

Korrekte Software: Grundlagen und Methoden
Vorlesung 4 vom 10.05.22
Äquivalenz der Operationalen und Denotationalen Semantik

Serge Autexier, Christoph Lüth

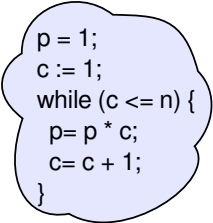
Universität Bremen

Sommersemester 2022

Fahrplan

- ▶ Einführung
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Der Floyd-Hoare-Kalkül I
- ▶ Der Floyd-Hoare-Kalkül II: Invarianten
- ▶ Korrektheit des Floyd-Hoare-Kalküls
- ▶ Strukturierte Datentypen
- ▶ Verifikationsbedingungen
- ▶ Vorwärts mit Floyd und Hoare
- ▶ Funktionen und Prozeduren I
- ▶ Funktionen und Prozeduren II
- ▶ Referenzen und Speichermodelle
- ▶ Ausblick und Rückblick

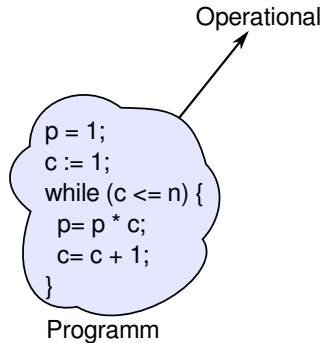
Operationale und Denotationale Semantik



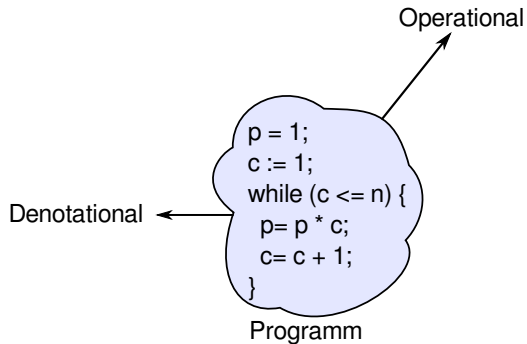
```
p = 1;  
c := 1;  
while (c <= n) {  
  p = p * c;  
  c = c + 1;  
}
```

Programm

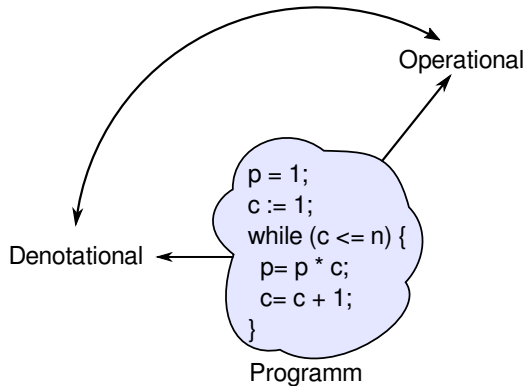
Operationale und Denotationale Semantik



Operationale und Denotationale Semantik



Operationale und Denotationale Semantik



Äquivalenz der Operationalen und Denotationalen Semantik

- ▶ Was müssen wir zeigen?

Äquivalenz der Operationalen und Denotationalen Semantik

- ▶ Was müssen wir zeigen?
- ▶ Auf oberster Ebene: für alle $c \in \mathbf{Stmt}$, $\sigma, \sigma' \in \Sigma$:

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket_c \quad (1)$$

- ▶ Semantik von Anweisungen ist über Semantik von Ausdrücken definiert, deshalb benötigen wir Hilfsaussagen

$$\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}} \quad (2)$$

$$\langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}} \quad (3)$$

- ▶ Wie zeigen wir das?

Operationale vs. denotationale Semantik

Operational $\langle a, \sigma \rangle \rightarrow_{Aexp} n$

$m \in \mathbf{Z}$

$$\langle m, \sigma \rangle \rightarrow_{Aexp} m$$

$x \in \mathbf{Loc}$

$$\frac{x \in Dom(\sigma)}{\langle x, \sigma \rangle \rightarrow_{Aexp} \sigma(x)}$$

Denotational $\llbracket a \rrbracket_{\mathcal{A}}$

$$\{(\sigma, m) \mid \sigma \in \Sigma\}$$

$$\{(\sigma, \sigma(x)) \mid \sigma \in \Sigma, x \in Dom(\sigma)\}$$

Operationale vs. denotationale Semantik

Operational $\langle a, \sigma \rangle \rightarrow_{Aexp} n$

$$a_1 \otimes a_2 \quad \frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} m}{\langle a_1 \otimes a_2, \sigma \rangle \rightarrow_{Aexp} n \otimes^I m}$$

$$\otimes \in \{+, *, -\}$$

$$a_1 / a_2 \quad \frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} m \quad m \neq 0}{\langle a_1 / a_2, \sigma \rangle \rightarrow_{Aexp} n \div m}$$

Denotational $\llbracket a \rrbracket_{\mathcal{A}}$

$$\{(\sigma, n \otimes^I m) \mid \sigma \in \Sigma, (\sigma, n) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, (\sigma, m) \in \llbracket a_2 \rrbracket_{\mathcal{A}}\}$$

$$\{(\sigma, n \div m) \mid \sigma \in \Sigma, (\sigma, n) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, (\sigma, m) \in \llbracket a_2 \rrbracket_{\mathcal{A}}, m \neq 0\}$$

Äquivalenz operationale und denotationale Semantik

- ▶ Zu zeigen Gleichung (3) von Folie 4:
- ▶ Für alle $a \in \mathbf{Aexp}$, für alle $n \in \mathbb{Z}$, für alle Zustände σ :

$$\langle a, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$$

- ▶ Beweis Prinzip?

