

Korrekte Software: Grundlagen und Methoden
Vorlesung 4 vom 24.04.17: Denotationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2017

Fahrplan

- ▶ Einführung
- ▶ Die Floyd-Hoare-Logik
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Korrektheit des Hoare-Kalküls
- ▶ Vorwärts und Rückwärts mit Floyd und Hoare
- ▶ Funktionen und Prozeduren
- ▶ Referenzen und Speichermodelle
- ▶ Verifikationsbedingungen Revisited
- ▶ Vorwärtsrechnung Revisited
- ▶ Programmsicherheit und Frame Conditions
- ▶ Ausblick und Rückblick

Überblick

- ▶ Kleinster Fixpunkt

- ▶ Denotationale Semantik für C0

Fixpunkt

- ▶ Sei $f : A \rightarrow A$ eine Funktion. Ein **Fixpunkt** von f ist ein $a \in A$, so dass $f(a) = a$.
- ▶ Beispiele
 - ▶ Fixpunkte von $f(x) = \sqrt{x}$ sind 0 und 1; ebenfalls für $f(x) = x^2$.
 - ▶ Für die Sortierfunktion ist

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

$$\frac{-}{2^2}$$

$$\frac{-}{2^3}$$

$$\frac{n \quad m}{n \cdot m}$$

- ▶ Regelinstanzen sind

$$\frac{-}{4}$$

$$\frac{-}{8}$$

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

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

$$\frac{16 \quad 32}{512}$$

$$\frac{3 \quad 5}{15}$$

...

Induktive Definierte Mengen

Definition

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

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

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

Beispiel

- ▶ Betrachte folgende Regelmenge R

$$\frac{-}{2^2}$$

$$\frac{-}{2^3}$$

$$\frac{n \quad m}{n \cdot m}$$

- ▶ Was sind

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

$$\hat{R}^2(\emptyset) = ?$$

$$\hat{R}^3(\emptyset) = ?$$

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

Induktive Definierte Mengen

Definition

Sei 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.

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

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 j so dass $x_1, \dots, x_k \subseteq A_j$.

Also auch:

$$\begin{aligned} y \in \hat{R}(A_j) &= \hat{R}(\hat{R}^j(\emptyset)) \\ &= \hat{R}^{j+1}(\emptyset) \\ &= A_{j+1} \subseteq A. \end{aligned}$$



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.



Kleinsten Fixpunkt Operator

Definition

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

ist der kleinste Fixpunkt.

Kleinsten Fixpunkt

- ▶ Betrachte folgende Regelmenge R

$$\frac{-}{2^2}$$

$$\frac{-}{2^3}$$

$$\frac{n \quad m}{n \cdot m}$$

- ▶ Was sind

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

$$\hat{R}^2(\emptyset) = ?$$

$$\hat{R}^3(\emptyset) = ?$$

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

Kleinsten Fixpunkt

- ▶ Betrachte folgende Regelmenge R

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

- ▶ Was sind

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

$$\hat{R}^2(\emptyset) = ?$$

$$\hat{R}^3(\emptyset) = ?$$

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

- ▶ 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

$$\langle c, \sigma \rangle \rightarrow_{Stmnt} \sigma' \mid \perp$$

▶ Denotationale Semantik

Eine Menge von Regeln, die ein Programm in eine partielle Funktion
von Zustand nach Zustand überführen

Denotat

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

Denotationale Semantik - Motivation

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

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

oder

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

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

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**

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

$$\mathcal{A}[[n]] = \{(\sigma, n) \mid \sigma \in \Sigma\}$$

$$\mathcal{A}[[x]] = \{(\sigma, \sigma(x)) \mid \sigma \in \Sigma, x \in \text{Dom}(\sigma)\}$$

$$\mathcal{A}[[a_0 + a_1]] = \{(\sigma, n_0 + n_1) \mid (\sigma, n_0) \in \mathcal{A}[[a_0]] \wedge (\sigma, n_1) \in \mathcal{A}[[a_1]]\}$$

$$\mathcal{A}[[a_0 - a_1]] = \{(\sigma, n_0 - n_1) \mid (\sigma, n_0) \in \mathcal{A}[[a_0]] \wedge (\sigma, n_1) \in \mathcal{A}[[a_1]]\}$$

$$\mathcal{A}[[a_0 * a_1]] = \{(\sigma, n_0 * n_1) \mid (\sigma, n_0) \in \mathcal{A}[[a_0]] \wedge (\sigma, n_1) \in \mathcal{A}[[a_1]]\}$$

$$\mathcal{A}[[a_0/a_1]] = \{(\sigma, n_0/n_1) \mid (\sigma, n_0) \in \mathcal{A}[[a_0]] \wedge (\sigma, n_1) \in \mathcal{A}[[a_1]] \wedge n_1 \neq 0\}$$

