

Exam Logical Verification 2014-2015

Thursday July 9, 2015, 12.00-14.45

6 exercises

Answers may be given in English or in Dutch.



Opgave 1. (*4+4+4+4 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**:
 $((A \rightarrow B) \rightarrow C) \rightarrow B \rightarrow C$.
- b) Give the type derivation in $\lambda \rightarrow$ corresponding to the proof of 1a.
- c) Give three different closed inhabitants in $\lambda \rightarrow$ of the following type:

$$B \rightarrow (B \rightarrow A) \rightarrow (B \rightarrow B) \rightarrow A$$

- d) Explain the Curry-Howard-de Bruijn correspondence between terms in $\lambda \rightarrow$ and proofs in **prop1** in detail.

Exercise 2. (*4+4+4+4+4 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**:
 $\forall x. (P(x) \rightarrow (\forall y. P(y) \rightarrow A) \rightarrow A)$.
- b) Give a λP -term corresponding to the formula in 2a.
- c) Give a closed inhabitant in λP of the answer to 2b.
- d) What is the type checking problem? Is it decidable for λP ?
- e) What is the type inhabitation problem? Is it decidable for λP ?

Exercise 3. (*4+4+4 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**:

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

- 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.

Exercise 4. (4+4+5 points)

This exercise is concerned with various typing and coding issues.

- a) Give the polymorphic identity in $\lambda 2$.
- b) First-order propositional logic can be defined in Coq as follows:

```
Parameter prop : Set.
Parameter imp : prop -> prop -> prop.
(* T expresses if a proposition in prop is valid
   if (T p) is inhabited then p is valid
   if (T p) is not inhabited then p is not valid *)
Parameter T : prop -> Prop.
```

Give variables modelling the introduction rule for implication and the elimination rule for implication.

- c) What is the (impredicative) definition of the type of the natural numbers, given as Church numerals, in $\lambda 2$?

What is a term in this type that corresponds to the number 2?

Exercise 5. (4+4+4 points)

This exercise is concerned with inductive data-types in Coq. The constructors of the type `nat` are called `0` and `S`.

- a) Give the definition of an inductive data-type `natpair` of pairs of natural numbers (with `nat` the type of natural numbers).
- b) Give the induction principle for `natpair`.
- c) Give two different definitions of data-types with zero elements.

Exercise 6. (4+4+4+5 points)

This exercise is concerned with inductive predicates in Coq. The constructors of the type `nat` are called `0` and `S`.

- 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) .
```

Give if possible an inhabitant of the following, if it is not possible explain shortly why not:

```
le 0 (S 0)
le (S 0) 0
```

- b) Give the definition in Coq of an inductive predicate **tre** on natural numbers that holds exactly if the number can be divided by 3.
- c) Give the Coq definition of an inductive predicate of type

```
even : nat -> Prop
```

that says whether a natural number is even or not.

- d) 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.