

Teil 4: Logik

- Regeln der Aussagenlogik
- Regelanwendung (vorwärts / rückwärts)
- Sichere / unsichere Regeln

Im Tutorial: Kap. 5

HOL - die Logik

Klassische Prädikatenlogik mit folgenden Besonderheiten:

- *Höherer Ordnung (HOL):*
Quantifizierung über Funktionen/Prädikate: $\exists f. \forall x. f\ x = x$
- *Formeln vom Typ bool:*
Gleichheit (=) und Äquivalenz (\leftrightarrow) werden gleichgesetzt
- *Auswahloperator:*
 $\varepsilon x. P\ x$ (ASCII: *SOME* $x.P\ x$) bedeutet: Ein x , das P erfüllt.

Regeln des Natürlichen Schließens

Notation:

Statt $R: \llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$ schreibe: $\frac{A_1 \dots A_n}{A} R$

Erlaubt nachvollziehbare Darstellung von **Ableitungen**,
z.B. von $\llbracket A \wedge B; C \rrbracket \Longrightarrow A \wedge (C \vee D)$

$$\frac{\frac{A \wedge B}{A} \text{conjunct1} \quad \frac{C}{C \vee D} \text{disjI1}}{A \wedge (C \vee D)} \text{conjI}$$

Regeln der Aussagenlogik

Einführungsregeln (*Intro*):

$$\frac{A \quad B}{A \wedge B} \text{ conjI}$$

$$\frac{A}{A \vee B} \text{ disjI1} \quad \frac{B}{A \vee B} \text{ disjI2}$$

$$\frac{A \implies B}{A \longrightarrow B} \text{ impI}$$

$$\frac{A \implies B \quad B \implies A}{A = B} \text{ iffI}$$

$$\frac{A \implies \text{False}}{\neg A} \text{ notI}$$

$$\frac{}{\text{True}} \text{ TrueI}$$

$$\frac{}{t = t} \text{ refl}$$

Beseitigungsregeln (*Elim*):

$$\frac{A \wedge B \quad [[A; B]] \implies C}{C} \text{ conjE}$$

$$\frac{A \vee B \quad A \implies C \quad B \implies C}{C} \text{ disjE}$$

$$\frac{A \longrightarrow B \quad A \quad B \implies C}{C} \text{ impE}$$

$$\frac{A = B}{A \implies B} \text{ iffD1} \quad \frac{A = B}{B \implies A} \text{ iffD2}$$

$$\frac{A \quad \neg A}{C} \text{ notE}$$

$$\frac{\text{False}}{C} \text{ FalseE}$$

$$\frac{s = t \quad P(s)}{P(t)} \text{ subst}$$

Aussagenlogik: Weitere Regeln (1)

Strukturell: $\frac{A_1 \dots A \dots A_n}{A}$ assumption

Weitere Regeln (z.T. abgeleitet):

$$\frac{A \wedge B}{A} \text{ conjunct1} \quad \frac{A \wedge B}{B} \text{ conjunct2}$$

$$\frac{A \longrightarrow B \quad A}{B} \text{ mp}$$

$$\frac{\neg A \Longrightarrow A}{A} \text{ classical} \quad \frac{\neg A \Longrightarrow \text{False}}{A} \text{ ccontr}$$

$$\frac{s = t}{t = s} \text{ sym} \quad \frac{r = s \quad s = t}{r = t} \text{ trans}$$

Aussagenlogik: Weitere Regeln (2)

Beachte: Prämissen der Form $\llbracket \dots \rrbracket \implies C$ entsprechen Teilbeweisen

Ableitung von conjunct1 aus conjE und assumption:

$$\frac{A \wedge B \quad \llbracket A; B \rrbracket \implies A}{A} \text{ conjE}$$

mit Teilbeweis $\llbracket A; B \rrbracket \implies A$:

$$\frac{A \quad B}{A} \text{ assumption}$$

- Regeln der Aussagenlogik
- **Regelanwendung (vorwärts / rückwärts)**
- Sichere / unsichere Regeln