Denotat von **Bexp**

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

$$\mathcal{B}[[1]] = \{(\sigma, 1) \mid \sigma \in \Sigma\}$$

$$\mathcal{B}[[0]] = \{(\sigma, 0) \mid \sigma \in \Sigma\}$$

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

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

Denotat von **Bexp**

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

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

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

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

Denotat von Stmt

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

$$\mathcal{C}[x = a] = \{(\sigma, \sigma[n/x]) \mid \sigma \in \Sigma \wedge (\sigma, n) \in \mathcal{A}[a]\}$$

$$\mathcal{C}[\{c \ c_s\}] = \mathcal{C}[c_s] \circ \mathcal{C}[c] \quad \text{Komposition von Relationen}$$

$$\mathcal{C}[\{\}] = \mathbf{Id} \quad \mathbf{Id} := \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$\begin{aligned} \mathcal{C}[\mathbf{if} (b) \ c_0 \ \mathbf{else} \ c_1] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_0]\} \\ &\cup \{(\sigma, \sigma') \mid (\sigma, 0) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_1]\} \end{aligned}$$

Denotat von Stmt

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

$$\mathcal{C}[x = a] = \{(\sigma, \sigma[n/x]) \mid \sigma \in \Sigma \wedge (\sigma, n) \in \mathcal{A}[a]\}$$

$$\mathcal{C}[\{c \ c_s\}] = \mathcal{C}[c_s] \circ \mathcal{C}[c] \quad \text{Komposition von Relationen}$$

$$\mathcal{C}[\{\}] = \mathbf{Id} \quad \mathbf{Id} := \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$\begin{aligned} \mathcal{C}[\mathbf{if} (b) \ c_0 \ \mathbf{else} \ c_1] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_0]\} \\ &\cup \{(\sigma, \sigma') \mid (\sigma, 0) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_1]\} \end{aligned}$$

Aber was ist

$$\mathcal{C}[\mathbf{while} (b) \ c] = ??$$

Denotationale Semantik für **while**

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

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

$$\begin{aligned} \mathcal{C}[[w]] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma') \in \mathcal{C}[[\{c w\}]]\} \\ &\quad \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]]\} \end{aligned}$$

Denotationale Semantik für **while**

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

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

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

Denotationale Semantik für **while**

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

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

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

Denotationale Semantik von **while**

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

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

$$\mathcal{C}[[w]]_0 = \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]](\sigma)\}$$

Denotationale Semantik von **while**

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

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

$$\mathcal{C}[[w]]_0 = \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]](\sigma)\}$$

$$\mathcal{C}[[w]]_1 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_0\}$$

Denotationale Semantik von **while**

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

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

$$\mathcal{C}[[w]]_0 = \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]](\sigma)\}$$

$$\mathcal{C}[[w]]_1 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_0\}$$

$$\mathcal{C}[[w]]_2 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_1\}$$

\vdots

Denotationale Semantik von **while**

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

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

$$\mathcal{C}[[w]]_0 = \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]](\sigma)\}$$

$$\mathcal{C}[[w]]_1 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_0\}$$

$$\mathcal{C}[[w]]_2 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_1\}$$

\vdots

$$\mathcal{C}[[w]]_{i+1} = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_i\}$$

Denotationale Semantik von **while**

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

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

$$\mathcal{C}[[w]]_0 = \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[[b]](\sigma)\}$$

$$\mathcal{C}[[w]]_1 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_0\}$$

$$\mathcal{C}[[w]]_2 = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_1\}$$

\vdots

$$\mathcal{C}[[w]]_{i+1} = \{(\sigma, \sigma') \mid \exists \sigma''. (\sigma, 1) \in \mathcal{B}[[b]] \wedge (\sigma, \sigma'') \in \mathcal{C}[[c]] \\ \wedge (\sigma'', \sigma') \in \mathcal{C}[[w]]_i\}$$

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

Denotationale Semantik von **while**

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

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

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

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

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

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

Denotation für **Stmt**

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

$$\mathcal{C}[x = a] = \{(\sigma, \sigma[n/X]) \mid \sigma \in \Sigma \wedge (\sigma, n) \in \mathcal{A}[a]\}$$

$$\mathcal{C}[\{c \ c_s\}] = \mathcal{C}[c_s] \circ \mathcal{C}[c] \quad \text{Komposition von Relationen}$$

$$\mathcal{C}[\{\}] = \mathbf{Id} \quad \mathbf{Id} := \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$\begin{aligned} \mathcal{C}[\mathbf{if} (b) \ c_0 \ \mathbf{else} \ c_1] &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_0]\} \\ &\quad \cup \{(\sigma, \sigma') \mid (\sigma, 0) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_1]\} \end{aligned}$$

$$\mathcal{C}[\mathbf{while} (b) \ c] = \mathit{fix}(\Gamma)$$

mit

$$\begin{aligned} \Gamma(\psi) &= \{(\sigma, \sigma') \mid (\sigma, 1) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \psi \circ \mathcal{C}[c]\} \\ &\quad \cup \{(\sigma, \sigma) \mid (\sigma, 0) \in \mathcal{B}[b]\} \end{aligned}$$

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

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

- ▶ Beweis an der Tafel

Fahrplan

- ▶ Einführung
- ▶ Die Floyd-Hoare-Logik
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Korrektheit des Hoare-Kalküls
- ▶ Vorwärts und Rückwärts mit Floyd und Hoare
- ▶ Funktionen und Prozeduren
- ▶ Referenzen und Speichermodelle
- ▶ Verifikationsbedingungen Revisited
- ▶ Vorwärtsrechnung Revisited
- ▶ Programmsicherheit und Frame Conditions
- ▶ Ausblick und Rückblick