

Induktion

Vollständige Induktion über den natürlichen Zahlen

B eine Aussage in der die Variable n frei vorkommt.

(i) Induktionsanfang

Nachweis von $B(0)$

(ii) Induktionsschritt

Nachweis von $\forall n (B(n) \rightarrow B(n + 1))$

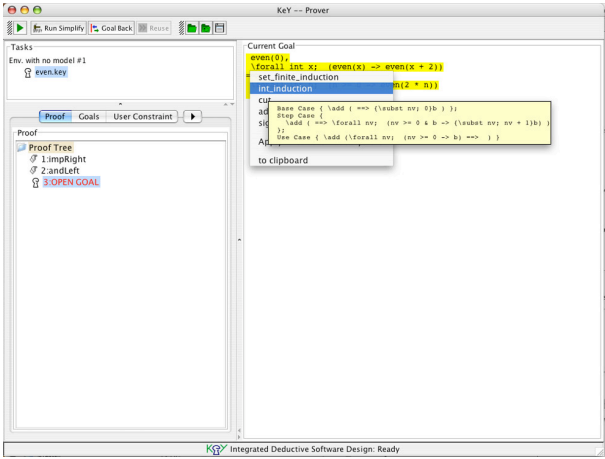
n

Induktionsvariable

1

Induktion in KeY

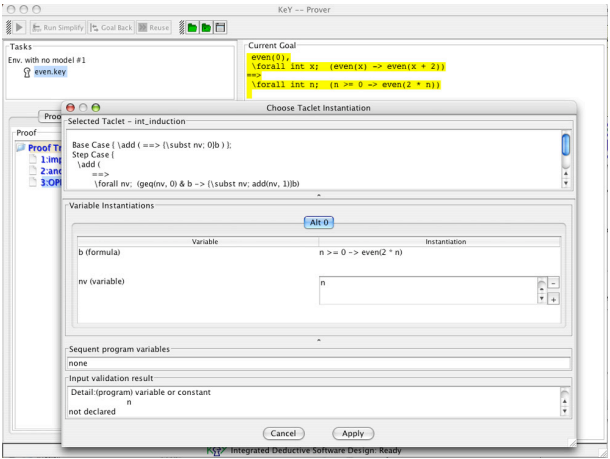
Auswahl der Regel nach Anklicken von $\Rightarrow$



2

Induktion in KeY

anschließend Angabe
der Aussage B und



der Induktionsvariable

3

Induktion in KeY

Anwendung der Regel bei Vorliegen des Ziels

$$\Phi \Rightarrow \Psi$$

Neue Beweisziele

1. Der Induktionsanfang: $\Phi \Rightarrow B, \Psi$
2. Der Induktionsschritt: $\Phi \Rightarrow \forall n (B(n) \rightarrow B(n + 1)), \Psi$
3. Die Anwendung: $\Phi, \forall n B \Rightarrow \Psi$

4

Strukturelle Induktion

hier exemplarisch für Folgen von Variablen

B Aussage, in der die Variable vs für Folgen frei vorkommt

Konstruktor $vcons$
leere Folge v_ϵ

(i) Induktionsanfang

Nachweis von $B(v_\epsilon)$

(ii) Induktionsschritt

Nachweis von $\forall vs (B(vs) \rightarrow \forall x B(vcons(x, vs)))$

5

Die Schnittregel

Eigentlich die saubere Verwendung von Hilfssätzen?

$$\frac{\Phi_1 \Rightarrow X, \Psi_1 \quad \Phi_2, X \Rightarrow \Psi_2}{\Phi_1, \Phi_2 \Rightarrow \Psi_1, \Psi_2}$$

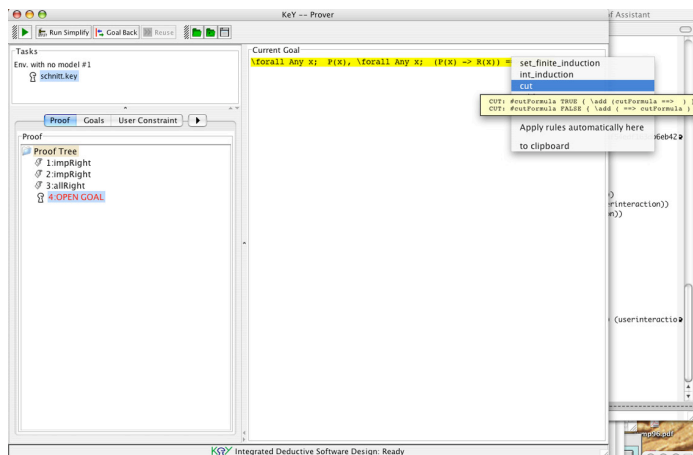
X

Schnittformel

6

Anwendung der Schnittregel in KeY

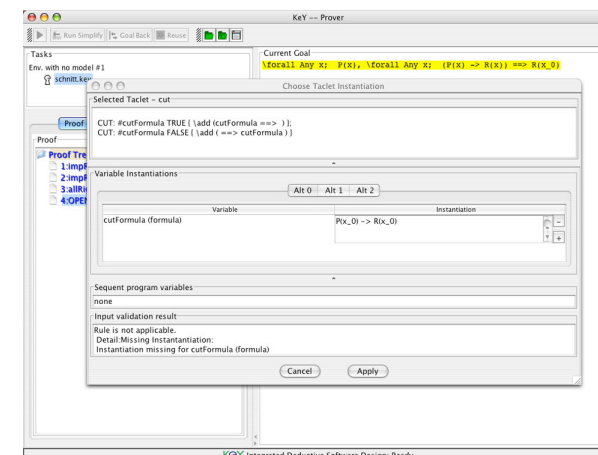
Anklicken von $\Rightarrow$ und Auswahl von cut



