2.7.1 — Foundations of Proof Systems

Exam

Nov. 27^{th} 2023

Durée de l'épreuve : 2 heures. Length of the exam : 2 hours.

1 HOL

For conciseness I write $\forall X^T \ldots$ instead of $(\forall_T \, \lambda X^T \ldots)$.

Question 1 Given two propositions A and B in HOL, what do the following propositions correspond to? (in natural language)

1. $\forall X^o \,.\, A \implies X^o$
2. $\forall X^o \,.\, (A \implies B \implies X^o) \implies X^o$
3. $\forall X^o \,.\, (A \implies X^o) \implies X^o$
4. $\forall X^o \,.\, ((A \implies B) \implies X^o) \implies ((B \implies A) \implies X^o) \implies X^o$ $\diamond$

Solution.
1. $\neg A$
2. $A \wedge B$
3. equivalent to A
4. $(A \implies B) \vee (B \implies A)$ $\square$

Question 2 Same question for the following constructions, given a property $P : \iota \to o$ and a relation $R : \iota \to \iota \to o$.

1. $\forall x^\iota . \forall y^\iota . (R\; x^\iota\; y^\iota) \implies (R\; y^\iota\; x^\iota) \implies \forall Q^{\iota \to o} \,.\, (Q\; x^\iota) \implies (Q\; y^\iota)$
2. $\forall X^o \,.\, (\forall x^\iota \,.\, (P\; x^\iota) \implies X^o) \implies X^o$
3. $\lambda a^\iota . \lambda b^\iota \,.\, \forall X^{\iota \to o} \,.\, (X^{\iota \to o}\; a^\iota) \implies (\forall x^\iota . \forall y^\iota . (X^{\iota \to o}\; x^\iota) \implies (R\; x^\iota\; y^\iota) \implies (X^{\iota \to o}\; y^\iota)) \implies (X^{\iota \to o}\; b^\iota)$ $\diamond$

Solution.
1. R is anti-symmetrical
2. $\exists x . P(x)$
3. The transitive closure of R $\square$

2 System F

We use the usual encoding of natural numbers in System F as Church Numerals of the following type :

$$\mathrm{nat} \equiv \forall X\,.\,X \to (X \to X) \to X$$

Question 3 Define the type NN which encodes the pairs of natural numbers, as well as the corresponding terms :

$$\begin{array}{rcl} pair & : & \mathrm{nat} \to \mathrm{nat} \to NN \\ \pi_1 & : & NN \to \mathrm{nat} \\ \pi_2 & : & NN \to \mathrm{nat} \end{array}$$

◇

Solution.

$$\begin{array}{rcl} NN & \equiv & \forall X.(\mathrm{nat} \to \mathrm{nat} \to X) \to X \\ pair & \equiv & \lambda a\; b : \mathrm{nat}\,.\Lambda X.\lambda f : \mathrm{nat} \to \mathrm{nat} \to X.f\; a\; b \end{array}$$

□

Question 4 Define the term $s : NN \to NN$ corresponding to the function $(n, m) \mapsto (S\; n, n)$. ◇

Solution.

$$s \equiv \lambda c : NN.pair\;(S\;(\pi_1\; c))\;(\pi_1\; c)$$

Question 5 Use this to define a predecessor function over nat. ◇

Solution.

$$pred \equiv \lambda n.\pi_1\;(n\; NN\;(pair\; 0\; 0)\; pp)$$

3 Lists in Type Theory

We start not in Type Theory, but in System T, that is simply-typed λ-calculus with the constants :

$$\begin{array}{rcll} 0 & : & N & \\ S & : & N \to N & \\ R_T & : & T \to (N \to T \to T) \to N \to T & \quad \text{(for any type } T\text{)} \end{array}$$

and the usual reduction rules for R_T.