Exkurs: Beweisprinzipien

- ▶ Induktion über $\mathbb{N}$ ($S(n)$ ist der **Nachfolger** von n):

$$\frac{P(0) \wedge \forall n \in \mathbb{N}. P(n) \implies P(S(n))}{\forall x \in \mathbb{N}. P(x)}$$

- ▶ Beispiel: Addition ist definiert durch

$$x + 0 = x$$

$$x + S(y) = S(x + y)$$

- ▶ Zeige $x + y = y + x$ durch Induktion über y :

- 1 Basis: $x + 0 = 0 + x$

- 2 Induktionsschritt: Annahme $x + y = y + x$, dann zeige $x + S(y) = S(y) + x$.

- ▶ Benötigt Hilfsbeweise $0 + x = x$ und $S(x + y) = S(x) + y$

Arbeitsblatt 4.1: Natürliche Induktion

- ▶ Zeigt durch natürliche Induktion:

$$0 + x = x \qquad S(x + y) = S(x) + y$$

- ▶ Welche Variable benutzt ihr für die Induktion? Was ist der Unterschied?

Wohlfundiertheit

Wohlfundiertheit

Eine binäre Relation $\prec \subseteq S \times S$ ist **wohlfundiert**, wenn es keine unendlich **absteigenden** Ketten gibt

$$\dots \prec a_3 \prec a_2 \prec a_1$$

Beispiele:

► $(\mathbb{N}, \leq)$?

Wohlfundiertheit

Wohlfundiertheit

Eine binäre Relation $\prec \subseteq S \times S$ ist **wohlfundiert**, wenn es keine unendlich **absteigenden** Ketten gibt

$$\dots \prec a_3 \prec a_2 \prec a_1$$

Beispiele:

► $(\mathbb{N}, \leq)$? Nein: $\dots \leq 1 \leq 1 \leq 1$

► $(\mathbb{N}, <)$?

Wohlfundiertheit

Wohlfundiertheit

Eine binäre Relation $\prec \subseteq S \times S$ ist **wohlfundiert**, wenn es keine unendlich **absteigenden** Ketten gibt

$$\dots \prec a_3 \prec a_2 \prec a_1$$

Beispiele:

- ▶ $(\mathbb{N}, \leq)$? Nein: $\dots \leq 1 \leq 1 \leq 1$
- ▶ $(\mathbb{N}, <)$? Ja.
- ▶ $(\mathbb{Z}, <)$?

Wohlfundiertheit

Wohlfundiertheit

Eine binäre Relation $\prec \subseteq S \times S$ ist **wohlfundiert**, wenn es keine unendlich **absteigenden** Ketten gibt

$$\dots \prec a_3 \prec a_2 \prec a_1$$

Beispiele:

- ▶ $(\mathbb{N}, \leq)$? Nein: $\dots \leq 1 \leq 1 \leq 1$
- ▶ $(\mathbb{N}, <)$? Ja.
- ▶ $(\mathbb{Z}, <)$? Nein: $\dots < -3 < -2 < -1 < 0$
- ▶ $(\mathbb{Q}^+, <)$?

Wohlfundiertheit

Wohlfundiertheit

Eine binäre Relation $\prec \subseteq S \times S$ ist **wohlfundiert**, wenn es keine unendlich **absteigenden** Ketten gibt

$$\dots \prec a_3 \prec a_2 \prec a_1$$

Beispiele:

- ▶ $(\mathbb{N}, \leq)$? Nein: $\dots \leq 1 \leq 1 \leq 1$
- ▶ $(\mathbb{N}, <)$? Ja.
- ▶ $(\mathbb{Z}, <)$? Nein: $\dots < -3 < -2 < -1 < 0$
- ▶ $(\mathbb{Q}^+, <)$? Nein: $\dots < \frac{1}{n} \dots < \frac{1}{4} < \frac{1}{3} < \frac{1}{2} < 1$

Eigenschaften wohlfundierter Relationen

- ▶ Eine wohlfundierte Relation ist **irreflexiv**: $\forall x \in S. x \not\prec x$

Eigenschaften wohlfundierter Relationen

- ▶ Eine wohlfundierte Relation ist **irreflexiv**: $\forall x \in S. x \not\prec x$
- ▶ Ansonsten gäbe es $\cdots \prec x \prec x \prec x$
- ▶ **Lemma**: $\prec$ ist wohlfundiert gdw. jede nicht-leere Untermenge $Q \subseteq S$ ein minimales Element $\min Q$ hat:

$$\min Q \in Q \wedge \forall b. b \prec \min Q \implies b \notin Q$$

Wohlfundierte Induktion

Noethersche Induktion (Wohlfundierte Induktion)

Sei $\prec \subseteq R \times R$ **wohlfundiert** und P eine Aussage über Elemente von R . Dann gilt

$$\frac{\forall v \in R. (\forall u \in R. u \prec v \implies P(u)) \implies P(v)}{\forall x \in R. P(x)}$$

Beispiele:

- ▶ Mit $S = \mathbb{N}$, $a \prec a + 1$: natürliche Induktion.
- ▶ Warum?

Wohlfundierte Induktion

Noethersche Induktion (Wohlfundierte Induktion)

Sei $\prec \subseteq R \times R$ **wohlfundiert** und P eine Aussage über Elemente von R . Dann gilt

$$\frac{\forall v \in R. (\forall u \in R. u \prec v \implies P(u)) \implies P(v)}{\forall x \in R. P(x)}$$

Beispiele:

- ▶ Mit $S = \mathbb{N}$, $a \prec a + 1$: natürliche Induktion.
- ▶ Warum? Fallunterscheidung über v : entweder $v = 0$, dann gibt es kein u so dass $u \prec 0$ und die Voraussetzung ist $P(0)$; oder $v = w + 1$, dann $w \prec w + 1$, und die Voraussetzung ist $P(w) \implies P(w + 1)$