Regelanwendung in Isabelle: Überblick

Regelanwendung mit:

- `rule`: Vorwärtsschließen
- `frule`: Rückwärtsschließen – Prämisse bleibt erhalten
- `drule`: Rückwärtsschließen – Prämisse wird gelöscht
- `erule`: Kombination von `rule` und `drule`

Regelanwendung in Isabelle: rule (1)

Vorwärtsschließen: apply (rule R)

Gegeben: Regel R : $\llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$

Aktuelles Beweisziel: $\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B$

Bedingung: $A \equiv B$ (vorerst: “syntaktisch gleich”)

Neue Beweisziele:

$\llbracket B_1 \dots B_m \rrbracket \Longrightarrow A_1$

...

$\llbracket B_1 \dots B_m \rrbracket \Longrightarrow A_n$

warum korrekt?

Regelanwendung in Isabelle: rule (2)

Regel R : $\llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$
 mit $A \equiv B$

Beweisziele:

$$\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B \quad \rightsquigarrow \quad \begin{array}{l} \llbracket B_1 \dots B_m \rrbracket \Longrightarrow A_1 \\ \dots \\ \llbracket B_1 \dots B_m \rrbracket \Longrightarrow A_n \end{array}$$

Beweisbaum:

$$\frac{B_1 \dots B_m}{B} \quad \rightsquigarrow \quad \frac{\frac{B_1 \dots B_m}{A_1} \quad \dots \quad \frac{B_1 \dots B_m}{A_n}}{B} R$$

Regelanwendung in Isabelle: frule / drule (1)

Rückwärtsschließen: apply (frule R) bzw. apply (drule R)

Gegeben: Regel R : $\llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$

Aktuelles Beweisziel: $\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B$

Bedingung: $A_1 \equiv B_i$

Neue Beweisziele:

$\llbracket B_1 \dots B_i^? \dots B_m \rrbracket \Longrightarrow A_2$

...

$\llbracket B_1 \dots B_i^? \dots B_m \rrbracket \Longrightarrow A_n$

$\llbracket B_1 \dots B_i^? \dots B_m, A \rrbracket \Longrightarrow B$

hierbei:

- B_i bleibt erhalten bei **frule**
- B_i wird gelöscht bei **drule**

Regelanwendung in Isabelle: frule / drule (2)

Regel R : $\llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$
 mit $A_1 \equiv B_i$

Beweisziele:

$$\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B \quad \rightsquigarrow \quad \begin{array}{l} \llbracket B_1 \dots B_i^? \dots B_m \rrbracket \Longrightarrow A_2 \\ \dots \\ \llbracket B_1 \dots B_i^? \dots B_m \rrbracket \Longrightarrow A_n \\ \llbracket B_1 \dots B_i^? \dots B_m, A \rrbracket \Longrightarrow B \end{array}$$

Beweisbaum: (mit $\mathcal{B} = B_1 \dots B_i^? \dots B_m$)

$$\frac{B_1 \dots B_m}{B} \quad \rightsquigarrow \quad \frac{\frac{B_1 \dots B_i \dots B_m}{A_1} \text{ ass.} \quad \frac{\mathcal{B}}{A_2} \quad \dots \quad \frac{\mathcal{B}}{A_n}}{A} R}{B}$$

Regelanwendung in Isabelle: erule

Kombiniertes Vorwärts- / Rückwärtsschließen: apply (erule R)

Gegeben: Regel R : $\llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$

Aktuelles Beweisziel: $\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B$

Bedingung: $A_1 \equiv B_i$ und $A \equiv B$

Neue Beweisziele:

$\llbracket B_1 \dots B_{i-1}, B_{i+1} \dots B_m \rrbracket \Longrightarrow A_2$

...

$\llbracket B_1 \dots B_{i-1}, B_{i+1} \dots B_m \rrbracket \Longrightarrow A_n$

hierbei: B_i wird gelöscht wie bei drule

Verwendung der Regeln (1)

Intro-Regeln eignen sich zur Zerlegung rechts von \implies .