Question 6 Extend this with corresponding constructions for a type list_T of lists whose elements are of type T with constants nil_T and $cons_T$. You can call RL_T the recursion operator over these lists. Give the corresponding reduction rules. ◇

Solution.

$$\begin{aligned} nil_T &: \mathsf{list}_T && (1)\\ cons_T &: T \to \mathsf{list}_T \to \mathsf{list}_T && (2)\\ RL_{T,U} &: U \to (T \to \mathsf{list}_T \to U \to U) \to \mathsf{list}_T \to U && (3) \end{aligned}$$

and the reductions :

$$\begin{aligned} (RL_{T,U}\ t_0\ t_c\ nil_T) &\rhd t_0 && (4)\\ (RL_{T,U}\ t_0\ t_c\ (cons_T\ u\ l)) &\rhd (t_c\ u\ l\ (RL_{T,U}\ t_0\ t_c\ l)) && (5) \end{aligned}$$

□

Question 7 Transpose this to Martin-Löf's Type Theory (MLTT) by giving a dependent typing for this RL_T operator, so that it becomes an extension of MLTT. ⋄

Solution. With $P : N \to \mathsf{Type}$,

$$RL_{T,P} : (P\ nil_T) \to (\forall x : T.\forall l : \mathsf{list}_T.(P\ l) \to (P\ (cons_T\ x\ l)) \to \forall l : \mathsf{list}_T \to (P\ l).$$

Independently, we extend MLTT with an operator $D : N \to \mathsf{Type}$ with two reduction rules :

$$\begin{aligned} (D\ 0) &\rhd \top\\ (D\ (S\ t)) &\rhd \bot \end{aligned}$$

Question 8 Use this new operator to prove $0 =_N (S\ 0) \to \bot$ in this extension of MLTT.⋄

Solution. You can prove $0 =_N (S\ 0) \to (D\ 0) \to (D\ 1))$ and since $(D\ 0)$ is provable, you get $0 =_N (S\ 0) \to (D\ 1))$ which is identical to $0 =_N (S\ 0) \to \bot$. □

Question 9 We now want to prove $\Pi x : T\ .\ \Pi l : \mathsf{list}_T\ .\ nil_T =_{\mathsf{list}_T} (cons_T\ x\ l) \to \bot$.

Do you need additional operator to prove this or can you do with the operator D of the previous question ? How do you proceed ? ⋄

Solution. You do not need any new operator or extension. Just translate lists to numbers with

$$tr \equiv \lambda l : \mathsf{list}_T.RL_{T,N}\ 0\ \lambda\_.\lambda\_.1\ l$$

and then use the previous question to prove $tr\ nil_T \neq cons_T\ n\ l$. □

4 Surjective Pairing

One considers the following additional reduction rule for Martin-Löf's Type Theory :

$$(\pi_1(t), \pi_2(t)) \rhd_{SR} t$$

This reduction rule is know as the *surjective pairing* reduction. Note that the rule is not linear (the two occurrences of t in the left hand part need to be identical).

Question 10 Show that this rule enjoys the subject reduction property. That is, if $\Gamma \vdash (\pi_1(t), \pi_2(t)) : U$, then $\Gamma \vdash t : U$. ⋄

Solution. We remember that we have uniqueness of typing modulo conversion : if $\Gamma \vdash u : U_1$ and $\Gamma \vdash u : U_2$, then $U_1 =_\beta U_2$.

If $(\pi_1(t), \pi_2(t))_{\Sigma x:A.B}$ is well typed in Γ, then so is $\pi_1(t)$ and thus there exists A and B such that $\Gamma \vdash t : \Sigma x : A.B$.

Thus $\Gamma \vdash \pi_1(t) : A$ and $\Gamma \vdash \pi_2(t) : Bi[x \setminus \pi_1(t)]$.

Thus $\Gamma \vdash (\pi_1(t), \pi_2(t)) : \Sigma x : A.B$.

Thus $\Gamma \vdash T : \mathsf{Type}$ and $T =_\beta \Sigma x : A.B$, and thus $\Gamma \vdash t : T$.

5 Markov's Principle

In this section, we work in Martin-Löf's Type Theory (MLTT). We consider that P is a predicate over natural numbers, that is an object of type $N \to \mathsf{Type}$.

Question 11 Show that, for at least some values of P, the proposition $\neg\neg(\Sigma n : N.P\ n) \to \Sigma n : N.P\ n$ is not provable in MLTT. ⋄

Solution. Take a variable $X : \mathsf{Type}$ and $P \equiv \lambda x : N.X$. Then $\Sigma n : N.P\ n$ is equivalent to X and the principle would entail $X + \neg X$ thus giving full classical logic. □

The soviet mathematician Andrei Markov proposed a version of this proposition, weakened in order to preserve constructivity. He suggested to admit the axiom $\neg\neg(\Sigma n : N.P\ n) \to \Sigma n : N.P\ n$ but only for decidable properties, that is provided the following is provable : $\forall n : N.P\ n + \neg(P\ n)$. (Here + denotes the sum type operator in MLTT).