Strukturelle Ordnung

Strukturelle Ordnung

Die strukturelle Ordnung auf arithmetischen Ausdrücken ist definiert als:

$$\forall a, a' \in \mathbf{Aexp.}, a' \prec a \iff a' \text{ ist Teilausdruck von } a$$

Dabei ist “Teilausdruck” formalisiert als $\otimes \in \{+, *, -, /\}$:

$$a \text{ Teilausdruck-von } (a_1 \otimes a_2) \iff \begin{pmatrix} a = a_1 \vee a \text{ Teilausdruck-von } a_1 \vee \\ a = a_2 \vee a \text{ Teilausdruck-von } a_2 \end{pmatrix}$$

- Beispiel für strukturelle Induktion: Rechtseindeutigkeit von $\llbracket - \rrbracket_{\mathcal{A}}$ ($\longrightarrow$ Vorlesung 3)

Arbeitsblatt 4.2: Strukturelle Induktion

- ▶ **Beweist**, dass die Relation “Teilausdruck-von” wohlfundiert ist.

Äquivalenz operationale und denotationale Semantik

- Für alle $a \in \mathbf{Aexp}$, für alle $n \in \mathbb{Z}$, für alle Zustände σ :

$$\langle a, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$$

- Beweis Prinzip?

Äquivalenz operationale und denotationale Semantik

- Für alle $a \in \mathbf{Aexp}$, für alle $n \in \mathbb{Z}$, für alle Zustände σ :

$$\langle a, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$$

- Beweis per struktureller Induktion über a . (Warum?)

Beweis: $\forall a \in \mathbf{Aexp}. \forall n \in \mathbb{Z}. \forall \sigma. \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$

Induktionsanfänge

► $a \equiv m \in \mathbf{Z}$:

$$\left[\begin{array}{l} \langle m, \sigma \rangle \rightarrow_{\mathbf{Aexp}} \llbracket m \rrbracket \\ \llbracket m \rrbracket_{\mathcal{A}} = \{(\sigma', \llbracket m \rrbracket) \mid \sigma' \in \Sigma\} \Rightarrow (\sigma, \llbracket m \rrbracket) \in \llbracket m \rrbracket_{\mathcal{A}} \end{array} \right] \iff$$

► $a \equiv X \in \mathbf{Loc}$:

① $X \in \text{Dom}(\sigma)$:

$$\left[\begin{array}{l} \langle X, \sigma \rangle \rightarrow_{\mathbf{Aexp}} \sigma(X) \\ \llbracket X \rrbracket_{\mathcal{A}} = \{(\sigma', \sigma'(X)) \mid \sigma' \in \Sigma, X \in \text{Dom}(\sigma')\} \Rightarrow (\sigma, \sigma(X)) \in \llbracket X \rrbracket_{\mathcal{A}} \end{array} \right] \iff$$

② $X \notin \text{Dom}(\sigma)$:

$$\left[\begin{array}{l} \langle X, \sigma \rangle \rightarrow_{\mathbf{Aexp}} \perp \\ \llbracket X \rrbracket_{\mathcal{A}} = \{(\sigma', \sigma'(X)) \mid \sigma' \in \Sigma, X \in \text{Dom}(\sigma')\} \Rightarrow \sigma \notin \text{Dom}(\llbracket X \rrbracket_{\mathcal{A}}) \end{array} \right] \iff$$

Beweis: $\forall a \in \mathbf{Aexp}. \forall n \in \mathbb{Z}. \forall \sigma. \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$

Induktionsschritte

- ▶ $a \equiv a_1 + a_2$ — Induktionsannahme: für alle m, n

$$\langle a_1, \sigma \rangle \rightarrow_{Aexp} m \iff (\sigma, m) \in \llbracket a_1 \rrbracket_A$$

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a_2 \rrbracket_A$$

Dann;

$$\langle a_1 + a_2, \sigma \rangle \rightarrow_{Aexp} m + n \xrightarrow{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{Aexp})} \langle a_1, \sigma \rangle \rightarrow_{Aexp} m \xleftrightarrow{\text{IA für } a_1} (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

&

&

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \xleftrightarrow{\text{IA für } a_2} (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

$$\begin{array}{c} \Updownarrow (Def. [\cdot]_{\mathcal{A}}) \\ (\sigma, m+n) \in \llbracket a_1 + a_2 \rrbracket_{\mathcal{A}} \end{array}$$

Beweis: $\forall a \in \mathbf{Aexp}. \forall n \in \mathbb{Z}. \forall \sigma. \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$

Induktionsschritte

► $a \equiv a_1/a_2$ — Induktionsannahme:

$$\langle a_1, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \iff (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

$$\langle a_2, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \iff (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

① Fall: $n \neq 0$

$$\langle a_1/a_2, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m/n \xLeftrightarrow{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Aexp}})} \langle a_1, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \xLeftrightarrow{\text{IA für } a_1} (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

&

&

$$\langle a_2, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \xLeftrightarrow{\text{IA für } a_2} (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

$$\Updownarrow (\text{Def. } \llbracket \cdot \rrbracket_{\mathcal{A}})$$

$$(\sigma, m/n) \in \llbracket a_1/a_2 \rrbracket_{\mathcal{A}}$$

Beweis: $\forall a \in \mathbf{Aexp}. \forall n \in \mathbb{Z}. \forall \sigma. \langle a, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$

Induktionsschritte

► $a \equiv a_1/a_2$ — Induktionsannahme:

$$\langle a_1, \sigma \rangle \rightarrow_{Aexp} m \iff (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

① Fall: $n = 0$

Dann gibt es kein v so dass $\langle a_1/a_2, \sigma \rangle \rightarrow_{Aexp} v$, aber auch $\sigma \notin \text{dom} \llbracket a_1/a_2 \rrbracket_{\mathcal{A}}$.

q.e.d.

