

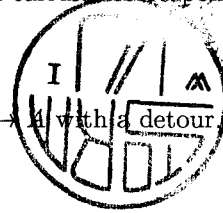
Tentamen Toegepaste Logica

16 januari 2002

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

Exercise 1.

- Show that the formula $(A \rightarrow A \rightarrow B) \rightarrow A \rightarrow C \rightarrow B$ is a tautology of first-order minimal propositional logic. (Give a proof in natural deduction with all assumptions canceled.)
(5 points)
- Give the (formal) type derivation in simply typed λ -calculus corresponding to the proof of 1a.
(5 points)
- Give a natural deduction proof of $A \rightarrow (B \rightarrow A) \rightarrow A$ with a detour and with all assumptions canceled.
(5 points)
- Give the typing derivation corresponding to the proof of 1c, and reduce the term to β -normal form.
(5 points)



Exercise 2.

- What is the interpretation of a sequent $A_1, \dots, A_m \vdash B_1, \dots, B_n$?
(2 points)
- Explain the intuition of the left rule for existential quantification in sequent calculus:

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

with y a fresh variable (not occurring in Γ, Δ).

(3 points)

- c. Give a derivation of $\vdash A \wedge (B \vee C) \rightarrow (A \wedge B) \vee (A \wedge C)$.

(5 points)

- d. For first-order propositional sequent calculus in PVS, we use two commands: **flatten**, corresponding to the rules of the sequent calculus where reading upwards a sequent is transformed into one sequent, and **split**, corresponding to the rules of the sequent calculus where reading upwards a sequent is transformed into two sequents.

Give the proof tree for the PVS derivation of

$\vdash A \wedge (B \vee C) \rightarrow (A \wedge B) \vee (A \wedge C)$.

(You may give a refinement of the PVS proof tree, using more steps where PVS uses only one step.)

(5 points)

Exercise 3.

- a. In PVS specifications can be written using the axiomatic and the definitional approach. Explain the two approaches and explain a drawback of each of them.

(5 points)

- b. 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.

(5 points)

- c. Explain the use of predicate subtypes in PVS.

(5 points)

Exercise 4.

- a. Consider the rule for introduction of the universal quantifier:

$$\frac{A}{\forall x. A} IV$$

with x not in uncanceled assumptions.

Explain the side condition and give an example that shows what can go wrong.

(5 points)

- b. Show that the formula $(\forall x. \forall y. (R(x, y) \rightarrow \neg R(y, x))) \rightarrow \forall z. \neg R(z, z)$ is a tautology of first-order predicate logic. (Give a derivation in (intuitionistic) natural deduction with all assumptions canceled.)

(5 points)



Exercise 5.

- a. Give the definition of the inductive type `nat` that is built in in Coq.
(5 points)
- b. Give your own definition in Coq of an inductive type `natpair : Set`, representing the set of pairs of natural numbers. (Hint: use one constructor `Pair`; the pair of n and m is then denoted in Coq by `(Pair n m)`.)
(5 points)
- c. What is the type of a predicate (i.e. a propositional function) on `natpair`?
(5 points)
- d. Give an inductive definition in Coq of the predicate `ordered` on `natpair`, such that we have in Coq that `(ordered (Pair n m))` if and only if `(leq n m)`, with `leq` the 'less than or equal' predicate that is built in in Coq.
(5 points)

Exercise 6.

This exercise is concerned with $\lambda 2$, the polymorphic λ -calculus.

Consider the operation `NatTwice` that, when applied to a function f of type $\text{Nat} \rightarrow \text{Nat}$ forms the composition of f with itself. So we have

$$\text{NatTwice } f \ n = f(f \ n)$$

- a. Give an explicit definition of `NatTwice` as a term of the simply typed λ -calculus, and give its type.
(5 points)
- b. Give a definition of `PolyTwice`, the polymorphic version of `NatTwice`, as a term of $\lambda 2$, and give its type.
(5 points)

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

Appendix: rules of first-order propositional sequent calculus

$$\begin{array}{c}
 \Gamma, A \vdash A, \Delta \\
 \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta} R \rightarrow \\
 \frac{B, \Gamma \vdash \Delta \quad \Gamma \vdash A, \Delta}{A \rightarrow B, \Gamma \vdash \Delta} L \rightarrow \\
 \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} R \wedge \\
 \frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} L \wedge \\
 \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta} R \vee \\
 \frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta} L \vee \\
 \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} R \neg \\
 \frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} L \neg \\
 \frac{\Gamma_1 \vdash \Delta_1}{\Gamma_2 \vdash \Delta_2} w
 \end{array}$$

(with $\Gamma_1 \subseteq \Gamma_2$ and $\Delta_1 \subseteq \Delta_2$)

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma \vdash A, \Delta}{\Gamma \vdash \Delta} c$$