Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Johannes Hölzl and Lars Hupel

Equational Logic
Summer Term 2015
Exercise Sheet 12

Exercise 60 (Type Inference)

Give a derivation tree for the following statement, and so determine the type τ:

$$[] \vdash \lambda xyz.\ x\ y\ (y\ z)\ : \tau$$

Exercise 61 (Recursive `let`)

Recursive `let` expressions are one way (besides Y-combinators) to add recursion to $\lambda^{\rightarrow}$.

$$t ::= x \mid (t_1\ t_2) \mid (\lambda x.\ t) \mid \texttt{letrec}\ x = t_1\ \texttt{in}\ t_2$$

a) Modify the standard typing rule for `let` to create a suitable rule for `letrec`.

b) Give a derivation tree for the following statement, and so determine the type τ:

$$[] \vdash\ \texttt{letrec}\ x = \lambda y.\ x\ (x\ y)\ \texttt{in}\ x\ x : \tau$$

Homework 62 (`let`-Polymorphism)

Give a derivation tree for the following statement, and so determine the type τ:

$$[z : \tau_0] \vdash\ \texttt{let}\ x = \lambda yz.\ z\ y\ y\ \texttt{in}\ x\ (x\ z)\ : \tau$$

Homework 63 (Constructive logic)

a) Prove the following statement using the calculus for intuitionistic propositional logic:

$$((c \to b) \to b) \to (c \to a) \to ((a \to b) \to b)$$

Hint: To make your proof tree more compact, you may remove unneeded assumptions to the left of the $\vdash$ during the proof as you see fit. For example, the following step is valid:

$$\frac{p \vdash p}{p, q \vdash p}$$

b) Give a well-typed expression in $\lambda^{\rightarrow}$ with the type

$$((\gamma \to \beta) \to \beta) \to (\gamma \to \alpha) \to ((\alpha \to \beta) \to \beta)$$

(You don't need to give the derivation tree.)