

Exam Logical Verification

April 2, 2007

There are six (6) exercises.

Answers may be given in Dutch or English. Good luck!

Exercise 1. This exercise is concerned with first-order propositional logic (`prop1`) and simply typed λ -calculus ($\lambda \rightarrow$).

- a. Give a derivation in `prop1` showing that $((B \rightarrow A \rightarrow B) \rightarrow A) \rightarrow A$ is a tautology.

(5 points)

- b. Give the type derivation in $\lambda \rightarrow$ corresponding to the proof in (1a.).

(5 points)

- c. Give closed inhabitants of the following types:

$$\begin{aligned} B &\rightarrow ((A \rightarrow B) \rightarrow C) \rightarrow C \\ A &\rightarrow (B \rightarrow A) \rightarrow A \\ (A \rightarrow A) &\rightarrow A \rightarrow A \\ A &\rightarrow (A \rightarrow A) \rightarrow A \end{aligned}$$

(NB: You do not have to give a type derivation.)

(5 points)

Exercise 2. This exercise is concerned with first-order predicate logic (`pred1`) and λ -calculus with dependent types (λP).

NB: the scope of the quantifier is as far to the right as possible.

- a. Give a derivation in `pred1` showing that $\forall x. P(x) \rightarrow (\forall y. P(y) \rightarrow A) \rightarrow A$ is a tautology.

(5 points)

- b. Give the λP -term corresponding to the formula of (2a.).

(5 points)

- c. The introduction rule for universal quantification is:

$$\frac{A}{\forall x.A} IV$$

Give an example of an application of this rule which is incorrect because the side-condition is violated.

(5 points)

Exercise 3. This exercise is concerned with second-order propositional logic (`prop2`) and polymorphic λ -calculus ($\lambda 2$).

- a. Give a derivation in `prop2` showing that

$$\forall a. a \rightarrow \forall b. (a \rightarrow b) \rightarrow b$$

is a tautology.

(5 points)

- b. Give the $\lambda 2$ type corresponding to the formula of (3a.).

(5 points)

- c. Give a closed inhabitant of the type found in (3b.).

(5 points)

Exercise 4. This exercise is concerned with impredicative encodings in $\lambda 2$. Assume $A : \star$, $B : \star$, and $C : \star$.

- a. Consider the following encoding of disjunction:

$$\text{Or } A B = \Pi c : \star. (A \rightarrow c) \rightarrow (B \rightarrow c) \rightarrow c$$

Use a term $M : A$ to construct an inhabitant of $\text{Or } A B$.

(5 points)

- b. Assume $P : \text{Or } A B$. Explain how we can construct an inhabitant of C , possibly using additional terms.

(5 points)

- c. We define the booleans \mathbf{B} and *true* (\mathbf{T}) and *false* (\mathbf{F}) as follows:

$$\mathbf{B} = \Pi a : \star. a \rightarrow a \rightarrow a$$

$$\mathbf{T} = \lambda a : \star. \lambda x : a. \lambda y : a. x$$

$$\mathbf{F} = \lambda a : \star. \lambda x : a. \lambda y : a. y$$

Give a definition of negation in $\lambda 2$.

(5 points)

Exercise 5.

- a. Give the two detours of `prop2`.
(5 points)
- b. Show how the detour for universal quantifications in `prop2` corresponds to a redex in $\lambda 2$.
(5 points)
- c. Explain how the product rule for λP

$$\frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash B : \square}{\Gamma \vdash \Pi x:A. B : \square}$$

is used to infer that the type of `natlist_dep` can be typed.
(5 points)

Exercise 6. This exercise is concerned with Coq.

- a. Consider the definition of an inductive predicate for `even`:

```
Inductive even : nat -> Prop :=
| even_zero      : even 0
| even_greater   : forall n:nat, even n -> even (S (S n)).
```

What is the type of `even 0`?

Give an inhabitant of `even 0`.

Give an inhabitant of `even 2`.

(5 points)

- b. Give an inductive definition of polymorphic binary trees with labels on the nodes (and no labels on the leaves).
(5 points)

- c. Give a recursive definition of a function `count` that takes as input a polymorphic binary tree and gives as output the number of nodes of the tree.

NB: You may assume the inductive type `nat` of natural numbers built from constructors for zero and for successor, and a function `plus` for addition of two natural numbers.

(5 points)

The final note is (the total amount of points plus 10) divided by 10.