

Korrekte Software: Grundlagen und Methoden
Vorlesung 4 vom 25.04.16: Denotationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2016

18:10:57 2016-07-07

1 [25]



Beweisen

Zwei Programme c_0, c_1 sind äquivalent gdw. sie die gleichen Zustandsveränderungen bewirken. Formal definieren wir

Definition

$$c_0 \sim c_1 \text{ iff } \forall \sigma, \sigma'. \langle c_0, \sigma \rangle \rightarrow_{Stmt} \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \rightarrow_{Stmt} \sigma'$$

Ein einfaches Beispiel:

Lemma

Sei $w \equiv \text{while}(b) c$ mit $b \in \mathbf{Bexp}$, $c \in \mathbf{Stmt}$.
Dann gilt: $w \sim \text{if}(b) \{c; w\} \text{ else } \{\}$

Beweis an der Tafel

Korrekte Software

2 [25]



Fahrplan

- ▶ Einführung
- ▶ Die Floyd-Hoare-Logik
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Semantiken
- ▶ Verifikation: Vorwärts oder Rückwärts?
- ▶ Korrektheit des Hoare-Kalküls
- ▶ Einführung in Isabelle/HOL
- ▶ Weitere Datentypen: Strukturen und Felder
- ▶ Funktionen und Prozeduren
- ▶ Referenzen und Zeiger
- ▶ Frame Conditions & Modification Clauses
- ▶ Ausblick und Rückblick

Korrekte Software

3 [25]



Überblick

- ▶ Kleinsten Fixpunkt
- ▶ Denotationale Semantik für C0

Korrekte Software

4 [25]



Regeln und Regelinstanzen

Definition

Sei R eine Menge von Regeln $\frac{x_1 \dots x_n}{y}$, $n \geq 0$.
Die Anwendung einer Regel auf spezifische $a_1 \dots a_n$ ist eine Regelinstanz

▶ Betrachte folgende Regelmenge R

$$\frac{}{2^2} \quad \frac{}{2^3} \quad \frac{n \ m}{n \cdot m}$$

▶ Regelinstanzen sind

$$\frac{}{4} \quad \frac{}{8} \quad \frac{4 \ 8}{32} \quad \frac{4 \ 4}{16}$$

$$\frac{16 \ 32}{512} \quad \frac{3 \ 5}{15} \quad \dots$$

Korrekte Software

5 [25]



Induktive Definierte Mengen

Definition

Seit R eine Menge von Regelinstanzen und B eine Menge. Dann definieren wir

$$\hat{R}(B) = \{y \mid \exists x_1, \dots, x_k \subseteq B. \frac{x_1, \dots, x_k}{y} \in R\} \text{ und}$$

$$\hat{R}^0(B) = B \text{ und } \hat{R}^{i+1}(B) = \hat{R}(\hat{R}^i(B))$$

Korrekte Software

6 [25]



Beispiel

▶ Betrachte folgende Regelmenge R

$$\frac{}{2^2} \quad \frac{}{2^3} \quad \frac{n \ m}{n \cdot m}$$

▶ Was sind

$$\hat{R}^1(\emptyset) = \hat{R}(\emptyset) = \{4, 8\}$$

$$\hat{R}^2 = ?$$

$$\hat{R}^3 = ?$$

$$\hat{R}^{i+1} = ?$$

Korrekte Software

7 [25]



Induktive Definierte Mengen

Definition

Seit R eine Menge von Regelinstanzen und B eine Menge. Dann definieren wir

$$\hat{R}(B) = \{y \mid \exists x_1, \dots, x_k \subseteq B. \frac{x_1, \dots, x_k}{y} \in R\} \text{ und}$$

$$\hat{R}^0(B) = B \text{ und } \hat{R}^{i+1}(B) = \hat{R}(\hat{R}^i(B))$$

Definition (Abgeschlossen und Monoton)

▶ Eine Menge S ist abgeschlossen unter R (R -abgeschlossen) gdw.

$$\hat{R}(S) \subseteq S$$

▶ Eine Operation f ist monoton gdw.

$$\forall A, B. A \subseteq B \Rightarrow f(A) \subseteq f(B)$$

Korrekte Software

8 [25]



Kleinster Fixpunkt Operator

Lemma

Für jede Menge von Regelinstanzen R ist die induzierte Operation \hat{R} monoton.

Lemma

Sei $A_i = \hat{R}^i(\emptyset)$ für alle $i \in \mathbb{N}$ und $A = \bigcup_{i \in \mathbb{N}} A_i$. Dann gilt

- (a) A ist R -abgeschlossen,
- (b) $\hat{R}(A) = A$, und
- (c) A ist die kleinste R -abgeschlossene Menge.



