c. Give a type derivation in $\lambda 2$ of the following type judgment, which gives the type of the type of the polymorphic identity:

$$\vdash \ \Pi A : *.\, A \to A \ : \ *$$

(*Below you will find the $\lambda 2$ type derivation rules.*)

(5 points)

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

Derivation rules of the Pure Type System $\lambda 2$

In these rules the variable s ranges over the set of sorts $\{*, \square\}$.

axiom

$$\frac{}{\vdash * : \square}$$

application

$$\frac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

abstraction

$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\, B : s}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B}$$

product

$$\frac{\Gamma \vdash A : s \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x : A.\, B : *}$$

weakening

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$$

variable

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$$

conversion

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \quad \text{where } B =_\beta B'$$

Give the variable condition for this rule.

(5 points)

c. Show that $(\forall x. \neg P(x)) \to \neg(\exists x. P(x))$ is a tautology of first-order intuitionistic predicate logic. (*Hint: use the existential quantification elimination rule as early as possible.*)

(5 points)

Exercise 5. This exercise is concerned with λ-calculus with dependent types.

a. What is the type of `natlist_dep`, the type of dependent lists? And what is the type of (`natlist_dep` 3)? Describe the elements of the type (`natlist_dep` 3).

(5 points)

b. The function `reverse` which reverses non-dependent lists has type

```
reverse : natlist -> natlist
```

What is the type of the analogous function `reverse_dep` on dependent lists?

(5 points)

c. The function `append_dep` appends two dependent lists. What are the types of the following two terms:

```
reverse_dep (plus n1 n2) (append_dep n1 n2 l1 l2)

append_dep n2 n1 (reverse_dep n2 l2) (reverse_dep n1 l1)
```

(In these terms `n1` and `n2` have type `nat`, while `l1` has type `(natlist_dep n1)` and `l2` has type `(natlist_dep n2)`.)

Are the two types of these two terms convertible?

(5 points)

Exercise 6. This exercise is concerned with second-order propositional logic and polymorphic λ-calculus.

a. Show that $(\forall c. ((a \to b \to c) \to c)) \to a$ is a tautology of second-order minimal propositional logic.

(5 points)

b. What is the impredicative definition of $\bot$ in second-order propositional logic?

(5 points)

Exercise 3. This exercise is concerned with inductive types and recursive function definitions in Coq.

a. Give the inductive definition of the datatype `natbintree` of binaries trees with unlabeled nodes and natural numbers at the leafs.

(5 points)

b. The Coq function that appends two lists of natural numbers is defined by

```
Fixpoint append (l k : natlist) {struct l} : natlist :=
  match l with
    nil => k
  | cons n l' => cons n (append l' k)
  end.
```

Explain what the '`{struct l}`' in this definition means.

(5 points)

c. Give the definition of a recursive function

`flatten : natbintree -> natlist`

which flattens the tree into a linear list. For example `flatten (node (node (leaf 3) (leaf 1)) (leaf 4))` should be `cons 3 (cons 1 (cons 4 nil))`.

In your definition you may use the function `append` from 3b.

(5 points)

Exercise 4. This exercise is concerned with first-order predicate logic.

a. Give the λ-term that under the Curry-Howard-de Bruijn isomorphism corresponds to the following proof in first-order predicate logic:

$$\dfrac{\dfrac{[\forall x.\, P(x)^u]}{P(a)}\,E\forall}{(\forall x.\, P(x)) \to P(a)}\,I[u]{\to}$$

In this term you can use constants $a : \texttt{Terms}$ and $P : \texttt{Terms} \to *$. (*NB: it is not asked to give the type derivation of this term.*)

(5 points)

b. The rule for elimination of an existential quantifier in first-order predicate logic is:

$$\dfrac{\begin{matrix}\vdots \\ \exists x.\, A\end{matrix} \qquad \begin{matrix}\vdots \\ \forall x.\, (A \to B)\end{matrix}}{B}\,E\exists$$

Exam Logical Verification

February 9, 2005

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

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

a. Show that the formula $(A \to A \to B) \to A \to B$ is a tautology.
   (5 points)

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

c. Replace in the following three terms the ?'s by simple types, such that we obtain typable λ-terms. (*NB: it is not asked to give the type derivations.*)
   $\lambda x : ?. \lambda y : ?. \, x$
   $\lambda x : ?. \lambda y : ?. \, x \, y \, y$
   $\lambda x : ?. \lambda y : ?. \lambda z : ?. \, x \, (y \, z)$
   (5 points)

Exercise 2. This exercise is concerned with detour elimination in first-order minimal propositional logic.

a. What is the definition of a detour in a natural deduction proof?
   (5 points)

b. Give a proof of $A \to A \to A$ in first-order minimal propositional logic that contains a detour.
   (5 points)

c. Give the λ-term that corresponds to the proof of 2b. Give the normal form of this λ-term. What subterm in the proof term corresponding to the proof of 2b is the β-redex that corresponds to the detour?
   (5 points)