Formale Methoden der Softwaretechnik
Vorlesung vom 16.06.11: Isabelle: Automatische Beweisprozeduren

Till Mossakowski & Christoph Lüth

Universität Bremen

Sommersemester 2011

Fahrplan

- Aussagenlogik
- Prädikatenlogik
- Isabelle I: Grundlagen
  - Aussagenlogik und natürlisches Schließen
  - Prädikatenlogik und Quantoren
  - Logik höherer Stufe
  - Isabelle: Definitionen und konservative Erweiterung
  - Isabelle: Automatische Beweisprozeduren
- Isabelle II: Anwendungen

Überblick

- Grundlagen der Termersetzung
- Automatische Beweisprozeduren im Überblick

Grundlagen der Termersetzung

- Gegeben: Menge von Gleichungen $\mathcal{E} = \{s_i = t_i\}_{i=1,\ldots,n}$

- Problem: Wann folgt $u = v$ aus $\mathcal{E}$ durch Gleichungsumformung?

Grundlagen der Termersetzung

- Gleichungen $s = t$ zu Regeln $s \to t$ orientieren
- Ableitungsrelation $s \Rightarrow_{\mathcal{R}} t$ definieren
- Äquivalenzabschluss bilden $s_1 \Leftarrow_{\mathcal{R}} s_2 \Rightarrow_{\mathcal{R}} s_3 \Rightarrow_{\mathcal{R}} s_4 \ldots s_n$
- Frage: Wann ist $s \Leftrightarrow_{\mathcal{R}} t$ dasselbe wie $s = t$?

Termersetzung

- Gegeben: Menge von Regeln $\mathcal{R} = \{s_i \to t_i\}_{i=1,\ldots,n}$
  Signatur Σ, Variablen X
- Ein Kontext ist $C \in T_\Sigma(X \cup \{\Box\})$, der $\Box$ genau einmal enthält.
- Für Kontext C, Term $t \in T_\Sigma(X)$ ist $C[t] = \sigma(C)$ mit $\sigma(\Box) = t$ und $\sigma(x) = x$ (für $x \neq \Box$).
- Ein-Schritt-Ersetzungsrelation:

$$\begin{aligned} s \to_{\mathcal{R}} t \iff & \exists\, l \to r \in \mathcal{R}, \\ & \exists \text{ Kontext } C, \sigma : X \to T_\Sigma(X) \\ & s = C[\sigma(l)], t = C[\sigma(r)] \end{aligned}$$

Relationen und Abschlüsse

- Für Relationen R, S ist Komposition
  $R \circ S = \{(a, c) \mid \exists b.\, (a, b) \in R, (b, c) \in S\}$
- Für Relation R ist die inverse Relation $R^{-1} = \{(b, a) \mid (a, b) \in R\}$
- R transitiv wenn $R \circ R \subseteq R$
- R reflexiv wenn $\forall x.\, (x, x) \in R$
- R symmetrisch wenn $R^{-1} \subseteq R$
- Transitiv-reflexiver Abschluss:

$$R^* = \bigcap_S R \subseteq S, S \text{ reflexiv und transitiv}$$

- Äquivalenzabschluss:

$$R^= = \bigcap_S R \subseteq S, S \text{ reflexiv, transitiv und symmetrisch}$$

- Notation: $\Rightarrow_{\mathcal{R}} = \rightarrow_{\mathcal{R}}^*$, $\Leftrightarrow_{\mathcal{R}} = \rightarrow_{\mathcal{R}}^=$

Eigenschaften von $\Rightarrow_{\mathcal{R}}$

- t ist in Normalform, wenn $t \Rightarrow_{\mathcal{R}} s \Longrightarrow s = t$
- s ist Normalform von t ($NF(t)$), wenn $t \Rightarrow_{\mathcal{R}} s$ und s in Normalform
- s und t sind reduzierbar ($s \downarrow t$), wenn $\exists u.\, s \Rightarrow_{\mathcal{R}} u, t \Rightarrow_{\mathcal{R}} u$
- Eigenschaften von $\Rightarrow_{\mathcal{R}}$:
  - Church-Rosser: $s \Leftrightarrow_{\mathcal{R}} t$ dann $s \downarrow t$
  - Konfluenz: $s_1 \Leftarrow_{\mathcal{R}} t \Rightarrow_{\mathcal{R}} s_2$ dann $s_1 \downarrow s_2$
  - Termination: Keine unendliche Kette $s_1 \rightarrow_{\mathcal{R}} s_2 \rightarrow_{\mathcal{R}} s_3 \ldots s_n$ ($\Rightarrow_{\mathcal{R}}$ wohlfundiert)

Sätze

- Lemma: $\Rightarrow_{\mathcal{R}}$ Church-Rosser gdw. $\Rightarrow_{\mathcal{R}}$ konfluent
- Lemma: Wenn $\Rightarrow_{\mathcal{R}}$ terminierend und konfluent, dann
  1. Normalform eindeutig
  2. $s \Leftrightarrow_{\mathcal{R}} t$ gdw. $NF(s) = NF(t)$
- Satz: $s \Leftrightarrow_{\mathcal{R}} t$ gdw. $s =_{\mathcal{E}} t$

Kriterien für Konfluenz und Termination

- $\Rightarrow_{\mathcal{R}}$ konfluent, wenn alle kritischen Paare reduzierbar
- $\Rightarrow_{\mathcal{R}}$ terminierend, wenn Terminationsordnung existiert

Simplifikation

- Simplifikation ist Termersetzung:
  - Gegeben Theorem $s = t$, ersetze s durch t.
- Benutzung: `apply (simp)`
- Nutzt Gleichungen und Ungleichungen:
  - Funktionsdefinitionen
  - Vereinfachungsregeln für Datentypen
  - Deklarierte Theoreme
  - Annahmen des lokalen Subgoals
- Benutzt bedingte Gleichungen: $s_1 = t_1, \ldots, s_n = t_n \Longrightarrow s = t$
  - Ersetzt s durch t, wenn Gleichungen $s_1 = t_1 \ldots s_n = t_n$ rekursiv gezeigt werden können.
- Instantiiert keine Meta-Variablen
- Erzeugt keine neuen Subgoals, nur Vereinfachung

Klassische Beweiser

- Beweisplaner: `blast`
  - Konstruiert Beweis durch Suche
  - Gelingt oder schlägt fehl: keine neuen Subgoals
- Klassischer Beweiser: `clarify`
  - Wendet Einführungs- und Eliminationsregeln systematisch an
  - Keine neuen Subgoals, nur Vereinfachung
  - Sicher: keine unbeweisbaren Subgoals
  - `clarsimp`: Kombination mit Simplifikation
- Vollautomatisch: `auto`
  - Kombination verschiedener Beweiser
  - Instantiiert `Meta-Variablen`, erzeugt neue Subgoals
  - Unsicher: kann unbeweisbare Subgoals erzeugen

Zusammenfassung

- Automatische Beweisprozeduren in Isabelle:
  `simp`, `blast`, `clarify`, `auto`
- Wann welche benutzen?
  - Gleichungsumformung, Reduktion: `simp`
  - Regeln der Logik (Quantoren und Junktoren): `clarify`, `blast`
  - "Triviale" Schritte: `auto` (nur als letzte Methode!)