



# Theorembeweiserpraktikum – SS 2016

<http://pp.ipd.kit.edu/lehre/SS2016/tba>

## Blatt 2: Simplifikation

Abgabe: 2. Mai 2016, 12:00 Uhr  
Besprechung: 3. Mai 2016

## 1 Prädikatenlogik

Es geht wieder um Beweise mit den Regeln des Kalküls des natürlichen Schließens. Zusätzlich zu den Regeln der letzten Übung können Sie nun auch folgende Regeln verwenden:

$allI: (\bigwedge x. P x) \implies \forall x. P x$      $allE: \forall x. P x \implies (P x \implies R) \implies R$   
 $exI: P x \implies \exists x. P x$      $exE: \exists x. P x \implies (\bigwedge x. P x \implies Q) \implies Q$

Es dürfen wieder nur die Befehle **proof** (mit (*rule* ...)), **assume**, **have**, **show**, **next**, **qed** und **from** sowie die darauf aufbauenden Abkürzungen verwendet werden. Zusätzlich dürfen die Befehle **fix** und **obtain** verwendet werden.

### Beispiel

**lemma** " $\forall x. P x \longrightarrow (\exists x. P x)$ "

**proof**(*rule allI*)

**fix**  $x$

**show** " $P x \longrightarrow (\exists x. P x)$ "

**proof**

**assume** " $P x$ "

**then show** " $\exists x. P x$ " **by** (*rule exI*)

**qed**

**qed**

**lemma** " $(\forall x. P x) \longleftrightarrow \neg (\exists x. \neg P x)$ "

*<solution>*

**lemma** " $(\neg (\forall x. P x)) \longleftrightarrow (\exists x. \neg P x)$ "

*<solution>*

**lemma** " $(\forall x. P x \longrightarrow Q) \longleftrightarrow ((\exists x. P x) \longrightarrow Q)$ "

*<solution>*

**lemma** " $(\exists x. \forall y. P x y) \longrightarrow (\forall y. \exists x. P x y)$ "

*<solution>*

**lemma** " $(\forall x. P x) \wedge (\forall x. Q x) \longleftrightarrow (\forall x. (P x \wedge Q x))$ "

*<solution>*

**lemma** " $(\exists x. P x) \vee (\exists x. Q x) \longleftrightarrow (\exists x. (P x \vee Q x))$ "

*<solution>*

Bei diesem Lemma darf mit Fallunterscheidung (Methode *cases*) gearbeitet werden. Erinnerung: Eine Variable, über die Sie nichts wissen (brauchen), erhalten Sie mit **fix**.

**lemma** " $\exists x. P\ x \longrightarrow (\forall x. P\ x)$ "

*<solution>*

## 2 Definitionen und Arbeiten mit Gleichheit

In klassischer Aussagenlogik können alle Aussagen allein aus *False* und *nor* gebildet werden. Definieren Sie den *nor*-Operator:

**definition**

**nor** :: "*bool*  $\Rightarrow$  *bool*  $\Rightarrow$  *bool*" (**infix** " $\downarrow$ " 37)

**where** "*A*  $\downarrow$  *B*  $\longleftrightarrow$  ...."

Nun leiten Sie Lemmas her, die die üblichen booleschen Junktoren nur mit *False* und *op*  $\downarrow$  darstellen. Theoretisch ist *op*  $\downarrow$  alleine schon universell. Der Einfachheit halber dürfen Sie in dieser Aufgabe aber auch noch *False* zusätzlich verwenden. Gehen Sie dabei wie folgt vor:

- Wenden Sie zu Beginn keine Regel an (**proof-**).
- Schreiben Sie erst mit **have** .... = .... den Ausdruck in eine Form um, die neben Ausdrücken der Form  $\neg$  (....  $\vee$  ....) nur Junktoren verwendet, für die Sie bereits Regeln geschrieben haben.
- Führen Sie diese dann Schrittweise mit **also have** in die gewünschte Form über. Dabei sollten Sie neben *nor\_def[symmetric]* nur die davor bewiesenen Regeln *rewrite\_foo* in ihren **from**-Befehlen aufführen müssen, und als Beweis dann **.** oder **by** (*rule arg\_cong*) verwenden.
- Wenn die Gleichungskette zum Lemma passt, lässt sie sich mit **finally show** *?thesis* **.** abschließen.

Um *True* zu zeigen verwenden Sie die Regel *TrueI: True*.

**lemma** *rewrite\_not*: " $\neg A \longleftrightarrow$  ...."

*<solution>*

**lemma** *rewrite\_or*: "*A*  $\vee$  *B*  $\longleftrightarrow$  ...."

*<solution>*

**lemma** *rewrite\_and*: "*A*  $\wedge$  *B*  $\longleftrightarrow$  ...."

*<solution>*

**lemma** *rewrite\_imp*: "*(A*  $\longrightarrow$  *B)*  $\longleftrightarrow$  ...."

*<solution>*

**lemma** *rewrite\_True*: "*True*  $\longleftrightarrow$  ...."

*<solution>*

Schreiben Sie nun den folgenden Ausdruck schrittweise so um, das er nur mit *op*  $\downarrow$  und *False* gebildet wird. Verwenden Sie dabei das **also ... finally** Konstrukt.

**lemma** "*(A*  $\longrightarrow$  (*A*  $\vee$  *B*))  $\wedge$   $\neg B \longleftrightarrow$  ...."

*<solution>*