Exam Logical Verification

January 19, 2005

There are six (6) exercises.
Answers may be given in Dutch or English. Good luck!

Exercise 1. This exercise is concerned with first-order minimal propositional logic and simply typed λ-calculus.

a. Show that the formula $(A \to B \to C) \to B \to A \to C$ is a tautology.

(5 points)

b. Give the type derivation in simply typed λ-calculus corresponding to the proof of 1a.

(5 points)

c. Replace in the following three terms the ?'s by simple types, such that we obtain typable λ-terms. (*NB: it is not asked to give the type derivations.*)

$\lambda x : ?.\, \lambda y : ?.\, \lambda z : ?.\, x\,(y\,z)$

$\lambda x : ?.\, \lambda y : ?.\, \lambda z : ?.\, x\,(z\,y)$

$\lambda x : ?.\, \lambda y : ?.\, \lambda z : ?.\, (x\,z)\,y$

(6 points)

d. Give a proof of $(A \to A) \to A \to A$ with a detour.

(4 points)

Exercise 2. This exercise is concerned with inductive types and recursive function definitions in Coq.

a. Give the inductive definition of the datatype `boollist` of lists of booleans (the type of booleans is called `bool`).

(5 points)

b. Give the type of `boollist_ind` which is used to give proofs by induction on these lists of booleans.

(5 points)

c. Give the definition of a recursive function

```
nat_of_boollist : boollist -> nat
```

which interprets a list of booleans as a natural number in binary notation (`false` for 0 and `true` for 1) but *with the bits in reverse.* For example `nat_of_boollist (cons true (cons true (cons false (cons true (cons false nil)))))` is the binary number 01011, which has decimal value 11.

In your definition you can use the functions `plus` for addition and `mult` for multiplication of natural numbers.

(5 points)

Exercise 3. This exercise is concerned with first-order predicate logic.

a. Which are the two rules of first-order intuitionistic predicate logic that have a variable condition? (*NB: you do not need to describe the variable conditions, just mentioning the names of the rules is sufficient.*)

(3 points)

b. Show that $(\exists x.\,(P(x) \vee Q(x))) \rightarrow \big((\exists x.\,P(x)) \vee (\exists x.\,Q(x))\big)$ is a tautology of first-order intuitionistic predicate logic.

(9 points)

Exercise 4. This exercise is concerned with program extraction.

a. What is the Brouwer-Heyting-Kolmogorov interpretation?

(5 points)

b. If one proves in Coq

```
forall n : nat, ~(n = 0) -> {m : nat | S m = n}
```

then one can extract a function from the proof. What is the type of this function? What does it compute?

(5 points)

Exercise 5. This exercise is concerned with λ-calculus with dependent types (λP).

a. How does one write the function type $A \rightarrow B$ in the form of a dependent product type?

(3 points)

b. Define (using Coq syntax) a dependent type

```
Is_true : bool -> Prop
```

such that `Is_true b` is inhabited if and only if `b` is equal to `true`.

(5 points)

c. Replace in the following judgment the two ?'s

$$b : ? \vdash \Pi x : b. * : ?$$

such that it becomes a correct λP judgment. (*For your convenience you will find the λP type derivation rules on the next page. NB: you do not need to give a type derivation of this judgment.*)

(5 points)

Exercise 6. This exercise is concerned with second-order propositional logic and polymorphic λ-calculus ($\lambda 2$).

a. Show that $b \to \forall a.\, (a \to b)$ is a tautology of second-order minimal propositional logic.

(5 points)

b. What is the $\lambda 2$ term that corresponds to the proposition in question 6a. (*NB: you do <u>not</u> need to give the $\lambda 2$ term that corresponds to the <u>proof</u>.*)

(3 points)

c. Given that b has type $*$, what is the type of $\Pi a : *.\, (a \to b)$.

(3 points)

d. Give the $\lambda 2$ derivation of the type judgment

$$b : * \vdash * : \Box$$

(*You will find the $\lambda 2$ type derivation rules on the next page.*)

(3 points)

e. Give the $\lambda 2$ derivation of the type judgment

$$b : *,\ a : * \vdash a \to b : *$$

(*You may abbreviate type derivations that are answers to earlier questions in the style:*

$$\frac{6d}{b : * \vdash * : \Box}$$

)

(3 points)

f. Give the $\lambda 2$ derivation of the type judgment that corresponds to your answer to question 6c.

(3 points)

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

Derivation rules of the Pure Type Systems λP and $\lambda 2$

In these rules the variable s ranges over the set of sorts $\{*, \Box\}$. The product rule differs between λP and $\lambda 2$.

axiom

$$\frac{}{\vdash * : \Box}$$

application

$$\frac{\Gamma \vdash M : \Pi x : A.B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

abstraction

$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.B : s}{\Gamma \vdash \lambda x : A.M : \Pi x : A.B}$$

product (λP)

$$\frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x : A.B : s}$$

product ($\lambda 2$)

$$\frac{\Gamma \vdash A : s \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x : A.B : *}$$

weakening

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$$

variable

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$$

conversion

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \text{ where } B =_\beta B'$$