Operationale vs. denotationale Semantik

Operational $\langle b, \sigma \rangle \rightarrow_{Bexp} false \mid true$

1 $\langle \mathbf{1}, \sigma \rangle \rightarrow_{Bexp} true$

0 $\langle \mathbf{0}, \sigma \rangle \rightarrow_{Bexp} false$

Denotational $\llbracket b \rrbracket_{\mathcal{B}}$

$\{(\sigma, true) \mid \sigma \in \Sigma\}$

$\{(\sigma, false) \mid \sigma \in \Sigma\}$

Operationale vs. denotationale Semantik

Operat. $\langle b, \sigma \rangle \rightarrow_{Bexp} t$

$$a_0 == a_1 \quad \frac{\frac{\langle a_0, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_1, \sigma \rangle \rightarrow_{Aexp} m \quad n = m}{\langle a_0 == a_1, \sigma \rangle \rightarrow_{Bexp} true} \quad \frac{\langle a_0, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_1, \sigma \rangle \rightarrow_{Aexp} m \quad n \neq m}{\langle a_0 == a_1, \sigma \rangle \rightarrow_{Bexp} false}$$

$a_1 < a_2$

Denotational $\llbracket b \rrbracket_{\mathcal{B}}$

$$\{(\sigma, true) \mid \sigma \in \Sigma, \\ (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}, \\ (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, \\ n_0 = n_1 \}$$

$\cup$

$$\{(\sigma, false) \mid \sigma \in \Sigma, \\ (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}, \\ (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, \\ n_0 \neq n_1 \}$$

analog

Operationale vs. denotationale Semantik

Operational $\langle a, \sigma \rangle \rightarrow_{Bexp} b$

$$b_1 \ \&\& \ b_2 \quad \frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} false}{\langle b_1 \ \&\& \ b_2, \sigma \rangle \rightarrow false}$$

$$\frac{\begin{array}{l} \langle b_1, \sigma \rangle \rightarrow_{Bexp} true \\ \langle b_2, \sigma \rangle \rightarrow_{Bexp} t \end{array}}{\langle b_1 \ \&\& \ b_2, \sigma \rangle \rightarrow t}$$

$b_1 \ || \ b_2$

$!n$

...

Denotational $\llbracket b \rrbracket_{\mathcal{B}}$

$$\{(\sigma, false) \mid (\sigma, false) \in \llbracket b_1 \rrbracket_{\mathcal{B}}\}$$

$$\{(\sigma, t) \mid (\sigma, true) \in \llbracket b_1 \rrbracket_{\mathcal{B}}, (\sigma, t) \in \llbracket b_2 \rrbracket_{\mathcal{B}}\}$$

analog

Äquivalenz operationale und denotationale Semantik

- ▶ Zu zeigen Gleichung (2) von Folie 4:
- ▶ Für alle $b \in \mathbf{Bexp}$, für alle $t \in \mathbb{B}$, for alle Zustände σ :

$$\langle b, \sigma \rangle \rightarrow_{Bexp} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}}$$

- ▶ Beweis Prinzip?

Äquivalenz operationale und denotationale Semantik

- ▶ Zu zeigen Gleichung (2) von Folie 4:
- ▶ Für alle $b \in \mathbf{Bexp}$, für alle $t \in \mathbb{B}$, for alle Zustände σ :

$$\langle b, \sigma \rangle \rightarrow_{Bexp} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}}$$

- ▶ Beweis per struktureller Induktion über b (unter Verwendung der Äquivalenz für AExp).
(Warum?)

Beweis $\langle b, \sigma \rangle \rightarrow_{Bexp} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}}$

Induktionsanfänge

► $b \equiv 0$:

$$\left[\begin{array}{l} \langle \mathbf{0}, \sigma \rangle \rightarrow_{Bexp} false \\ \llbracket \mathbf{0} \rrbracket_{\mathcal{A}} = \{(\sigma', false) \mid \sigma' \in \Sigma\} \Rightarrow (\sigma, false) \in \llbracket \mathbf{0} \rrbracket_{\mathcal{B}} \end{array} \right] \iff$$

► $b \equiv 1$:

$$\left[\begin{array}{l} \langle \mathbf{1}, \sigma \rangle \rightarrow_{Bexp} true \\ \llbracket \mathbf{1} \rrbracket_{\mathcal{A}} = \{(\sigma', true) \mid \sigma' \in \Sigma\} \Rightarrow (\sigma, true) \in \llbracket \mathbf{1} \rrbracket_{\mathcal{B}} \end{array} \right] \iff$$

Beweis $\langle b, \sigma \rangle \rightarrow_{Bexp} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}}$

Induktionsschritte

► $b \equiv b_1 \&\& b_2$ — Induktionsannahme:

$$\langle b_1, \sigma \rangle \rightarrow_{Bexp} v \iff (\sigma, v) \in \llbracket b_1 \rrbracket_{\mathcal{B}}$$

$$\langle b_2, \sigma \rangle \rightarrow_{Bexp} w \iff (\sigma, w) \in \llbracket b_2 \rrbracket_{\mathcal{B}}$$

① Fall $v = false$

$$\begin{aligned} \langle b_1 \&\& b_2, \sigma \rangle \rightarrow_{Bexp} false &\stackrel{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{Bexp})}{\iff} \langle b_1, \sigma \rangle \rightarrow_{Bexp} false &\stackrel{\text{IA f\"ur } b_1}{\iff} (\sigma, false) \in \llbracket b_1 \rrbracket_{\mathcal{B}} \\ & & \stackrel{\text{Def. } \llbracket \cdot \rrbracket_{\mathcal{B}}}{\iff} (\sigma, false) \in \llbracket b_1 \&\& b_2 \rrbracket_{\mathcal{B}} \end{aligned}$$

Beweis $\langle b, \sigma \rangle \rightarrow_{Bexp} t \iff (\sigma, t) \in \llbracket b \rrbracket_{\mathcal{B}}$