In other words, Markov proposed to accept the following axiom scheme, which is thus known as *Markov's principle* :

$$(\forall n : N.P\ n + \neg(P\ n)) \to \neg\neg(\Sigma n : N.P\ n) \to \Sigma n : N.P\ n.$$

Question 12 Explain informally why Markov's principle can be constructive ; that is how one could give evidence for Markov's principle in Heyting's semantics. ⋄

Solution. Evidence for (1) $\forall n : N.P\ n + \neg(P\ n)$ is a function giving evidence for $P\ n + \neg(P\ n)$ for any n.

Evidence for $\neg\neg(\Sigma n : N.P\ n)$ entails, classically, that there exists a number α for which $P\ \alpha$ is true.

So enumerating all natural numbers and checking (1) one will find α and a proof of $P\ \alpha$. □

(For the record, it is possible, but difficult, to show that Markov's principle is not provable in MLTT (or in Heyting's arithmetic).)

One proposes to extend MLTT with a specific term corresponding to Markov's principle in the Curry-Howard setting.

Given terms P, d, p, n, one has a new term $MP_P(d, p, n)$. One adds the following typing rule :

$$\frac{\Gamma \vdash P : N \to \mathsf{Type} \qquad \Gamma \vdash d : \forall n : N.P\ n + \neg(P\ n) \qquad \Gamma \vdash p : \neg\neg(\Sigma n : N.P\ n)}{\Gamma \vdash MP_P(d, p, 0) : \Sigma n : N.P\ n}$$

One suggests the following reduction rule :

$$(R_{MP}) \quad MP_P(d, p, n) \quad \rhd \quad \delta(d\ n, x.(n\ x), y.MP_P(d, p, (S\ n)))$$

Remember δ is the elimination operator for sum types, that is logical disjunction.

Question 13 Explain the idea behind this MP operator and this reduction rule. ⋄

Solution. It is precisely what is described in the response to the previous question. □

Question 14 Show that this R_{MP} reduction rule is not strongly normalizable (or in other words, that MLTT with this reduction rule in not strongly normalizable). *This should be very short.* ⋄

Solution. The reduction rule can obviously be repeated infinitely :

$$\begin{array}{rcl} MP_P(d,p,0) & \rhd & \delta(d\ n, x.(n\ x), y.MP_P(d, p, (S\ n))) \\ & \rhd & \delta(d\ 0, x.(n\ x), y.\delta(d\ 1, x.(n\ x), y.MP_P(d, p, 2))) \\ & \rhd & \delta(d\ 0, x.(n\ x), y.\delta(d\ 1, x.(n\ x), y.\delta(d\ 1, x.(n\ x), y.MP_P(d, p, 3)))) \\ & \rhd & \ldots \end{array}$$

□

Question 15 Show that the system with the R_{MP} reduction rule is not weakly normalizable either. *Hint : you may look at the next question to find the idea.* ⋄

Solution. If we are in an incoherent context with $b : \bot$, then one can use b to prove $\neg\neg\Sigma x : N.\bot$. Using the simple proof of $\forall x : N.\bot + \neg\bot$ which always returns the proof of $\neg\bot$, the operator will never find a witness and loop forever. □

One therefore suggests the following restriction : *the R_{MP} reduction can only be performed when the terms d and p are closed (that is they contain no free variable).*

Question 16 Sketch a proof of weak normalization for MLTT extended by this restricted R_{MP} reduction rule. ⋄

Solution. For any well-typed term t, we call $\#(t)$ the size of its normal form (for conventional reduction, that is without considering R_{MP}.

Suppose $\Gamma \vdash t : T$. We show by induction over $\#(t)$ that t has a normal form for the extended reduction.

We take the conventional normal form of t. Suppose it contains a R_{MP} redex $MP(d, p, 0)$ (the third argument of MP must be convertible to 0 because of the typing rule and because we have not performed any R_{MP} reduction). By induction hypothesis, we can normalize d and p. We suppose thus that d and p are closed and normal.

We can thus argue that there must exists a closed term $S^{(i)}\ 0$ sucht that $d\ S^{(i)}\ 0$ reduces to some $i(q)$ (because p is closed). Thus $MP(d, p, 0)$ reduces to $(S^{(i)}\ 0, q)$. □