Exam Logical Verification

August 25, 2010

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

(5 points)

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

(5 points)

c. Give three different closed inhabitants of the type $A \to (A \to A) \to A$.

(5 points)

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

a. Give the two detour elimination rules for minimal pred1.

(5 points)

b. Assume a term a.

Give a proof of $(\forall x. P(x)) \to P(a)$ with a detour for $\forall$.

(5 points)

c. Give the λP-term corresponding to the formula $(\forall x. P(x)) \to P(a)$ from 2b.

(5 points)

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

(5 points)

Exercise 3. 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:

$$\forall a.\, a \to \forall b.\, b \to a$$

(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)

d. Show how the polymorphic identity is instantiated to the identity on $\mathsf{nat} : \star$ by means of application and β-reduction.

(5 points)

Exercise 4. This exercises is concerned with encodings.

a. Give a definition of *false* in prop2 and show that the elimination rule for *false* (stating that from *false* follows any proposition) can be derived.

(5 points)

b. We define $\mathsf{and}\, A\, B$ with $A : *$ and $B : *$ in $\lambda 2$ as follows:

$$\mathsf{and}\, A\, B := \Pi c : *.\, (A \to B \to c) \to c$$

Assume an inhabitant $P : \mathsf{and}\, A\, B$. Give an inhabitant of A.

(5 points)

Exercise 5. This definition is concerned with inductive data-types in Coq.

a. Give the definition of an inductive data-type `natpair` of pairs of natural numbers. (You can use the data-type `nat` of natural numbers.)

(5 points)

b. Give the type of `natpair_ind`, for the induction principle for your data-type from 5a.

(5 points)

Exercise 6. This exercise is concerned with inductive predicates in Coq.

a. Consider the following inductive predicates:

```
Inductive ev : nat -> Prop :=
| ev0 : ev 0
| evS : forall n:nat , odd n -> ev (S n)
with odd : nat -> Prop :=
| oddS : forall n:nat , ev n -> odd (S n) .
```

Give inhabitants of the following:

```
ev 0
odd 1
ev 2
```

(6 points)

b. Complete the following definition of the inductive predicate `even`:

```
Inductive even : nat -> Prop :=
| even0  :
| evenSS :
```

(5 points)

c. Complete the following definition of conjunction:

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

(4 points)

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