Induktionsschritte

► $b \equiv b_1 \&\& b_2$ — Induktionsannahme:

$$\langle b_1, \sigma \rangle \rightarrow_{Bexp} v \iff (\sigma, v) \in \llbracket b_1 \rrbracket_{\mathcal{B}}$$

$$\langle b_2, \sigma \rangle \rightarrow_{Bexp} w \iff (\sigma, w) \in \llbracket b_2 \rrbracket_{\mathcal{B}}$$

① Fall $v = true$

$$\langle b_1 \&\& b_2, \sigma \rangle \rightarrow_{Bexp} w \xLeftrightarrow{\text{(Def. } \langle \cdot, \cdot \rangle \rightarrow_{Bexp} \cdot \text{)}} \langle b_1, \sigma \rangle \rightarrow_{Bexp} true \xLeftrightarrow{\text{IA für } b_1} (\sigma, true) \in \llbracket b_1 \rrbracket_{\mathcal{B}}$$

&

&

$$\langle b_2, \sigma \rangle \rightarrow_{Bexp} w \xLeftrightarrow{\text{IA für } b_2} (\sigma, w) \in \llbracket b_2 \rrbracket_{\mathcal{B}}$$

$$\text{Def. } \llbracket \cdot \rrbracket_{\mathcal{B}} \updownarrow$$

$$(\sigma, w) \in \llbracket b_1 \&\& b_2 \rrbracket_{\mathcal{B}}$$

Arbeitsblatt 4.3: Beweis Induktionsanfang

$$\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} v \iff (\sigma, v) \in \llbracket a_1 == a_2 \rrbracket_{\mathcal{B}}$$

Beweist obige Aussage unter Verwendung des für arithmetische Ausdrücke geltenden Lemmas

$$\forall a \in \mathbf{Aexp}. \forall n \in \mathbb{Z}. \forall \sigma. \langle a, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}$$

- 1 Was sind die Annahmen?
- 2 Welche Fälle unterscheiden wir?

Beweis $\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} v \iff (\sigma, v) \in \llbracket a_1 == a_2 \rrbracket_{\mathcal{B}}$

► Annahmen: für $n, m \in \mathbb{B}$:

$$\langle a_1, \sigma \rangle \rightarrow_{Aexp} m \iff (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{B}}$$

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{B}}$$

► 1. Fall: $v = true$ ($m = n$)

$$\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} true \xrightarrow{\text{(Def. } \langle \cdot, \cdot \rangle \rightarrow_{Bexp} \cdot \text{)}} \langle a_1, \sigma \rangle \rightarrow_{Bexp} m \xrightarrow{\text{Annahme für } a_1} (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

&

&

$$\langle a_2, \sigma \rangle \rightarrow_{Bexp} m \xrightarrow{\text{Annahme für } a_2} (\sigma, m) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

$$\text{Def. } \llbracket \cdot \rrbracket_{\mathcal{B}} \updownarrow$$

$$(\sigma, true) \llbracket a_1 == a_2 \rrbracket_{\mathcal{B}}$$

Beweis $\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} v \iff (\sigma, v) \in \llbracket a_1 == a_2 \rrbracket_{\mathcal{B}}$

► Annahmen: für $m, n \in \mathbb{B}$:

$$\langle a_1, \sigma \rangle \rightarrow_{Aexp} m \iff (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{B}}$$

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \iff (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{B}}$$

► 2. Fall: $v = false$ ($m \neq n$)

$$\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} false \stackrel{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{Bexp} \cdot)}{\iff} \langle a_1, \sigma \rangle \rightarrow_{Aexp} m \stackrel{\text{Annahme für } a_1}{\iff} (\sigma, m) \in \llbracket a_1 \rrbracket_{\mathcal{A}}$$

&

&

$$\langle a_2, \sigma \rangle \rightarrow_{Aexp} n \stackrel{\text{Annahme für } a_2}{\iff} (\sigma, n) \in \llbracket a_2 \rrbracket_{\mathcal{A}}$$

Def. $\llbracket \cdot \rrbracket_{\mathcal{B}}$

$$(\sigma, false) \llbracket a_1 == a_2 \rrbracket_{\mathcal{B}}$$

Operationale vs. denotationale Semantik

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

$\{\}$

$$\frac{}{\langle \{\}, \sigma \rangle \rightarrow_{Stmt} \sigma}$$

$c_1; c_2$

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{Stmt} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{Stmt} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{Stmt} \sigma''}$$

$x = a$

$$\frac{\langle a, \sigma \rangle \rightarrow_{Aexp} n}{\langle x = a, \sigma \rangle \rightarrow_{Stmt} \sigma[x \mapsto n]}$$

Denotational $\llbracket c \rrbracket_c$

$$\llbracket \{\} \rrbracket_c = Id$$

$$\llbracket c_1 \rrbracket_c \circ \llbracket c_2 \rrbracket_c$$

$$\{(\sigma, \sigma[x \mapsto n]) \mid (\sigma, n) \in \llbracket a \rrbracket_{\mathcal{A}}\}$$

Operationale vs. denotationale Semantik

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

$$\text{if } (b) \ c_0 \quad \frac{\langle b, \sigma \rangle \rightarrow_{Bexp} true \quad \langle c_0, \sigma \rangle \rightarrow_{Stmt} \sigma'}{\langle c, \sigma \rangle \rightarrow_{Stmt} \sigma'}$$

$$\text{else } c_1 \quad \frac{\langle b, \sigma \rangle \rightarrow_{Bexp} false \quad \langle c_1, \sigma \rangle \rightarrow_{Stmt} \sigma'}{\langle c, \sigma \rangle \rightarrow_{Stmt} \sigma'}$$