Beweis von Lemma (a).

A ist R -abgeschlossen:

Sei $\frac{x_1, \dots, x_k}{y} \in R$ und $x_1, \dots, x_k \subseteq A$. Da $A = \bigcup_{i \in \mathbb{N}} A_i$ gibt es ein l so dass $x_1, \dots, x_k \subseteq A_l$. Also auch:
 $y \in \hat{R}(A_l) = \hat{R}(\hat{R}^l(\emptyset)) = \hat{R}^{l+1}(\emptyset) = A_{l+1} \subseteq A$. □

Beweis von Lemma (b): $\hat{R}(A) = A$.

► $\hat{R}(A) \subseteq A$:

Da A R -abgeschlossen gilt auch $\hat{R}(A) \subseteq A$.

► $A \subseteq \hat{R}(A)$:

Sei $y \in A$. Dann $\exists n > 0$, $y \in A_n$ und $y \notin \hat{R}(A_{n-1})$. Folglich muss es eine Regelinstanz $\frac{x_1, \dots, x_k}{y} \in R$ geben mit $x_1, \dots, x_k \subseteq A_{n-1} \subseteq A$. Also ist $y \in \hat{R}(A)$. □



Beweis von Lemma (c).

A ist die kleinste R -abgeschlossene Menge, d.h. für jede R -abgeschlossene Menge B gilt $A \subseteq B$.

Beweis per Induktion über n dass gilt $A_n \subseteq B$:

Basisfall $A_0 = \emptyset \subseteq B$

Induktionsschritt Da B R -abgeschlossen ist gilt: $\hat{R}(B) \subseteq B$.

Induktionsannahme: $A_n \subseteq B$.

Dann gilt $A_{n+1} = \hat{R}(A_n) \subseteq \hat{R}(B) \subseteq B$ weil \hat{R} monoton und B ist R -abgeschlossen. □



Kleinster Fixpunkt Operator

Definition

$$\text{fix}(\hat{R}) = \bigcup_{n \in \mathbb{N}} \hat{R}^n(\emptyset)$$

ist der kleinste Fixpunkt.



Kleinster Fixpunkt

► Betrachte folgende Regelmengen R

$$\frac{-}{2^2} \quad \frac{-}{2^3} \quad \frac{n \ m}{n \cdot m}$$

► Was sind

$$\hat{R}^1(\emptyset) = \hat{R}(\emptyset) = \{4, 8\}$$

$$\hat{R}^2 = ?$$

$$\hat{R}^3 = ?$$

$$\hat{R}^{i+1} = ?$$

► Wie sieht $\text{fix}(\hat{R})$ aus?



Denotationale Semantik - Motivation

► Operationale Semantik:

Eine Menge von Regeln, die einen Zustand und ein Programm in einen neuen Zustand oder Fehler überführen

$$\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

► Denotationale Semantik: Eine Menge von Regeln, die ein Programm in eine partielle Funktion von Zustand nach Zustand überführen

Denotat

$$\mathcal{D}[[c]] : \Sigma \rightarrow \Sigma$$



Denotationale Semantik - Motivation

Zwei Programme sind äquivalent gdw. sie immer zum selben Zustand (oder Fehler) auswerten

$$c_0 \sim c_1 \text{ iff } (\forall \sigma, \sigma'. \langle c_0, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \equiv \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma')$$

or

Zwei Programme sind äquivalent gdw. sie die selbe partielle Funktion **denotieren**

$$c_0 \sim c_1 \text{ iff } \{(\sigma, \sigma') \mid \langle c_0, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'\} = \{(\sigma, \sigma') \mid \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'\}$$



Denotierte Funktionen

► jeder $a : \mathbf{Aexp}$ denotiert eine partielle Funktion $\Sigma \rightarrow \mathbf{N}$

► jeder $b : \mathbf{Bexp}$ denotiert eine partielle Funktion $\Sigma \rightarrow \mathbf{T}$

► jedes $c : \mathbf{Stmt}$ denotiert eine partielle Funktion $\Sigma \rightarrow \Sigma$



