Theorem Proving in Isabelle

Lutz Schröder

10. Dezember 2004

Isabelles Metalogik

- Typen: Basistypen, Typvariablen, Funktionstypen $\sigma \Rightarrow \tau$
- Basistypen ggf. parametrisiert, z.B. $'a\ list$
- Terme: Konstanten, Variablen, Abstraktionen $\lambda x.\, t$, Applikationen $t\, s$
- Typregeln:

$$\frac{[x :: \sigma] \quad t :: \tau}{\lambda x.\, t :: \sigma \Rightarrow \tau} \qquad \frac{s :: \sigma \quad t :: \sigma \Rightarrow \tau}{t\, s :: \tau}$$

Isabelles Metalogik (Forts.)

Eingebaut: Typ $prop$, Konstanten

$$\Longrightarrow \;::\; prop \Rightarrow prop \Rightarrow prop \qquad \text{(Meta-Implikation)}$$

$$\bigwedge \;::\; ('a \Rightarrow prop) \Rightarrow prop \qquad \text{(Meta-Allquantor)}$$

$$\equiv \;::\; 'a \Rightarrow 'a \Rightarrow prop \qquad \text{(Meta-Gleichheit)}$$

Meta-Regeln

Typischerweise eine Einführungsregel (I) und eine Eleminationsregel (E) pro logischer Operation, z.B.

$$(\Longrightarrow I)\ \frac{\begin{array}{c}[\phi]\\ \vdots \\ \psi\end{array}}{\phi \Longrightarrow \psi} \qquad (\Longrightarrow E)\ \frac{\phi \quad \phi \Longrightarrow \psi}{\psi}$$

Gleichheit von Funktionen

$$(\alpha)\ (\lambda x.\, t) = \lambda y.\, t[y/x]$$

$$(\beta)\ (\lambda x.t)\, s = t[s/x]$$

$$(ext)\ \frac{f\, x = g\, x}{f = g}\ (x \text{ nicht frei in Annahmen von } f\, x = g\, x)$$

Meta-Regeln in ML

- Theoreme sind ML-Datentyp thm
- Verzeichnet werden neben der Aussage selbst u.a. auch
  - ihr Beweis
  - ihre lokalen Annahmen
- Meta-Regeln sind Funktionen nach thm, z.B.

```
implies_elim:: thm => thm => thm
```

- thm ist verkapselt: Werte können nur mittels der Regeln erzeugt werden und sind somit immer korrekt.