**Universität Innsbruck**
**Institut für Informatik**
**Prof. Dr. Clemens Ballarin**

**SS 2008**
**Exercise Sheet 2**
**Due 17 April 2008**

# Automatic Deduction — Introduction to Isabelle

# LVA 703522

## 1 Type Inference

▷ Infer most general types for the following $\lambda$-terms. Give the full typing derivations.

(a) $\lambda f\, n.\, f\, n$

(b) $\lambda f\, n.\, f\,(f\, n)$

(c) $\lambda f\, n.\, f\,(f\,(f\, n))$

(d) $(\lambda x.\, x)\,(\lambda x.\, x)$

▷ Is the term $\lambda x.\, x\, x$ type correct? Justify your answer.

## 2 Natural Deduction

We will use the calculus of natural deduction to prove some lemmas of propositional and predicate logic in Isabelle.

### 2.1 Propositional Logic

- Only use these rules in the proofs:

```
notI: (P ⟹ False) ⟹ ¬ P
notE: ⟦¬ P; P⟧ ⟹ R
conjI: ⟦P; Q⟧ ⟹ P ∧ Q
conjE: ⟦P ∧ Q; ⟦P; Q⟧ ⟹ R⟧ ⟹ R
disjI1: P ⟹ P ∨ Q
disjI2: Q ⟹ P ∨ Q
disjE: ⟦P ∨ Q; P ⟹ R; Q ⟹ R⟧ ⟹ R
impI: (P ⟹ Q) ⟹ P ⟶ Q
impE: ⟦P ⟶ Q; P; Q ⟹ R⟧ ⟹ R
mp: ⟦P ⟶ Q; P⟧ ⟹ Q
iffI: ⟦P ⟹ Q; Q ⟹ P⟧ ⟹ P = Q
iffE: ⟦P = Q; ⟦P ⟶ Q; Q ⟶ P⟧ ⟹ R⟧ ⟹ R
```

- Only use the methods `(rule` $r$`)`, `(erule` $r$`)` and `assumption`, where $r$ is one of the rules given above.

▷ Prove the following lemmas in Isabelle.

**lemma** "((A ∨ B) ∨ C) ⟶ A ∨ (B ∨ C)"

**lemma** "(A ∨ A) = (A ∧ A)"

**lemma** "(D ⟶ A) ⟶ (A ⟶ (B ∧ C)) ⟶ (B ⟶ ¬ C) ⟶ ¬ D"

**lemma** "(A ⟶ ¬ B) = (B ⟶ ¬ A)"

## 2.2 Pierce's law

Prove Pierce's law `((A ⟶ B) ⟶ A) ⟶ A`.

▷ First give a paper proof using case distinction and/or proof by contradiction.

▷ Now give a proof in Isabelle. In addition to the rules and methods from Exercise 2.1, you may use `(case_tac` $P$`)` (where $P$ is a Boolean expression, e.g. a variable) for case distinctions, `back` to select a different unifier when applying a method, and the theorem `classical: (¬ P ⟹ P) ⟹ P`.

## 2.3 Predicate Logic

We are again talking about proofs in the calculus of Natural Deduction. In addition to the theorems given in the exercise "Propositional Logic" (Exercises 2), you may now also use

```
exI: P x ⟹ ∃x. P x
exE: ⟦∃x. P x; ⋀x. P x ⟹ Q⟧ ⟹ Q
allI: (⋀x. P x) ⟹ ∀x. P x
allE: ⟦∀x. P x; P x ⟹ R⟧ ⟹ R
```

▷ Give a proof of the following propositions or an argument why the formula is not valid:

**lemma** "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"

**lemma** "(∀x. P x ⟶ Q) = ((∃x. P x) ⟶ Q)"

**lemma** "((∃ x. P x) ∧ (∃ x. Q x)) = (∃ x. (P x ∧ Q x))"

**lemma** `"((∃ x. P x) ∨ (∃ x. Q x)) = (∃ x. (P x ∨ Q x))"`

The following lemma also requires `classical: (¬ P ⟹ P) ⟹ P` (or an equivalent theorem) in order to be proved.

**lemma** `"(¬ (∀ x. P x)) = (∃ x. ¬ P x)"`

### 2.4 A Riddle: Rich Grandfather

▷ First prove the following formula, which is valid in classical predicate logic, informally with pen and paper. Use case distinctions and/or proof by contradiction.

*If every poor man has a rich father,*
*then there is a rich man who has a rich grandfather.*

**theorem**
```
"∀x. ¬ rich x ⟶ rich (father x) ⟹
 ∃x. rich (father (father x)) ∧ rich x"
```

▷ Now prove the formula in Isabelle using a sequence of rule applications (i.e. only using the methods `rule`, `erule` and `assumption`). In addition to the theorems that were allowed in the exercise "Predicate Logic", you may now also use `classical: (¬ P ⟹ P) ⟹ P`.