Denotat von Aexp

$$\mathcal{E}[a] : \mathbf{Aexp} \rightarrow (\Sigma \rightarrow \mathbf{N})$$

$$\begin{aligned} \mathcal{E}[n] &= \{(\sigma, n) \mid \sigma \in \Sigma\} \\ \mathcal{E}[x] &= \{(\sigma, \sigma(x)) \mid \sigma \in \Sigma, x \in \text{Dom}(\sigma)\} \\ \mathcal{E}[a_0 + a_1] &= \{(\sigma, n_0 + n_1) \mid (\sigma, n_0) \in \mathcal{E}[a_0] \wedge (\sigma, n_1) \in \mathcal{E}[a_1]\} \\ \mathcal{E}[a_0 - a_1] &= \{(\sigma, n_0 - n_1) \mid (\sigma, n_0) \in \mathcal{E}[a_0] \wedge (\sigma, n_1) \in \mathcal{E}[a_1]\} \\ \mathcal{E}[a_0 * a_1] &= \{(\sigma, n_0 * n_1) \mid (\sigma, n_0) \in \mathcal{E}[a_0] \wedge (\sigma, n_1) \in \mathcal{E}[a_1]\} \\ \mathcal{E}[a_0 / a_1] &= \{(\sigma, n_0 / n_1) \mid (\sigma, n_0) \in \mathcal{E}[a_0] \wedge (\sigma, n_1) \in \mathcal{E}[a_1] \wedge n_1 \neq 0\} \end{aligned}$$



Denotat von Bexp

$$\mathcal{B}[a] : \mathbf{Bexp} \rightarrow (\Sigma \rightarrow \mathbf{T})$$

$$\begin{aligned} \mathcal{B}[1] &= \{(\sigma, 1) \mid \sigma \in \Sigma\} \\ \mathcal{B}[0] &= \{(\sigma, 0) \mid \sigma \in \Sigma\} \\ \mathcal{B}[a_0 == a_1] &= \{(\sigma, 1) \mid \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{E}[a_0](\sigma), \\ &\quad (\sigma, n_1) \in \mathcal{E}[a_1], n_0 = n_1\} \\ &\quad \cup \{(\sigma, 0) \mid \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{E}[a_0](\sigma), \\ &\quad (\sigma, n_1) \in \mathcal{E}[a_1], n_0 \neq n_1\} \\ \mathcal{B}[a_0 \leq a_1] &= \{(\sigma, 1) \mid \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{E}[a_0](\sigma), \\ &\quad (\sigma, n_1) \in \mathcal{E}[a_1], n_0 \leq n_1\} \\ &\quad \cup \{(\sigma, 0) \mid \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{E}[a_0](\sigma), \\ &\quad (\sigma, n_1) \in \mathcal{E}[a_1], n_0 > n_1\} \end{aligned}$$



Denotat von Bexp

$$\mathcal{B}[a] : \mathbf{Bexp} \rightarrow (\Sigma \rightarrow \mathbf{T})$$

$$\begin{aligned} \mathcal{B}[b] &= \{(\sigma, 1) \mid \sigma \in \Sigma, (\sigma, 0) \in \mathcal{B}[b]\} \\ &\quad \cup \{(\sigma, 0) \mid \sigma \in \Sigma, (\sigma, 1) \in \mathcal{B}[b]\} \\ \mathcal{B}[b_1 \ \&\& \ b_2] &= \{(\sigma, 0) \mid \sigma \in \Sigma, (\sigma, 0) \in \mathcal{B}[b_1]\} \\ &\quad \cup \{(\sigma, t_2) \mid \sigma \in \Sigma, (\sigma, 1) \in \mathcal{B}[b_1], (\sigma, t_2) \in \mathcal{B}[b_2]\} \\ \mathcal{B}[b_1 \ \parallel \ b_2] &= \{(\sigma, 1) \mid \sigma \in \Sigma, (\sigma, 1) \in \mathcal{B}[b_1]\} \\ &\quad \cup \{(\sigma, t_2) \mid \sigma \in \Sigma, (\sigma, 0) \in \mathcal{B}[b_1], (\sigma, t_2) \in \mathcal{B}[b_2]\} \end{aligned}$$



Denotat von Stmt

$$\mathcal{D}[\cdot] : \mathbf{Stmt} \rightarrow (\Sigma \rightarrow \Sigma)$$

$$\begin{aligned} \mathcal{D}[x = a] &= \{(\sigma, \sigma(x \mapsto n)) \mid \sigma \in \Sigma \wedge (\sigma, n) \in \mathcal{E}[a]\} \\ \mathcal{D}[\{c \ c_s\}] &= \mathcal{D}[c] \circ \mathcal{D}[c_s] \quad \text{Komposition von Relationen} \\ \mathcal{D}\{\{\}\} &= \text{Id} \quad \text{Id} := \{(\sigma, \sigma) \mid \sigma \in \Sigma\} \\ \mathcal{D}[\text{if } (b) \ c_0 \ \text{else } c_1] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[c_0]\} \\ &\quad \cup \{(\sigma, \sigma') \mid (\sigma, 0) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[c_1]\} \end{aligned}$$

