PRFSYS — Foundations of Proof Systems

Exam

Nov. 25^{th} 2025

1 Trees in system F

We suppose we have already defined the type nat of natural numbers together with 0 and S and the addition function over them.

Question 1 Define a type Tree representing binary trees with natural numbers at the nodes, and nothing at the leaves.

That is, define Tree and two terms :

— Leaf : Tree

— Node : nat → Tree → Tree → Tree. ◇

Question 2 Construct a function count : Tree → nat which counts the number of nodes of a tree (without taking the leaves into account). ◇

Question 3 Construct function tsum : Tree → nat which returns the sum of the values at the nodes. ◇

2 Impredicative Definitions of Properties

We are in HOL. We consider as given the type nat with 0 and S, as well as a type list and two constants nil : list and cons : nat → list → list.

Question 4 Define ≤: nat → nat → o which states that a number is less or equal than another. ◇

Question 5 Define low : nat → list → o which states that either a number is less or equal than the first element of a list, or the list is empty. ◇

Question 6 Define sorted : list → o which states that a list is sorted. ◇

3 Exercises in Type Theory

Question 7 We are in Martin-Löf's Type Theory.

Fill in the blanks, so that the following statements ought to hold ; or say when they cannot hold.

For instance, for $[] \vdash 0 : \boxed{}$, you should answer N (it is not necessary to mention well-formed types convertible to N). You may want to add some remarks or side conditions, but do not go into long explanations.

$$
\begin{array}{rcll}
[] & \vdash & (a,b) : \boxed{} & (1)\\
\Gamma & \vdash & \lambda x : A.t : \boxed{} & (2)\\
[] & \vdash & \boxed{} : A + B & (3)\\
\Gamma & \vdash & (\mathsf{refl}_A\ t) : \boxed{} & (4)\\
[] & \vdash & \boxed{} : a =_A b & (5)\\
[(x : N)] & \vdash & \boxed{} : N & (6)\\
[] & \vdash & \boxed{} : \bot & (7)
\end{array}
$$

◇

4 A paradox in Type : Type

We consider the pure type system with a unique sort Type such that Type : Type. That is Martin-Löf's original, paradoxical type theory of 1971.

$$
\frac{}{[]\ \mathsf{wf}} \qquad \frac{\Gamma \vdash T : \mathsf{Type}}{\Gamma(x : T)\ \mathsf{wf}} \qquad \frac{\Gamma\ \mathsf{wf}}{\Gamma \vdash \mathsf{Type} : \mathsf{Type}} \qquad \frac{\Gamma\ \mathsf{wf}}{\Gamma \vdash x : T}\ (\text{If } (x : T) \in \Gamma)
$$

$$
\frac{\Gamma \vdash T_1 : \mathsf{Type} \quad \Gamma(x : T_1) \vdash T_2 : \mathsf{Type}}{\Gamma \vdash \Pi x : T_1.T_2 : \mathsf{Type}} \qquad \frac{\Gamma(x : U) \vdash t : T}{\Gamma \vdash \lambda x : U.t : \Pi x : U\,.\,T}
$$

$$
\frac{\Gamma \vdash t : \Pi x : U\,.\,T \quad \Gamma \vdash u : U}{\Gamma \vdash (t\ u) : T[x \setminus u]} \qquad \frac{\Gamma \vdash t : T \quad \Gamma \vdash U : \mathsf{Type}}{\Gamma \vdash t : U}\ (\text{if } T =_\beta U)
$$

Question 8 Describe a (very) simple transformation f over terms of PTSs, such that :

— if $\Gamma \vdash t : T$ in some PTS,

— then $f(\Gamma) \vdash f(t) : f(T)$ in the PTS above.

— With the requirement that if $t \triangleright_\beta t'$ then $f(t) \triangleright_\beta f(t')$. ◇

Question 9 We define

$$\mathsf{U} \equiv \Pi X : \mathsf{Type}.X \to \mathsf{Type}.$$

What is the type of U ? ◇

Question 10 Given $u : \mathsf{U}$, how do you, simply, turn u into a property over U, that is a term of type $\mathsf{U} \to \mathsf{Type}$? ◇

Question 11 Construct a relation $\in : \mathsf{U} \to \mathsf{U} \to \mathsf{Type}$.
From now on, we write $u \in v$ for $(\in\ u\ v)$. ◇

