Exercises Toegepaste Logica 99-00

1. This question concerns minimal logic and simply typed lambda calculus.

   a. Give a proof of the formula

   $$(A \to B \to B \to C) \to A \to B \to C$$

   in minimal logic such that the proof contains a detour and doesn't contain open assumptions.

   b. Give the typing derivation corresponding to the proof given as anwer to question 1a.

   c. Reduce the term of the answer to question 1b to beta-normal form and give its typing derivation.

   d. Give the proof corresponding to the typing derivation of the answer to question 1c.

2. This question concerns minimal logic and the Curry-Howard-De Bruijn isomorphism.

   a. Give three different ways to extend minimal intuitionistic logic to minimal classical logic.

   b. Explain in detail the correspondence of formulas, proofs and detour elimination in minimal logic with concepts in simply typed lambda calculus.

   c. What is inhabitation? To which concept in minimal logic does it correspond?

   d. What is type checking? To which concept in minimal logic does it correspond?

3. This question concerns Gödel's **T**.

   a. Consider the term

   $$\mathsf{mul} = \lambda m{:}\mathsf{Nat}.\, \lambda n{:}\mathsf{Nat}.\, \mathsf{r}_{\mathsf{Nat}}(\mathsf{z}, \lambda x{:}\mathsf{Nat}.\, \lambda y{:}\mathsf{Nat}.\, \mathsf{plus}\ x\ n, m).$$

   Give a type derivation of this term in Gödel's system **T**, using $\mathsf{plus} : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$ without giving the (sub)derivation. (If such a question is asked at the examination, the typing rules of Gödel's system **T** will be given.)

   b. Consider the term

   $$\mathsf{plus} = \lambda m{:}\mathsf{Nat}.\, \lambda n{:}\mathsf{Nat}.\, \mathsf{r}_{\mathsf{Nat}}(n, \lambda x{:}\mathsf{Nat}.\, \lambda y{:}\mathsf{Nat}.\, \mathsf{s}(x), m).$$

   Show that this term represents addition. That is, show that we have

   $$\begin{array}{lcl} \mathsf{plus}\ \mathsf{z}\ q & =_{\mathbf{T}} & q \\ \mathsf{plus}\ \mathsf{s}(p)\ q & =_{\mathbf{T}} & \mathsf{s}(\mathsf{plus}\ p\ q) \end{array}$$

   for $p : \mathsf{Nat}$ and $q : \mathsf{Nat}$.

4. This question concerns inhabitation for simply typed lambda calculus.

    a. Give the definition of the η-reduction rule.

    b. Explain why every type A can be written as $A_1 \to \ldots \to A_n \to b$ with b a base type.

    c. Give an informal description of a procedure that decides whether a type A is inhabited.

5. This question concerns inductive types in Coq.

    a. Give and explain the definition of an inductive type `natlist` of finite lists of natural numbers in Coq.

    b. Give the type of the term `natlist_ind` that is automatically generated by Coq once the inductive definition of `natlist` is given.

    c. Explain how the term `natlist_ind` is used for induction on lists of natural numbers.

6. This question concerns termination of simply typed lambda calculus.

    a. Give the definition of the interpretation of a type (written as $[A]$).

    b. Give, using the fact that $[A \to B] = [A] \to [B]$, a proof of termination of simply typed lambda calculus.

7. This question concerns lambda calculus with dependent types (λP) and the Curry-Howard-De Bruijn isomorphism between a fragment of λP and first-order predicate logic.

    a. How is a function symbol f of arity n of first-order predicate logic represented in λP?

    b. How is a predicate symbol r of arity n of first-order predicate logic represented in λP?

    c. Explain the correpondence between formulas in first-order predicate logic and types in λP.

    d. The formal definition of the typing system of λP makes use of two constants: $\star$ and $\Box$. What do the universes `Set`, `Prop`, and `Type` of Coq correspond to?

    e. Give a derivation in the typing system of λP of the following statement:

    $$\mathsf{Nat} : \star, P : \mathsf{Nat} \to \star, n : \mathsf{Nat} \vdash (\Pi m{:}\mathsf{Nat}.\, P\, m) \to P\, n : \star$$

    See the homework of week 9. Note that we use the notation $A \to B$ for $\Pi x{:}A.\, B$ if x doesn't occur in B. (If such a question is asked at the examination, the rules of the typing system of λP will be given.)

    f. Explain why the conversion rule

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}$$

with $B =_\beta B'$ is necessary for lambda calculus with dependent types.

8. This question concerns lambda calculus with besides dependent also polymorphic types ($\lambda P2$).

   a. Give an example of the use of polymorphic types from a programming point of view.

   b. Give a derivation in the typing system of $\lambda P2$ of the following statement:

   $$A : \star, B : \star \vdash \Pi a{:}\star .\, (A \to a) \to (B \to a) \to a : \star$$

   (If such a question is asked at the examination, the rules of the typing system of $\lambda P2$ will be given.)

9. This question concerns Coq.

   (a) Explain what program extraction is (this is done in Coq using the tactic `Extraction`).

   (b) Explain the use of the tactic `Program`.