Universität Innsbruck — **SS 2008**
Institut für Informatik — **Exercise Sheet 3**
Prof. Dr. Clemens Ballarin — **Due 8 May 2008**

Automatic Deduction — Introduction to Isabelle

LVA 703522

1 Isar

1.1 Propositional Logic

▷ Provide Isar proof texts that prove the following lemmas. You may only use the methods `rule` and `-`.

```
lemma "(A ∨ A) = (A ∧ A)"
proof
  assume "A ∨ A"
  then have a: "A" by rule
  from a a show "A ∧ A" by rule
next
  assume "A ∧ A"
  then have "A" by rule
  then show "A ∨ A" by rule
qed
```

Variation, using backticks.

```
lemma "(A ∨ A) = (A ∧ A)"
proof
  assume "A ∨ A"
  then have "A" by rule
  from `A` `A` show "A ∧ A" by rule
next
  assume "A ∧ A"
  then have "A" by rule
  then show "A ∨ A" by rule
qed
```

```
lemma "(A ∨ B) ∨ C ⟶ A ∨ (B ∨ C)"
proof
  assume "(A ∨ B) ∨ C"
  then show "A ∨ (B ∨ C)"
  proof
    assume "A ∨ B"
    then show ?thesis
    proof
      assume "A"
```

```
      then show ?thesis by rule
    next
      assume "B"
      then have "B ∨ C" by rule
      then show ?thesis by rule
    qed
  next
    assume "C"
    then have "B ∨ C" by rule
    then show ?thesis by rule
  qed
qed
```

1.2 Predicate Logic

▷ Provide Isar proof texts that prove the following lemmas. Again, you may not use automation.

```
lemma "((∀ x. P x) ∧ (∀ x. Q x)) = (∀ x. (P x ∧ Q x))"
proof
  assume pq: "(∀x. P x) ∧ (∀x. Q x)"
  from pq have p: "∀x. P x" ..
  from pq have q: "∀x. Q x" ..
  show "∀x. P x ∧ Q x"
  proof
    fix x
    from p have p': "P x" ..
    from q have q': "Q x" ..
    from p' q' show "P x ∧ Q x" ..
  qed
next
  assume pq: "∀x. P x ∧ Q x"
  show "(∀x. P x) ∧ (∀x. Q x)"
  proof
    show "∀x. P x"
    proof
      fix x
      from pq have "P x ∧ Q x" ..
      then show "P x" ..
    qed
  next
    show "∀x. Q x"
    proof
      fix x
      from pq have "P x ∧ Q x" ..
      then show "Q x" ..
    qed
  qed
qed
```

```
lemma "((∃ x. P x) ∨ (∃ x. Q x)) = (∃ x. (P x ∨ Q x))"
proof
  assume "(∃x. P x) ∨ (∃x. Q x)"
  then show "∃x. P x ∨ Q x"
  proof
    assume "∃x. P x"
    then obtain y where "P y" ..
    then have "P y ∨ Q y" ..
    then show ?thesis ..
  next
    assume "∃x. Q x"
    then obtain y where "Q y" ..
    then have "P y ∨ Q y" ..
    then show ?thesis ..
  qed
next
  assume "∃x. P x ∨ Q x"
  then obtain y where "P y ∨ Q y" ..
  then show "(∃ x. P x) ∨ (∃ x. Q x)"
  proof
    assume "P y"
    then have "∃x. P x" ..
    then show ?thesis ..
  next
    assume "Q y"
    then have "∃x. Q x" ..
    then show ?thesis ..
  qed
qed
```

The following lemma also requires `classical: (¬ P ⟹ P) ⟹ P` (or an equivalent theorem) in order to be proved. You need to invoke this explicitly with **proof** `rule classical` or similar.

```
lemma "(¬ (∀ x. P x)) = (∃ x. ¬ P x)"
proof
  assume "¬ (∀x. P x)"
  show "∃x. ¬ P x"
  proof (rule classical)
    assume "¬ (∃x. ¬ P x)"
    have "∀x. P x" proof
      fix x
      show "P x"
      proof (rule classical)
        assume "¬ P x"
        then have "∃x. ¬ P x" ..
        with ‘¬ (∃x. ¬ P x)‘ show ?thesis ..
      qed
    qed
```

```
    with ‘¬ (∀x. P x)‘ show ?thesis ..
  qed
next
  assume "∃x. ¬ P x"
  then obtain x where "¬ P x" by rule
  show "¬ (∀x. P x)" proof
    assume "∀x. P x"
    then have "P x" by rule
    from ‘¬ P x‘ ‘P x‘ show "False" by rule
  qed
qed
```

Hint: it may be useful to study the natural deduction proofs of these lemmas before attempting to provide Isar proofs.