Eindhoven University of Technology

Faculty of Mathematics and Computer Science

Written Exam (resit) **Proving with Computer Assistance 2IF65**
Monday June 22 2015, 13.30–16.30
The maximum number of points per exercise is indicated in the margin. Maxium 100 points in total **NB** Typing derivations may be given in "flag style" or in "sequent style".

---

(10) 1. Construct a term M of type $((\alpha \to \beta) \to \beta \to \alpha) \to \beta \to \alpha$ in simple type theory $(\lambda\!\to)$ à la Church and give a typing derivation of

$$M : ((\alpha \to \beta) \to \beta \to \alpha) \to \beta \to \alpha.$$

(10) 2. Show that the following term P is typable in simple type theory $(\lambda\!\to)$ à la Church.

$$\lambda x : (\alpha \to \alpha) \to \alpha . x(\lambda y : \alpha . x(\lambda z : \alpha . y))$$

Give a typing derivation that gives the type of P.

(20) 3. Consider the following two terms

- $\lambda x.\lambda y.x(\lambda z.y)\, y$
- $\lambda x.\lambda y.x(\lambda z.x)\, y$

For each of these terms, compute its principle type in simple type theory $(\lambda\!\to)$ à la Curry, if it exists.
Give the end result and show your computation; if the term has no principle type, show how your computation yields 'fail'.

(10) 4. Let $x : \top$ and remember that $\top := \forall \alpha{:}\, * . \alpha \to \alpha$. Give a type to the term

$$\lambda y . x\, y\;\, x(\lambda z . z\, x\, z)$$

in $\lambda 2$ à la Curry. Also give the typing derivation of your result.

Continue on the other side

5. In $\lambda 2$ à la Church we define the type T of *ternary trees* as follows.

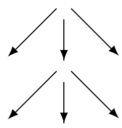
$$T \quad := \quad \forall \alpha{:}*.\alpha \to (\alpha \to \alpha \to \alpha \to \alpha) \to \alpha$$

The constant $\mathsf{leaf} : T$ is defined as follows.

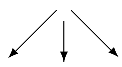
$$\mathsf{leaf} \quad := \quad \lambda \alpha{:}*.\lambda l{:}\alpha.\lambda j{:}\alpha \to \alpha \to \alpha \to \alpha.l$$

(5) (a) Define the function $\mathsf{join} : T \to T \to T \to T$

(5) (b) Define a term $t_0 : T$ that represents the following tree



(10) (c) Define a term $f : T \to T$ that takes a tree t and replaces each leaf by the subtree



6. In the system λP, consider the context

$$\begin{aligned}\Gamma \quad := \quad & A : *, c : A, R : A \to A \to *, f : A \to A, k : \Pi x{:}A.R\,c\,x, \\ & h : \Pi x, y{:}A.R\,x\,y \to R\,(f\,x)\,(f\,y), r : \Pi x, y{:}A.R\,x\,y \to R\,x\,(f\,y)\end{aligned}$$

(5) (a) Construct a term M (and give a typing derivation in short form) such that

$$\Gamma \vdash M : \Pi x{:}A.R\,(f\,c)\,(f(f\,x)).$$

(10) (b) Construct a term N (and give a typing derivation in short form) such that

$$\Gamma \vdash N : \Pi x, y{:}A.R\,(f\,x)\,y \to R\,(f(f\,x))\,(f(f\,y)).$$

7. We define, in the Calculus of Constructions λC, given $A : *$,

$$T := \lambda a, b{:}A.\Pi R{:}A \to A \to *.(\Pi x{:}A.R\,x\,x) \to (\Pi x, y{:}A.R\,x\,y \to R\,y\,x) \to R\,a\,b$$

(7) (a) Give a term of type

$$\Pi a{:}A.T\,a\,a$$

(8) (b) Give a term of type

$$\Pi a, b{:}A.T\,a\,b \to T\,b\,a$$

NB. In both cases, give a typing derivation in short version (where you don't have to establish the well-formedness of the types themselves).

END