Denotational $\llbracket c \rrbracket_c$

$$\{(\sigma, \sigma') \mid (\sigma, true) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_0 \rrbracket_c\}$$

$$\{(\sigma, \sigma') \mid (\sigma, false) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c\}$$

Operationale vs. denotationale Semantik

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

Denotational $\llbracket c \rrbracket_c$

while (b) c
w

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} false}{\langle w, \sigma \rangle \rightarrow_{Stmt} \sigma}$$

$fix(\Gamma)$

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} true \quad \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \quad \langle w, \sigma' \rangle \rightarrow_{Stmt} \sigma''}{\langle w, \sigma \rangle \rightarrow_{Stmt} \sigma''}$$

mit

$$\begin{aligned} \Gamma(\varphi) = & \{(\sigma, \sigma') \mid (\sigma, true) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c \rrbracket_c \circ \varphi\} \\ & \cup \{(\sigma, \sigma) \mid (\sigma, false) \in \llbracket b \rrbracket_B\} \end{aligned}$$

Äquivalenz operationale und denotationale Semantik

- ▶ Zu zeigen Gleichung (1) von Folie 4:
- ▶ Für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ' :

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket c$$

- ▶ $\implies$ Beweis Prinzip?
- ▶ $\impliedby$ Beweis Prinzip?

Operationale Semantik: C0 Programme

► $\text{Stmt}_C ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

$$\begin{array}{c}
 \frac{}{\langle \{ \}, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma} \quad \frac{\langle a, \sigma \rangle \rightarrow_{\text{Aexp}} n \in \mathbb{Z}}{\langle x = a, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma[x \mapsto n]} \quad \frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''} \\
 \\
 \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'} \quad \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'} \\
 \\
 \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma} \\
 \\
 \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle \text{while } (b) \ c, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}
 \end{array}$$

Operationale Semantik: C0 Programme

► $\text{Stmt}_C ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

Programmstruktur

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle \text{while } (b) \ c, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma}$$



Operationale Semantik: C0 Programme

► $\text{Stmt}_C ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

Programmstruktur

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle \text{while } (b) \ c, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma}$$



Strukturelle Induktion
über c **nicht** möglich.

Ableitungstiefe für Programme

- Die Ableitungstiefe einer Programmauswertung mittels Regeln der operationaler Semantik ist die **Anzahl der Regelanwendungen** mit Conclusion der Form $\langle ., . \rangle \rightarrow_{Stmt} ..$

$$\frac{\begin{array}{c} \vdots \\ \text{Prämisse}_1 \end{array} \quad \cdots \quad \begin{array}{c} \vdots \\ \text{Prämisse}_n \end{array}}{\text{Conclusion}}$$

Operationale Semantik: C0 Programme

► $\text{Stmt}_C ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

Programmstruktur

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle \text{while } (b) \ c, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma}$$



Operationale Semantik: C0 Programme

► $\text{Stmt}_C ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

Programmstruktur

Ableitungstiefe

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$



Äquivalenz operationale und denotationale Semantik

- Für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ' :

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket c$$

- $\implies$ Beweis Prinzip?

- $\impliedby$ Beweis Prinzip?

Äquivalenz operationale und denotationale Semantik

- Für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ' :

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket_c$$

- $\implies$ Beweis per Induktion über **die (Tiefe der) Ableitung** in der operationalen Semantik (Warum?)
- $\impliedby$ Beweis Prinzip?

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsanfang — Ableitungstiefe 1

► Fall $c \equiv x = a$:

$$\llbracket x = a \rrbracket_c = \{(\sigma, \sigma[x \mapsto m]) \mid (\sigma, m) \in \llbracket a \rrbracket_{\mathcal{A}}\}$$

Sei $\langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \in \mathbb{Z}$:

$$\begin{array}{ccc} \langle x = a, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma[x \mapsto m] & & \\ \updownarrow (\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} \cdot) & & \\ \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \in \mathbb{Z} & \xleftrightarrow{\text{Lemma für } a} & (\sigma, m) \in \llbracket a \rrbracket_{\mathcal{A}} \\ & & \downarrow \text{Def. } \llbracket \cdot \rrbracket_c \\ & & (\sigma, \sigma[x \mapsto m]) \in \llbracket x = a \rrbracket_c \end{array}$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsanfang — Ableitungstiefe 1

► Fall $c \equiv x = a$:

$$\llbracket x = a \rrbracket_c = \{(\sigma, \sigma[x \mapsto m]) \mid (\sigma, m) \in \llbracket a \rrbracket_{\mathcal{A}}\}$$

Sei $\langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \in \mathbb{Z}$:

$$\begin{array}{ccc} \langle x = a, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma[x \mapsto m] & & \\ \updownarrow (\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} \cdot) & & \\ \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} m \in \mathbb{Z} & \xleftrightarrow{\text{Lemma für } a} & (\sigma, m) \in \llbracket a \rrbracket_{\mathcal{A}} \\ & & \downarrow \text{Def. } \llbracket \cdot \rrbracket_c \\ & & (\sigma, \sigma[x \mapsto m]) \in \llbracket x = a \rrbracket_c \end{array}$$

► Fall $c \equiv \{\}$: ...

