

Exam Logical Verification

May 1, 2003

There are six (6) exercises.

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

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

- a. Show that the formula $((A \rightarrow B \rightarrow A) \rightarrow B) \rightarrow B$ is a tautology of first-order minimal propositional logic.
(5 points)
- b. Give the (formal) type derivation in simply typed λ -calculus corresponding to the proof of 1a.
(5 points)
- c. Give the correspondence between terms in simply typed λ -calculus and proofs in first-order minimal propositional logic in detail.
(6 points)
- d. What is the provability problem in first-order minimal propositional logic? What is the corresponding problem in simply typed λ -calculus?
(4 points)

Exercise 2. This exercise is concerned with polymorphic lambda-calculus and second-order minimal propositional logic.

- a. What is the type of the polymorphic identity?
(5 points)
- b. Show how the polymorphic identity is used to get the identity on the type `nat` of natural numbers.
(5 points)
- c. Give the polymorphic version of the following function:
 $\lambda f:\text{nat} \rightarrow \text{bool} \rightarrow \text{nat}. \lambda x:\text{nat}. \lambda y:\text{bool}. f\ x\ y.$
(In the polymorphic variant neither `nat` nor `bool` occurs.)
(5 points)

d. Explain why the following proof is not correct:

$$\frac{\frac{\exists a. a \rightarrow b \quad \frac{[a \rightarrow b^x]}{(a \rightarrow b) \rightarrow (a \rightarrow b)}}{a \rightarrow b}}{I[x] \rightarrow} E\exists$$

(5 points)

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

a. Show that the following formula is a tautology of first-order predicate logic: $(\forall x. A \rightarrow P(x)) \rightarrow A \rightarrow \forall y. P(y)$.

(The variable x doesn't occur in A .)

(5 points)

b. Give the two different detours in minimal first-order predicate logic in schematic form.

(5 points)

Exercise 4. This exercise is concerned with Coq.

a. Give the inductive type `bintree` of binary trees with natural numbers (type `nat`) on the leaves.

(5 points)

b. Give the type of `nat_ind`, for induction on `nat`.

(5 points)

c. Let `list` be the inductive type of lists of type `A`:

```
Inductive list : Set :=
  nil : list | cons : A -> list -> list.
```

Explain the definition of the following predicate `P`:

```
Fixpoint P [a:A;l:list] : Prop :=
  Cases l of nil => False
    | (cons b m) => (b=a) /\ (P a m) end .
```

(5 points)

d. We continue in the setting of 4c.

In addition we have the following function:

```
Definition s := [l,m:list](a:A)(P a l) -> (P a m).
```

Explain this definition.

(5 points)

Exercise 5. This exercise is concerned with sequent calculus. The rules of the sequent calculus are given in the appendix.

- a. What is the interpretation of a sequent $A_1, \dots, A_m \vdash B_1, \dots, B_n$?
(2 points)
- b. Prove the following sequent:
 $\vdash ((A \rightarrow B) \wedge (A \rightarrow C)) \rightarrow (A \rightarrow (B \wedge C))$.
(4 points)
- c. Prove the following sequent:
 $\vdash \forall x.P(x) \vee \exists x.\neg P(x)$.
(4 points)

Exercise 6. This exercise is concerned with PVS.

- a. Consider the abstract datatype specification for stacks:

```
stack [t:TYPE] : DATATYPE
BEGIN

empty : emptystack?
push(top:t, pop:stack) : nonemptystack?

END stack
```

Explain the notions of constructors, recognizers, and accessors.

(6 points)

- b. What is a predicate in PVS? Give an example of a predicate subtype.
(4 points)

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