7

Anwendung der Schnittregel in KeY

Im nächsten Dialogfenster wird die Schnittformel angegeben.



8

Anwendung der Schnittregel in KeY

$$\frac{\Phi_1 \Rightarrow X, \Psi_1 \quad \Phi_2, X \Rightarrow \Psi_2}{\Phi_1, \Phi_2 \Rightarrow \Psi_1, \Psi_2}$$

Beweisziele nach Anwendung der Schnittregel

1. CUT: X TRUE

Verwendung der Schnittformel, entspricht der zweiten Prämisse

2. CUT: X FALSE

Beweis der Schnittformel, entspricht der ersten Prämisse.

(eine Beweisverpflichtung)

Hilfssatz 3. Wf-Exp-Nicht-true

$$\forall x: Var \ \forall vs: Vars \ (\neg vs = v_\varepsilon \wedge \neg In(x, vs) \rightarrow Wf-Exp(f(x, vs, t_\varepsilon)))$$

Unter Verwendung von: Wf-Exp-Vs

$$\forall x: Var \ (\forall vs: Vars \ (\forall tli: Tlist \ (vs \neq v_\varepsilon \wedge \neg In(x, vs) \wedge Is-Asg-List(x, tli) \rightarrow Wf-Exp(f(x, vs, tli)))))$$

Nachweis von Wf-Exp-Vs

$$\forall x: Var \ (\forall vs: Vars \ (\forall tli: Tlist \ (vs \neq v_\varepsilon \wedge \neg In(x, vs) \wedge Is-Asg-List(x, tli) \rightarrow Wf-Exp(f(x, vs, tli)))))$$

durch Induktion über die Länge von vs .

Die Aussage wird verallgemeinert:

$$\forall x: Var \ (\forall vs: Vars \ (\forall tli: Tlist \ (lg(vs) = n \wedge n \geq 0 \rightarrow (\neg In(x, vs) \wedge Is-Asg-List(x, tli) \rightarrow Wf-Exp(f(x, vs, tli)))))$$

Taclets für lg

<i>Lg_Leer</i>	$lg(vs) = 0 \leftrightarrow vs = v_\varepsilon$
<i>Lg_Leer2</i>	$vs = v_\varepsilon \leftrightarrow lg(vs) = 0$
<i>Lg_NichtLeer</i>	$lg(vs) > 0 \leftrightarrow \exists x, vs_1 \ vs = vcons(x, vs_1)$
<i>Lg_Cons</i>	$lg(vcons(x, vs)) \leftrightarrow lg(vs) + 1$
<i>Lg_ge0</i>	$lg(vs) \geq 0 \leftrightarrow true;$

Ersetzt wird die linke Seite durch die rechte

In *Lg_NichtLeer* dürfen x und vs_1 nicht frei in vs vorkommen.

Induktionsschritt

Hier strukturelle Induktion

$\forall vs: Vars$

$$\begin{aligned} & (\forall x: Var (\forall tli: Tlist (\neg In(x, vs) \wedge Is-Asg-List(x, tli) \rightarrow Wf-Exp(f(x, vs, tli)))))) \\ & \rightarrow \forall x: Var (\forall y: Var (\forall tli: Tlist \\ & \quad (\neg In(x, vcons(y, vs)) \wedge Is-Asg-List(x, tli) \rightarrow Wf-Exp(f(x, vcons(y, vs), tli)))))) \end{aligned}$$

unter Verwendung des Hilfssatz *Is-Asg-List-Conc1*

Hilfssatz *Is-Asg-List-Conc1*

$$\forall x: Var (\forall y: Var (\forall vs: Vars (\forall tli: Tlist (\neg In(x, vcons(y, vs)) \wedge Is-Asg-List(x, tli) \rightarrow Is-Asg-List(x, conc_1(tli, mk-rasg(x, x, y)))))))$$

Für die Aussage

$$\begin{aligned} & \forall x: Var (\forall tli: Tlist (\forall r: RAsg \\ & \quad Is-Asg-List(x, tli) \wedge Is-R-Asg(r) \wedge x = s - tar(r) \\ & \quad \rightarrow Is-Asg-List(x, conc_1(tli, r))))), \end{aligned}$$

die festlegt, wann *conc₁* eine korrekte Asg-Liste erzeugt, ist das Taclet *Conc1_AsgListDef* vorhanden.