We also add to the type system the, usual, primitive equality, together with the additional axiom K which states unicity of equality proofs :

$$
\begin{array}{rcll}
= & : & \Pi X : \mathsf{Type} \,.\, X \to X \to X \to \mathsf{Type} & \text{we write } t =_T u \text{ for } (=\ T\ t\ u). \\
\mathsf{refl} & : & \Pi X : \mathsf{Type} \,.\, \Pi x : X \,.\, x =_X x & \\
L & : & \Pi X : \mathsf{Type} \,.\, \Pi x : X \,.\, \Pi y : X \,.\, \Pi P : X \to \mathsf{Type} \,.\, (P\ x) \to x =_X y \to (P\ y) & \\
K & : & \Pi X : \mathsf{Type} \,.\, \Pi x : X \,.\, \Pi P : x =_X x \to \mathsf{Type} \,.\, \Pi e : x =_X x \,.\, (P\ (\mathsf{refl}\ X\ x)) \to (P\ e) &
\end{array}
$$

with the (usual) reduction rules :

$$
\begin{array}{rcl}
(L\ T\ t\ t'\ P\ p\ (\mathsf{refl}\ T'\ t'')) & \rhd & p \\
(K\ T\ t\ P\ (\mathsf{refl}\ T'\ t')\ p) & \rhd & p
\end{array}
$$

Remark : you will not need the second reduction rule.

Using this equality, we will now build the construction dual to $\in$. That is construct an term $\mathsf{comp} : (\mathsf{U} \to \mathsf{Type}) \to \mathsf{U}$, such that $u \in (\mathsf{comp}\ P)$ will be equivalent to $P\ u$.

Question 12 Construct

$$\mathsf{tr} : \Pi X : \mathsf{Type}.\mathsf{U} =_{\mathsf{Type}} X \to (\mathsf{U} \to \mathsf{Type}) \to (X \to \mathsf{Type}).$$

What is a notable reduct of $(\mathsf{tr}\ \mathsf{U}\ (\mathsf{refl}\ \mathsf{U})\ P)$? ⋄

We can now define

$$\mathsf{comp} \equiv \lambda P : \mathsf{U} \to \mathsf{Type} \,.\, \lambda X : \mathsf{Type} \,.\, \lambda x : X \,.\, \Pi e : \mathsf{U} =_{\mathsf{Type}} X \,.\, \mathsf{tr}\ X\ e\ P\ x.$$

Question 13 Give a proof $\mathsf{trid} : \Pi e : \mathsf{U} =_{\mathsf{Type}} \mathsf{U} \,.\, \mathsf{tr}\ \mathsf{U}\ e\ P = P$. ⋄

Question 14 Give a term :

$$g : \Pi P : \mathsf{U} \to \mathsf{Type}.\Pi u : \mathsf{U} \,.\, P\ u \to (u \in (\mathsf{comp}\ P))$$

Question 15 Given $u : \mathsf{U}$, $P : \mathsf{U} \to \mathsf{Type}$ and $i : (In\ u\ (\mathsf{comp}\ P))$, give a term $t : (P\ u)$. ⋄

We now give ourselves a variable $\alpha : \mathsf{Type}$. We write $\neg^\alpha T$ for $T \to \alpha$.

Question 16 Define R of type U which corresponds to Russell's paradoxical set $\{x | \neg^\alpha (x \in x)\}$.

Show $\neg^\alpha(\mathsf{R} \in \mathsf{R})$ ⋄

Question 17 Show $(\mathsf{R} \in \mathsf{R})$. Deduce α. ⋄

Question 18 Explain, from this, why the type system cannot enjoy normalization. ⋄

Fixed-Point Operator

In this last part, we want to go a little further and build a real fixed-point operator. For that, for any $x : \mathsf{U}$ and $P : \mathsf{U} \to \mathsf{Type}$, we admit that we can construct two terms :

$$\begin{array}{rcl} F_P^x & : & P\,x \to x \in (\mathsf{comp}\,P) \\ G_P^x & : & x \in (\mathsf{comp}\,P) \to P\,x \end{array}$$

with the property that $(G_P^x\,(F_P^x\,p)) \rhd_\beta^* p$.

We write :

$$\begin{array}{rcl} P & \equiv & \lambda x : \mathsf{U}\,.\,\neg^\alpha(x \in x) \\ F_0 & \equiv & F_P^{\mathsf{R}} : \neg^\alpha(\mathsf{R} \in \mathsf{R}) \to \mathsf{R} \in \mathsf{R} \\ G_0 & \equiv & G_P^{\mathsf{R}} : \mathsf{R} \in \mathsf{R} \to \neg^\alpha(\mathsf{R} \in \mathsf{R}) \end{array}$$

Question 19 Using G_0 and F_0, construct two terms of type :

$$\begin{array}{rcl} H_1 & : & \neg^\alpha(\mathsf{R} \in \mathsf{R}) \\ H_2 & : & \mathsf{R} \in \mathsf{R} \end{array}$$

What does $(H_1\ H_2)$ reduce to ? ⋄

Question 20 We give ourselves a variable $f : \alpha \to \alpha$. By modifying the terms of the previous question, build a term $Y : \alpha$, such that $Y \rhd_\beta^* f\ Y$. ⋄

Question 21 Show that all λ-terms are typable in this type system. ⋄