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsschritt:

► Fall $c \equiv \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2$:

$$\begin{aligned} \llbracket \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2 \rrbracket_c = & \{ (\sigma, \sigma') \mid (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c, (\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B \} \\ & \cup \{ (\sigma, \sigma') \mid (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c, (\sigma, \mathbf{false}) \in \llbracket b \rrbracket_B \} \end{aligned}$$

► Fall $\langle \sigma, b \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{true}$ mit $\langle c_1, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$:

$$\langle \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \xLeftrightarrow{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} \cdot)} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{true} \xLeftrightarrow{\text{Lemma für } b} (\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B$$

&

&

$$\langle c_1, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \xRightarrow{\text{IH für } c_1} (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c$$

$$\text{Def. } \llbracket \cdot \rrbracket_c \Downarrow$$

$$(\sigma, \sigma') \in \llbracket \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2 \rrbracket_c$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsschritt:

► Fall $c \equiv \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2$:

$$\begin{aligned} \llbracket \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2 \rrbracket_c = & \{ (\sigma, \sigma') \mid (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c, (\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B \} \\ & \cup \{ (\sigma, \sigma') \mid (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c, (\sigma, \mathbf{false}) \in \llbracket b \rrbracket_B \} \end{aligned}$$

► Fall $\langle \sigma, b \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{false}$ mit $\langle c_2, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$:

$$\langle \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \xLeftrightarrow{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} \cdot)} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{false} \xLeftrightarrow{\text{Lemma für } b} (\sigma, \mathbf{false}) \in \llbracket b \rrbracket_B$$

&

&

$$\langle c_2, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \xRightarrow{\text{IH für } c_2} (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c$$

$$\text{Def. } \llbracket \cdot \rrbracket_c \Downarrow$$

$$(\sigma, \sigma') \in \llbracket \mathbf{if}(b) \ c_1 \ \mathbf{else} \ c_2 \rrbracket_c$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsschritt:

► Fall $c \equiv \mathbf{while}(b) c$:

$$\llbracket \mathbf{while}(b) c \rrbracket_c = \text{fix}(\Gamma)$$

► Fall $\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \text{true}$ mit $\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma', \langle \mathbf{while}(b) c, \sigma' \rangle \rightarrow_{\mathbf{Stmt}} \sigma''$

$$\langle \mathbf{while}(b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'' \xLeftrightarrow{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} \cdot)} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \text{true} \xLeftrightarrow{\text{Lemma für } b} (\sigma, \text{true}) \in \llbracket b \rrbracket_{\mathcal{B}}$$

&

&

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \xRightarrow{\text{IH für } \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'} (\sigma, \sigma') \in \llbracket c \rrbracket_c$$

&

&

$$\langle \mathbf{while}(b) c, \sigma' \rangle \rightarrow_{\mathbf{Stmt}} \sigma'' \xRightarrow{\text{IH für } \langle \mathbf{while}(b) c, \sigma' \rangle \rightarrow_{\mathbf{Stmt}} \sigma''} (\sigma', \sigma'') \in \llbracket \mathbf{while}(b) c \rrbracket_c$$

Def. $\llbracket \cdot \rrbracket_c$
 $\Downarrow$

$$(\sigma, \sigma'') \in \llbracket \mathbf{while}(b) c \rrbracket_c$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \implies (\sigma, \sigma') \in \llbracket c \rrbracket_c$

Induktionsschritt:

► Fall $c \equiv \mathbf{while}(b) c$:

$$\llbracket \mathbf{while}(b) c \rrbracket_c = \text{fix}(\Gamma)$$

► Fall $\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \text{false}, \langle \mathbf{while}(b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma$

$$\begin{aligned} \langle \mathbf{while}(b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma &\stackrel{(\text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}})}{\iff} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \text{false} \stackrel{\text{Lemma für } b}{\iff} (\sigma, \text{false}) \in \llbracket b \rrbracket_b \\ &\stackrel{\text{Def. } \llbracket \cdot \rrbracket_c}{\Downarrow} \\ &(\sigma, \sigma) \in \llbracket \mathbf{while}(b) c \rrbracket_c \end{aligned}$$

Äquivalenz operationale und denotationale Semantik

- Für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ' :

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket_c$$

- $\implies$ Beweis per Induktion über **die (Tiefe der) Ableitung** in der operationalen Semantik (Warum?)
- $\impliedby$ Beweis Prinzip?

Äquivalenz operationale und denotationale Semantik

- Für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ' :

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket_c$$

- $\implies$ Beweis per Induktion über **die (Tiefe der) Ableitung** in der operationalen Semantik (Warum?)
- $\impliedby$ Beweis per struktureller Induktion über c (Verwendung der Äquivalenz für arithmetische und boolsche Ausdrücke). Für die While-Schleife Rückgriff auf Definition des Fixpunkts und Induktion über die Teilmengen $\Gamma^i(\emptyset)$ des Fixpunkts. (Warum?)

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsanfang:

► Fall $c \equiv x = a$:

$$\llbracket x = a \rrbracket_c = \{(\sigma, \sigma[x \mapsto t]) \mid (\sigma, t) \in \llbracket a \rrbracket_A\}$$

$$(\sigma, \sigma[x \mapsto t]) \in \llbracket x = a \rrbracket_c \wedge \underbrace{(\sigma, t) \in \llbracket a \rrbracket_A}$$

Lemma **Aexp**
 $\implies$

$$\langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} t$$

$$\text{Def. } \langle \cdot, \cdot \rangle \xrightarrow{\mathbf{Stmt}} \langle x = a, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma[x \mapsto t]$$

► Fall $c \equiv \{\}$

$$\llbracket \{\} \rrbracket_c = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$(\sigma, \sigma) \in \llbracket \{\} \rrbracket_c$$

$$\text{Def. } \langle \cdot, \cdot \rangle \xrightarrow{\mathbf{Stmt}} \langle \{\}, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsschritt:

► Fall **if** (b) c_1 **else** c_2 :

$$\llbracket \mathbf{if} (b) c_1 \mathbf{else} c_2 \rrbracket_c = \{(\sigma, \sigma') \mid (\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c\} \\ \cup \{(\sigma, \sigma') \mid (\sigma, \mathbf{false}) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c\}$$

