lambda y.R_N(0)(\lambda n, k.yk)(y0)$.

Give expressions for the normal form and the derivational complexity of tx in terms of α. Give moduli of continuity for the functionals $F_t : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ induced by each of these terms.

Exercise H3
Indicate which of the following functionals $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous, and give a modulus of continuity where relevant.

(a) $F(\alpha) := \alpha^{\alpha(0)}(\alpha(1)) + 5$.

(b) $F(\alpha) := \alpha(0)$ if the Riemann hypothesis is true, else $\alpha(1)$.

(c) $F(\alpha) := n$ where n is the least number with $\alpha(n) \leq \alpha(n+1)$ if it exists, else 0.

(d) $F(\alpha) := n$ where n is the least number with $\alpha(n) \geq \alpha(n+1)$ if it exists, else 0.

(e) $F(\alpha) := n$ where n is the least number with $\alpha(n) \geq 2$ and $a^{\alpha(n)} + b^{\alpha(n)} = c^{\alpha(n)}$ for some positive integers a, b, c if it exists, else 0.

(f) $F(\alpha) := n$ where n is the least number with $\alpha(n) > 2$ and $a^{\alpha(n)} + b^{\alpha(n)} = c^{\alpha(n)}$ for some positive integers a, b, c if it exists, else 0.

Exercise H4 (+)
Show that there exists a type 3 functional $\Gamma : (\mathbb{N} \to C_2) \to \mathbb{N}$ definable in System T, such that any modulus of continuity $|\Gamma|$ for Γ satisfying

$$\forall A, B : \mathbb{N} \to C_2 \; (A =_{|\Gamma|(A)} B \Rightarrow \Gamma(A) = \Gamma(B))$$

allows us to construct a modulus of continuity functional $\Phi : C_2 \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ at type 2 satisfying

$$\forall F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}, \alpha, \beta : \mathbb{N}^{\mathbb{N}} (\alpha =_{\Phi(F,\alpha)} \beta \Rightarrow F(\alpha) = F(\beta)).$$

Exercise H5 (P)
Take your favourite programming language, and suppose that you are given a piece of code $P[\alpha]$ which returns a natural number but calls some subroutine α. How would you modify your code to work out on exactly which inputs α is called?