Anwendung mit `rule`:

lemma " $P \implies A \wedge B$ "

apply (`rule conjI`)

Ergibt:

1. $P \implies A$

2. $P \implies B$

Elim-Regeln eignen sich zur Zerlegung links von \implies .

Anwendung mit `erule`:

lemma " $A \wedge B \implies C$ "

apply (`erule conjE`)

Ergibt:

1. $\llbracket A; B \rrbracket \implies C$

Verwendung der Regeln (2)

Dest-Regeln beschreiben (evtl. schwächere) Konsequenzen.
Anwendung mit `frule/drule`:

constdefs

```
prime :: "nat  $\Rightarrow$  bool"
```

```
"prime p == 1 < p  $\wedge$  ( $\forall$  m. m dvd p  $\longrightarrow$  m = 1  $\vee$  m = p)"
```

```
even :: "nat  $\Rightarrow$  bool"
```

```
"even n ==  $\exists$  k. n = 2 * k"
```

```
odd :: "nat  $\Rightarrow$  bool"
```

```
"odd n ==  $\neg$  even n"
```

```
lemma even_prime_2: "[[ prime p; even p ]]  $\Longrightarrow$  p = 2"
```

```
lemma "[[ prime p; 2 < p ]]  $\Longrightarrow$  odd p"
```

```
apply (simp add: odd_def)
```

```
apply (rule notI)
```

```
apply (drule even_prime_2)
```

```
  apply assumption
```

```
apply simp
```

```
done
```

Regelanwendung und Unifikation

Anwendbarkeit: Statt syntaktischer Gleichheit genügt *Unifizierbarkeit*

Speziell für (rule R):

Gegeben: Regel $R: \llbracket A_1 \dots A_n \rrbracket \Longrightarrow A$

Aktuelles Beweisziel: $\llbracket B_1 \dots B_m \rrbracket \Longrightarrow B$

Bedingung: A und B unifizierbar mit Unifikator σ , d.h. $\sigma(A) \equiv \sigma(B)$

Neue Beweisziele:

$\llbracket \sigma(B_1) \dots \sigma(B_m) \rrbracket \Longrightarrow \sigma(A_1)$

...

$\llbracket \sigma(B_1) \dots \sigma(B_m) \rrbracket \Longrightarrow \sigma(A_n)$

Analog für (frule R), (drule R), (erule R)

- Regeln der Aussagenlogik
- Regelanwendung (vorwärts / rückwärts)
- **Sichere / unsichere Regeln**

Sichere / unsichere Regeln (1)

Sichere Regel: Ohne Informationsverlust: Beweisbarkeit bleibt erhalten

Bsp.:

lemma " $A \wedge B \implies P$ "

apply (*erule conjE*)

Ergibt:

1. $\llbracket A; B \rrbracket \implies P$

Daher: wende sichere Regeln zuerst an

Sichere aussagenlog. Regeln:

conjI, impI, notI, iffI, TrueI, refl, classical, conjE, disjE, FalsE

Sichere / unsichere Regeln (2)

Unsichere Regel: Mit Informationsverlust: Beweisbarkeit kann verloren gehen

Bsp.:

lemma " $\llbracket A; B; C \rrbracket \implies (A \wedge B) \vee (C \wedge D)$ "

apply (*rule disjI2*)

Ergibt:

$$1. \llbracket A; B; C \rrbracket \implies C \wedge D$$

Daher: wende unsichere Regeln möglichst spät an

Unsichere aussagenlog. Regeln:

disjI1, disjI2, impE, iffD1, iffD2, notE, FalseE, subst

Sichere / unsichere Regeln (3)

Auch bei sicheren Regeln:

Aufspaltung des Beweisbaums vermeiden (iffE, disjE, conjI)

Vergleiche:

Anwendung von conjE vor disjE, conjI

Anwendung von disjE, conjI vor conjE

bei

lemma " $\llbracket A \wedge B; C \vee D \rrbracket \implies A \wedge (D \vee C)$ "