Vollständigkeit der Beweissysteme

http://www.informatik.uni-bremen.de/theorie/teach/veri

Renate Klempien-Hinrichs

- Vollständigkeit von PD und TD
- Ausdruckskraft
- Zusätzliche Beweisregeln

Vollständigkeit

- Ein Beweissystem K für eine Klasse C von Programmen ist vollständig für die partielle Korrektheit von Programmen aus C, wenn für alle Korrektheitsformeln $\{p\}\ S\ \{q\}$ mit $S \in C$ gilt:

$$\models \{p\}\ S\ \{q\} \text{ impliziert } \vdash_K \{p\}\ S\ \{q\}\,.$$

- Ein Beweissystem K für eine Klasse C von Programmen ist vollständig für die totale Korrektheit von Programmen aus C, wenn für alle Korrektheitsformeln $\{p\}\ S\ \{q\}$ mit $S \in C$ gilt:

$$\models_{tot} \{p\}\ S\ \{q\} \text{ impliziert } \vdash_K \{p\}\ S\ \{q\}\,.$$

Vollständigkeit von PD

Satz (Vollständigkeit von PD)

PD ist vollständig für die partielle Korrektheit von deterministischen Programmen.

Beachte:

Für die Vollständigkeit von PD ist es notwendig, dass alle gültigen Formeln in PD enthalten sind; denn es gibt kein vollständiges Beweissystem für die Menge aller gültigen Formeln.

Vollständigkeit von TD

Achtung:

Das System TD ist **nicht** vollständig für die totale Korrektheit von deterministischen Programmen, wenn sich nur Terminierungsfunktionen ausdrücken lassen, die Polynome sind (z.B. wenn nur die Funktionen $+$ und $\cdot$ zur Verfügung stehen).

Die so genannte Ausdruckskraft (siehe nächste Folie) lässt sich erreichen, indem die Menge der $\mathbf{integer}$-Ausdrücke so erweitert wird, dass alle partiell definierten berechenbaren Funktionen beschreibbar sind. Dann gilt:

Satz (Vollständigkeit von TD)

TD ist vollständig für die totale Korrektheit von deterministischen Programmen über ausdruckskräftigen $\mathbf{integer}$-Ausdrücken.

Ausdruckskraft

- Sei $S \equiv \textbf{while } B \textbf{ do } S_1 \textbf{ od}$ eine Schleife, x eine **integer**-Variable, die weder in B noch in S_1 vorkommt, und

$$S_x \equiv x := 0;\ \textbf{while } B \textbf{ do } x := x + 1;\ S_1 \textbf{ od}\,.$$

  Sei σ ein Zustand, so dass $\mathcal{M}_{tot}[\![S_x]\!](\sigma) = \{\tau\}$ mit $\tau \neq \perp$ ist. Dann bezeichnet $iter(S, \sigma)$ die natürliche Zahl $\tau(x)$.

  ($iter(S, \sigma)$ ist die Anzahl der Schleifeniterationen von S aus dem Zustand σ heraus.)

- Die Menge aller **integer**-Ausdrücke heißt ausdruckskräftig, wenn es für jede **while**-Schleife S einen **integer**-Ausdruck t gibt, so dass

$$\hat{\sigma}(t) = iter(S, \sigma)$$

  für jeden Zustand σ mit $\mathcal{M}_{tot}[\![S]\!](\sigma) \neq \{\perp\}$.

Zusätzliche Beweisregeln

A1 $\{p\}\ S\ \{p\}$ wobei $free(p) \cap change(S) = \emptyset$ — Invarianz I

A2 $$\frac{\{p\}\ S\ \{q\},\ \ \{r\}\ S\ \{q\}}{\{p \vee r\}\ S\ \{q\}}$$ — Disjunktion

A3 $$\frac{\{p_1\}\ S\ \{q_1\},\ \ \{p_2\}\ S\ \{q_2\}}{\{p_1 \wedge p_2\}\ S\ \{q_1 \wedge q_2\}}$$ — Konjunktion

A4 $$\frac{\{p\}\ S\ \{q\}}{\{\exists x : p\}\ S\ \{q\}}$$ wobei $x \notin var(S) \cup free(q)$ — $\exists$-Einführung

A5 $$\frac{\{r\}\ S\ \{q\}}{\{p \wedge r\}\ S\ \{p \wedge q\}}$$ wobei $free(p) \cap change(S) = \emptyset$ — Invarianz II

Korrektheit der zusätzlichen Beweisregeln

Satz (Korrektheit von A1–A5)

(1) Regel A1 ist korrekt für die partielle Korrektheit von deterministischen Programmen.

(2) Die Regeln A2–A5 sind korrekt für die partielle und die totale Korrektheit von deterministischen Programmen.