Aber was ist

$$\mathcal{D}[\text{while } (b) \ c] = ??$$



Denotationale Semantik für while

Sei $w \equiv \text{while } (b) \ \text{do } c$ (und $\sigma \in \Sigma$). Wir wissen bereits, dass gilt

$$w \sim \text{if } (b) \ \{c; w\} \ \text{else } \{\}$$

$$\begin{aligned} \mathcal{D}[w] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[\{c; w\}]\} \\ &\quad \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\} \\ &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[w] \circ \mathcal{D}[c] \circ \text{Id}\} \\ &\quad \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\} \\ &= \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \wedge (\sigma'', \sigma') \in \mathcal{D}[w]\} \\ &\quad \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\} \end{aligned}$$



Denotationale Semantik von while

Sei $w \equiv \text{while } (b) \ c$ (und $\sigma \in \Sigma$). Wir wissen bereits, dass gilt

$$w = \text{if } (b) \ \{c; w\} \ \text{else } \{\}$$

$$\begin{aligned} \mathcal{D}[w]_0 &= \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b](\sigma)\} \\ \mathcal{D}[w]_1 &= \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \\ &\quad \wedge (\sigma'', \sigma') \in \mathcal{D}[w]_0\} \\ \mathcal{D}[w]_2 &= \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \\ &\quad \wedge (\sigma'', \sigma') \in \mathcal{D}[w]_1\} \\ &\vdots \\ \mathcal{D}[w]_{i+1} &= \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \\ &\quad \wedge (\sigma'', \sigma') \in \mathcal{D}[w]_i\} \end{aligned}$$

$$\Gamma(\varphi) = \{(\sigma, \sigma') \mid \exists \sigma''. \mathcal{B}[b](\sigma) = \text{true} \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \wedge (\sigma'', \sigma') \in \varphi\} \cup \{(\sigma, \sigma) \mid \mathcal{B}[b](\sigma) = \text{false}\}$$



Denotationale Semantik von while

Sei $w \equiv \text{while } (b) \ c$ (und $\sigma \in \Sigma$). Wir wissen bereits, dass gilt

$$w = \text{if } (b) \ \{c; w\} \ \text{else } \{\}$$

$$\Gamma(\psi) = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \wedge (\sigma'', \sigma') \in \psi\} \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\}$$

Γ ist wie \hat{R} , wobei R definiert ist wie folgt:

$$R = \left\{ \frac{(\sigma'', \sigma')}{(\sigma, \sigma')} \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma'') \in \mathcal{D}[c] \right\} \cup \left\{ \frac{}{(\sigma, \sigma)} \mid (\sigma, 0) \in \mathcal{B}[b] \right\}$$

und die Semantik von w ist der Fixpunkt von Γ , d.h. $\mathcal{D}[w] = \text{fix}(\Gamma)$



Denotation für Stmt

$$\mathcal{D}[\cdot] : \mathbf{Stmt} \rightarrow (\Sigma \rightarrow \Sigma)$$

$$\begin{aligned} \mathcal{D}[x = a] &= \{(\sigma, \sigma[x/X]) \mid \sigma \in \Sigma \wedge (\sigma, n) \in \mathcal{E}[a]\} \\ \mathcal{D}[\{c \ c_s\}] &= \mathcal{D}[c] \circ \mathcal{D}[c_s] \quad \text{Komposition von Relationen} \\ \mathcal{D}\{\{\}\} &= \text{Id} \quad \text{Id} := \{(\sigma, \sigma) \mid \sigma \in \Sigma\} \\ \mathcal{D}[\text{if } (b) \ c_0 \ \text{else } c_1] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[c_0]\} \\ &\quad \cup \{(\sigma, \sigma') \mid (\sigma, 0) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{D}[c_1]\} \\ \mathcal{D}[\text{while } (b) \ c] &= \text{fix}(\Gamma) \end{aligned}$$

mit

$$\Gamma(\psi) = \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \psi \circ \mathcal{D}[c]\} \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\}$$



Weitere Intuition zur Fixpunkt Konstruktion

- ▶ Sei $w \equiv \mathbf{while} (b) c$
- ▶ Zur Erinnerung: Wir haben begonnen mit $w \sim \mathbf{if} (b) \{ c; w \} \mathbf{else} \{ \}$
- ▶ Dann müsste auch gelten

$$\mathcal{D}[w] \stackrel{!}{=} \mathcal{D}[\mathbf{if} (b) \{ c; w \} \mathbf{else} \{ \}]$$

- ▶ Beweis an der Tafel

