Exam Logical Verification

May 31, 2013

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

Exercise 1. *(5+5+6 points)*
This exercise is concerned with first-order propositional logic (prop1) and simply typed λ-calculus ($\lambda\!\!\rightarrow$).

a) Show that the following formula is a tautology of minimal prop1:
$(A \to A \to B) \to (C \to A) \to (C \to B)$.

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

c) Give, if possible, closed inhabitants in $\lambda\!\!\rightarrow$ of the following types:
$((B \to A \to B) \to A) \to A$
$A \to A \to B$
$A \to B \to A$

Exercise 2. *(5+3+5+3 points)*
This exercise is concerned with first-order predicate logic (pred1) and λ-calculus with dependent types (λP).

a) Show that the following formula is a tautology of minimal pred1:
$(\forall x.\,(P(x) \to Q(x))) \to \forall x.\,((Q(x) \to R(x)) \to P(x) \to R(x))$.

b) Give a λP-term corresponding to the formula in 2a.
(Use Terms for the domain that is quantified over.)

c) Give a closed inhabitant in λP of the answer to 2b.

d) Consider the following question Q:
is the formula $\forall x.\,P(x) \to (\forall y.\,P(y) \to A) \to A$ a tautology?

What is the counterpart of the question Q in λ-calculus/type theory?

Exercise 3. *(5+3+5+3 points)*
This exercise is concerned with second-order propositional logic (prop2) and polymorphic λ-calculus ($\lambda 2$).

a) Show that the following formula is a tautology of minimal prop2:
$a \to \forall b.\, ((\forall c.\, a \to c) \to b)$.

b) Give the $\lambda 2$-term corresponding to the formula in 3a.

c) Give a closed inhabitant in $\lambda 2$ of the answer to 3b.

d) What is the proof checking problem? Is it decidable for $\lambda 2$?

Exercise 4. *(2+4+4+6 points)*
This question is concerned with various typing issues.

a) The typing rules we considered use $*$ and $\Box$.
What is/are the counterpart(s) in Coq of $*$ and of $\Box$?

b) We define $\mathsf{and}\, C\, D$ with $C : *$ and $D : *$ in $\lambda 2$ as follows:

$$\mathsf{and}\, C\, D := \Pi a : *.\, (C \to D \to a) \to a$$

Assume an inhabitant $P : \mathsf{and}\, C\, D$. Give an inhabitant of C (provide the informal typing derivation).

c) Give the polymorphic identity in $\lambda 2$.

Next, assume $\mathsf{nat} : *$ and show how the polymorphic identity is instantiated to the identity on nat using application and β-reduction.

d) Consider the typing rules for a product $\Pi x{:}A.\, B$:

$$\textit{product } (\lambda \to \textit{ and } \lambda P \textit{ and } \lambda 2) \qquad \frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x{:}A.\, B : *}$$

$$\textit{product } (\lambda P) \qquad \frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : \Box}{\Gamma \vdash \Pi x{:}A.\, B : \Box}$$

$$\textit{product } (\lambda 2) \qquad \frac{\Gamma \vdash A : \Box \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x{:}A.\, B : *}$$

We assume $\mathsf{nat} : *$ and $\mathsf{vec} : \mathsf{nat} \to *$. Explain informally how the following products can be typed using the appropriate product rule.

(i) $\mathsf{nat} \to \mathsf{nat}$

(ii) $\Pi a{:} *\,.\, a \to a$

(iii) $\Pi n{:}\mathsf{nat}.\, \mathsf{vec}\, n$

Exercise 5. (*5+4+4 points*)
This exercise is concerned with inductive datatypes in Coq.

a) Give the definition of an inductive datatype `three` with exactly three elements. Also, give the type of `three_ind` for the induction principle on `three_ind`.

b) Give the inductive definition of the datatype `natsnoclist` of lists of natural numbers, but where the constructor for adding an element to a list adds this element at the end.

c) Give the type of `natsnoclist_ind` for the induction principle on `natsnoclist`.

Exercise 6. (*4+5+4 points*)
This exercise is concerned with inductive predicates in Coq.

a) Consider the inductive predicate for less-than-equal in Coq:

```
Inductive le (n:nat) : nat -> Prop :=
| le_n : le n n
| le_S : forall m:nat , le n m -> le n (S m) .
```

Prove that $1 \leq 2$, that is, give an inhabitant of `le (S O) (S (S O))`.

b) Give the definition of an inductive predicate `evenlist` on the usual datatype `natlist` (of lists of natural numbers) such that `evenlist l` holds exactly if the list `l` has an even number of elements.

c) Complete the following definition of conjunction in Coq:

```
Inductive and (A : Prop) (B : Prop) : Prop :=
```

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