Universität Karlsruhe (TH)
Lehrstuhl für Programmierparadigmen

Theorembeweiser und ihre Anwendungen SS 2009 http://pp.info.uni-karlsruhe.de/
Übungsleiter: Daniel Wasserrab `wasserra@ipd.info.uni-karlsruhe.de`

Übungsblatt 2 Besprechung: 05.05.2009

Prädikatenlogik

Es geht wiederum um Beweise mit den Regeln des Kalküls des natürlichen Schließens. Zusätzlich zu den Regeln von Blatt 1 können Sie nun auch folgende Regeln verwenden:

allI: $(\bigwedge x.\ P\ x) \Longrightarrow \forall x.\ P\ x$
allE: $[\![\forall x.\ P\ x;\ P\ x \Longrightarrow R]\!] \Longrightarrow R$
exI: $P\ x \Longrightarrow \exists x.\ P\ x$
exE: $[\![\exists x.\ P\ x;\ \bigwedge x.\ P\ x \Longrightarrow Q]\!] \Longrightarrow Q$

Es dürfen wiederum nur die Methoden *rule*, *erule* und *assumption* verwendet werden.

Achten Sie darauf, für *exI* und *allE* nur *rule-tac* bzw. *erule-tac* mit gleichzeitiger Instantiierung der Variablen zu verwenden, ansonsten erhalten sie beliebige Variablen, die den Beweis erschweren können!

Beispiel:

lemma $\forall x.\ P\ x \longrightarrow (\exists x.\ P\ x)$
apply(*rule allI*)
apply(*rule impI*)
apply(*rule-tac x=x* **in** *exI*)
apply *assumption*
done

lemma $(\forall x.\ P\ x) = (\neg\ (\exists x.\ \neg\ P\ x))$
oops

lemma $(\neg\ (\forall x.\ P\ x)) = (\exists x.\ \neg\ P\ x)$
oops

lemma $(\forall x.\ P\ x \longrightarrow Q) = ((\exists x.\ P\ x) \longrightarrow Q)$
oops

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

lemma $((\forall x.\ P\ x) \wedge (\forall x.\ Q\ x)) = (\forall x.\ (P\ x \wedge Q\ x))$
oops

lemma $((\exists x.\ P\ x) \vee (\exists x.\ Q\ x)) = (\exists x.\ (P\ x \vee Q\ x))$
oops

lemma $\exists x.\ P\ x \longrightarrow (\forall x.\ P\ x)$
oops