Exam Logical Verification

January 15, 2003

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

Exercise 1.

a. Show that the formula $((A \to B) \to (C \to D)) \to C \to B \to D$ is a tautology of first-order minimal propositional logic. (Give a proof in natural deduction with all assumptions canceled.)

   (4 points)

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

   (4 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 inhabitation problem in simply typed λ-calculus?
   What is the corresponding problem in first-order minimal propositional logic?

   (4 points)

Exercise 2.

a. Give the polymorphic identity, its type, and the formula corresponding to this type.

   (6 points)

b. Give the polymorphic version of the following function:
   $\lambda f{:}\mathsf{nat} \to \mathsf{bool}.\, \lambda g{:}\mathsf{bool} \to \mathsf{nat}.\, \lambda x{:}\mathsf{nat}.\, g\,(f\,x)$.

   (In the polymorphic variant neither nat nor bool occurs.)

   (5 points)

c. Explain briefly the principle of program extraction.

   (4 points)

Exercise 3.

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

(5 points)

b. Indicate the error(s) in the following proof:

$$
\begin{prooftree}
\AxiomC{$[\exists x.P(x,y)^u]$}
\AxiomC{$[P(x,y)^v]$}
\RightLabel{$I[v]\to$}
\UnaryInfC{$P(x,y) \to P(x,y)$}
\RightLabel{$E\exists$}
\BinaryInfC{$P(x,y)$}
\RightLabel{$I\forall$}
\UnaryInfC{$\forall x.P(x,y)$}
\RightLabel{$I\forall$}
\UnaryInfC{$\forall y.\forall x.P(x,y)$}
\RightLabel{$I[u]\to$}
\UnaryInfC{$\exists x.P(x,y) \to \forall y.\forall x.P(x,y)$}
\end{prooftree}
$$

(6 points)

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

(4 points)

Exercise 4. This exercise is concerned with Coq.

a. Give the inductive type `natlist` of finite lists of natural numbers.

(4 points)

b. Give the type of `natlist_ind`, for induction on `natlist`.

(4 points)

c. Consider the following definition in Coq:

```
Inductive positive : Set :=
  build_odd  : positive -> positive
| build_even : positive -> positive
| one        : positive.
```

Explain how this defines the positive integers (i.e. $\{1, 2, 3, \ldots\}$).

(Hint: postive integers are even or odd.)

(4 points)

d. In addition to the type `positive` given in 4c we now also have:

```
Inductive Z : Set :=
  ZERO : Z | POS : positive -> Z | NEG : positive -> Z.
```

Explain how this defines the integers.

(4 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 $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$?

   (2 points)

b. Prove the following sequent: $\vdash A \wedge (B \vee C) \rightarrow (A \wedge B) \vee (A \wedge C)$.

   (4 points)

c. Prove the following sequent: $\vdash \forall x.P(x) \rightarrow \neg\exists y.(\neg P(y))$

   (4 points)

Exercise 6. This exercise is concerned with PVS.

a. Give an abstract datatype specification of the type consisting of the finite lists where the elements are of some unspecified type t. That is, complete the following:

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

END lists
```

   (The precise syntax is not important. It is important to make clear what are the constructors, accessors, and recognizers.)

   (6 points)

b. The following is a (naive) specification of the integers.

```
integers: THEORY
BEGIN
int    : TYPE
zero   : int
succ   : int -> int
pred   : int -> int
END integers
```

   What axiom(s) is (are) natural to add? Why?

   (6 points)

c. 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.

Appendix: sequent calculus rules for first-order predicate logic

1. The rule *propositional axiom*:

$$\Gamma, A \vdash A, \Delta$$

2. The rules for *implication*:

$$\frac{B, \Gamma \vdash \Delta \qquad \Gamma \vdash A, \Delta}{A \to B, \Gamma \vdash \Delta} \; L\to$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta} \; R\to$$

3. The rules for *conjunction*:

$$\frac{A, B, \Gamma \vdash \Delta}{A \land B, \Gamma \vdash \Delta} \; L\land$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \land B, \Delta} \; R\land$$

4. The rules for *disjunction*:

$$\frac{A, \Gamma \vdash \Delta \qquad B, \Gamma \vdash \Delta}{A \lor B, \Gamma \vdash \Delta} \; L\lor$$

$$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \lor B, \Delta} \; R\lor$$

5. The rules for *negation*:

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \; L\neg$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \; R\neg$$

6. The rules for *universal quantification*:

$$\frac{\Gamma, A[x := M] \vdash \Delta}{\Gamma, \forall x.\, A \vdash \Delta} \; L\forall$$

Here M is a term.

$$\frac{\Gamma \vdash A[x := y], \Delta}{\Gamma \vdash \forall x.\, A, \Delta} \; R\forall$$

Here y is a fresh variable (not occurring in Γ and Δ).

7. The rules for *existential quantification*:

$$\frac{\Gamma, A[x := y] \vdash \Delta}{\Gamma, \exists x.\, A \vdash \Delta} \; L\exists$$

Here y is a fresh variable (not occurring in Γ and Δ).

$$\frac{\Gamma \vdash A[x := M], \Delta}{\Gamma \vdash \exists x.\, A, \Delta} \; R\exists$$

Here M is a term.

8. The *weakening rule*:

$$\frac{\Gamma_1 \vdash \Delta_1}{\Gamma_2 \vdash \Delta_2} \; w$$