Part II Types: Exercise Sheet 3

Nathanael Alcock, Dylan McDermott, Ian Orton, Shaun Steenkamp, Dima Szamozvancev, *et al.*

November 2018

Questions

1. Complete Exercises 1 and 2 from Lecture 9:
   - Show that $\neg A \vee B, A; \cdot \vdash B$ true is derivable.
   - Show that $\neg(\neg A \wedge \neg B); \cdot \vdash A \vee B$ true is derivable.
2. Give the proof (and refutation) terms corresponding to the derivations in the previous question.
3. Let $\mathsf{upc}(p) \triangleq \mu u : A.\ \langle p \mid_A u \rangle$ be a proof (and refutation) term from the calculus presented in Lectures 9 and 10.
   (a) Show that $p : A; \cdot \vdash \mathsf{upc}(p) : A$ true.
   (b) Show that for all $k : \neg A$ we have $\langle \mathsf{upc}(p) \mid_A k \rangle \mapsto \langle p \mid_A k \rangle$.
   (c) Terms $p : A$ true correspond to proofs of A. Describe, in English, the proof that corresponds to $\mathsf{upc}(p)$ with respect to the proof corresponding to p.
4. Complete Exercises 1 and 2 from Lecture 10:
   - Give the embedding (i.e., the e° and k° translations) of classical into intuitionistic logic for the Gödel-Gentzen translation. You just need to give the embeddings for sums, since that is the only case different from the lectures.
   - Using the intuitionistic (λ-) calculus extended with continuations, give a typed term proving Peirce's law:
   $$((X \to Y) \to X) \to X$$
5. Using the `amb` primitive from Lecture 11 implement a function:

   ```
   eq-at : α list -> α list -> int * int
   ```

   such that `eq-at xs ys` returns `(i,j)` if `nth(xs,i) = nth(ys,j)` and fails otherwise. You may assume the existence of any helper functions without definition.
6. What logical operator do Π-types (or *dependent products*) correspond to? Justify your answer.
7. Using the dependent type theory introduced in Lecture 12 show that if $\Gamma \vdash A$ type then the following typing judgement holds:
   $$\Gamma \vdash \mathsf{sym}_A : \Pi x : A.\, \Pi y : A.\, ((x = y : A) \to (y = x : A))$$
   where
   $$\mathsf{sym}_A \triangleq \lambda x : A.\, \lambda y : A.\, \lambda p : (x = y : A).\, \mathsf{subst}[z : A.\, (z = x : A)](p, \mathsf{refl}\, x)$$
   and $X \to Y$ is shorthand for $\Pi x : X. Y$ if x does not appear in Y.
8. Define terms with the following types:
   (a) $\Gamma \vdash \mathsf{trans}_A : \Pi x : A.\, \Pi y : A.\, \Pi z : A.\, ((x = y : A) \to (y = z : A) \to (x = z : A))$
   (b) $\Gamma \vdash \mathsf{cong}_{A,B} : \Pi x : A.\, \Pi y : A.\, \Pi f : (A \to B).\, ((x = y : A) \to (f x = f y : B))$

   assuming that $\Gamma \vdash A$ type and $\Gamma \vdash B$ type.

9. Consider types $\Gamma \vdash A$ type and $\Gamma, x : A \vdash B$ type. If we have terms a_1 and a_2 and a proof that they are equal, $\Gamma \vdash p : (a_1 = a_2 : A)$, then the types $[a_1/x]B$ and $[a_2/x]B$ should also be "equal" in some sense. And so, given $\Gamma \vdash b_1 : [a_1/x]B$ and $\Gamma \vdash b_2 : [a_2/x]B$ we might want to consider the type of equalities between b_1 and b_2.

   (a) Show that the following rule is not (in general) derivable:

$$\frac{\Gamma \vdash A\ \mathsf{type} \quad \Gamma, x : A \vdash B\ \mathsf{type} \quad \Gamma \vdash a_2 : A \quad \Gamma \vdash p : (a_1 = a_2 : A) \quad \Gamma \vdash b_1 : [a_1/x]B \quad \Gamma \vdash b_2 : [a_2/x]B}{\Gamma \vdash (b_1 = b_2 : [a_1/x]B)\ \mathsf{type}}$$

   (b) Define the type of *heterogeneous equalities* like so:

$$(b_1 \approx b_2 : B\,\mathsf{over}\,p) \triangleq (\mathsf{subst}[x : A.\, B](p, b_1) = b_2 : [a_2/x]B)$$

   Show that the following rule is admissible:

$$\frac{\Gamma \vdash A\ \mathsf{type} \quad \Gamma, x : A \vdash B\ \mathsf{type} \quad \Gamma \vdash a_2 : A \quad \Gamma \vdash p : (a_1 = a_2 : A) \quad \Gamma \vdash b_1 : [a_1/x]B \quad \Gamma \vdash b_2 : [a_2/x]B}{\Gamma \vdash (b_1 \approx b_2 : B\,\mathsf{over}\,p)\ \mathsf{type}}$$

   (c) Define a term $\mathsf{hrefl}\,b$ such that the following rule is derivable:

$$\frac{\Gamma \vdash A\ \mathsf{type} \quad \Gamma, x : A \vdash B\ \mathsf{type} \quad \Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B}{\Gamma \vdash \mathsf{hrefl}\,b : (b \approx b : B\,\mathsf{over}\,(\mathsf{refl}\,a))}$$

Extension

10. Download and install Agda and try out some of the examples from the lectures:

`https://agda.readthedocs.io/en/latest/getting-started/index.html`

   If you need help or have any questions, the mailing list or #agda on the Freenode IRC network are a good source.

11. (a) Given your answer to Question 6, what logical operator are we still missing?

   (b) Extend the syntax of the dependently typed language introduced in the lectures with this dual of Π-types (also called *Σ-types* or *dependent sums*) and give suitable typing rules for them. (Research "dependent sums" if you are unsure.)

   (c) In first-order logic, the axiom of choice can be stated as:

$$(\forall x \in A.\, \exists y \in B.\, P(x, y)) \implies (\exists f : A \to B.\, \forall x \in A.\, P(x, f(x)))$$

   Given $\Gamma \vdash A$ type and $\Gamma \vdash B$ type, give a type $\Gamma \vdash \mathsf{AC}$ type corresponding to the axiom of choice.

   (d) Define a term $\Gamma \vdash \mathsf{ac} : \mathsf{AC}$.

12. The notes state that the rule for equality elimination is "not the most general form". Consider the following alternative elimination rule:

$$\frac{\Gamma \vdash A\ \mathsf{type} \quad \Gamma \vdash p : (a_1 = a_2 : A) \quad \Gamma, x : A, y : (a_1 = x) \vdash B\ \mathsf{type} \quad \Gamma \vdash b : [a_1/x, (\mathsf{refl}\,a_1)/y]B}{\Gamma \vdash J[x : A, y : (a_1 = x).\, B](p, b) : [a_2/x, p/y]B}$$

   such that $J[x : A, y : (a_1 = x).\, B](\mathsf{refl}\,a_1, b) \equiv b$.

   (a) Show that subst can be derived from J.

   (b) For sym_A defined as in Question 7 define:

$$\mathsf{SymInv} \triangleq \Pi x : A.\, \Pi y : A.\, \Pi p : (x = y : A).\, (\mathsf{sym}_A\, y\, x\, (\mathsf{sym}_A\, x\, y\, p) = p : (x = y : A))$$

   Show that $\Gamma \vdash \mathsf{SymInv}$ type and define a term $\Gamma \vdash \mathsf{symInv} : \mathsf{SymInv}$. You will need to use J.