Induktionsannahme gilt für c_1 und c_2

► Fall: $(\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B$ mit $(\sigma, \sigma') \in \llbracket c_1 \rrbracket_c$

$$\begin{array}{ll} & (\sigma, \mathbf{true}) \in \llbracket b \rrbracket_B \wedge (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c \\ \text{Lemma } \mathbf{Bexp} & \implies \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{true} \wedge (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c \\ \text{IA für } c_1 & \implies \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{true} \wedge \langle c_1, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \\ \text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\mathbf{Stmt}} & \implies \langle \mathbf{if} (b) c_1 \mathbf{else} c_2, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \end{array}$$

Beweis: $\forall c \in \text{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$

Induktionsschritt:

► Fall **if** (*b*) *c*₁ **else** *c*₂:

$$\llbracket \text{if } (b) \ c_1 \ \text{else} \ c_2 \rrbracket_c = \{(\sigma, \sigma') \mid (\sigma, \text{true}) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_1 \rrbracket_c\} \\ \cup \{(\sigma, \sigma') \mid (\sigma, \text{false}) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c\}$$

Induktionsannahme gilt für *c*₁ und *c*₂

► Fall: $(\sigma, \text{false}) \in \llbracket b \rrbracket_B$ mit $(\sigma, \sigma') \in \llbracket c_2 \rrbracket_c$

$$\begin{array}{ll} & (\sigma, \text{false}) \in \llbracket b \rrbracket_B \wedge (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c \\ \text{Lemma Bexp} & \implies \langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \wedge (\sigma, \sigma') \in \llbracket c_2 \rrbracket_c \\ \text{IA für } c_2 & \implies \langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \wedge \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \\ \text{Def. } \langle \cdot, \cdot \rangle \rightarrow_{\text{Stmt}} \cdot & \implies \langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \end{array}$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsschritt:

► Fall **while** (b) c

$$\llbracket \mathbf{while} (b) c \rrbracket_c = \text{fix}(\Gamma)$$

$$\begin{aligned} \text{mit } \Gamma(s) = & \{(\sigma, \sigma') \mid (\sigma, \text{true}) \in \llbracket b \rrbracket_{\mathcal{B}} \wedge (\sigma, \sigma') \in \llbracket c \rrbracket_c \circ s\} \\ & \cup \{(\sigma, \sigma) \mid (\sigma, \text{false}) \in \llbracket b \rrbracket_{\mathcal{B}}\} \end{aligned}$$

Induktionsannahme gilt für c

$$(\sigma, \sigma') \in \llbracket \mathbf{while} (b) c \rrbracket_c \implies (\sigma, \sigma') \in \text{fix}(\Gamma)$$

nach Def. $\llbracket . \rrbracket_c$

$$\implies (\sigma, \sigma') \in \bigcup_{i \in \mathbb{N}} \Gamma^i(\emptyset)$$

nach Def. $\text{fix}(\Gamma)$

$$\implies (\sigma, \sigma') \in \Gamma^i(\emptyset) \text{ für ein } i \in \mathbb{N}$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsschritt:

► Fall **while** (b) c

$$\llbracket \mathbf{while} (b) c \rrbracket_c = \text{fix}(\Gamma)$$

$$\text{mit } \Gamma(s) = \{(\sigma, \sigma') \mid (\sigma, \text{true}) \in \llbracket b \rrbracket_{\mathcal{B}} \wedge (\sigma, \sigma') \in \llbracket c \rrbracket_c \circ s\} \\ \cup \{(\sigma, \sigma) \mid (\sigma, \text{false}) \in \llbracket b \rrbracket_{\mathcal{B}}\}$$

Induktionsannahme gilt für c

$$\begin{aligned} (\sigma, \sigma') \in \llbracket \mathbf{while} (b) c \rrbracket_c &\implies (\sigma, \sigma') \in \text{fix}(\Gamma) && \text{nach Def. } \llbracket . \rrbracket_c \\ &\implies (\sigma, \sigma') \in \bigcup_{i \in \mathbb{N}} \Gamma^i(\emptyset) && \text{nach Def. } \text{fix}(\Gamma) \\ &\implies (\sigma, \sigma') \in \Gamma^i(\emptyset) \text{ für ein } i \in \mathbb{N} \end{aligned}$$

Unterbeweis:

$$\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \Rightarrow \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \quad (\text{UB})$$

Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsschritt:

► Fall **while** (b) c

$$\llbracket \mathbf{while} (b) c \rrbracket_c = \mathit{fix}(\Gamma)$$

$$\text{mit } \Gamma(s) = \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in \llbracket b \rrbracket_{\mathcal{B}} \wedge (\sigma, \sigma') \in \llbracket c \rrbracket_c \circ s\} \\ \cup \{(\sigma, \sigma) \mid (\sigma, \mathit{false}) \in \llbracket b \rrbracket_{\mathcal{B}}\}$$

Induktionsannahme gilt für c

$$\begin{aligned} (\sigma, \sigma') \in \llbracket \mathbf{while} (b) c \rrbracket_c &\implies (\sigma, \sigma') \in \mathit{fix}(\Gamma) && \text{nach Def. } \llbracket \cdot \rrbracket_c \\ &\implies (\sigma, \sigma') \in \bigcup_{i \in \mathbb{N}} \Gamma^i(\emptyset) && \text{nach Def. } \mathit{fix}(\Gamma) \\ &\implies (\sigma, \sigma') \in \Gamma^i(\emptyset) \text{ für ein } i \in \mathbb{N} \\ &\implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' && \text{nach (UB)} \end{aligned}$$

Unterbeweis:

$$\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \Rightarrow \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \quad (\text{UB})$$

Unterbeweis: $\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{Stmt} \sigma'$

Es gilt die Induktionsannahme für c :

$$\forall \rho, \rho'. (\rho, \rho') \in \llbracket c \rrbracket_c \implies \langle c, \rho \rangle \rightarrow_{Stmt} \rho' \quad (*)$$

Beweis per Induktion über i :

► Induktionsanfang $i = 0$:

$$(\sigma, \sigma') \in \Gamma^0(\emptyset) \implies (\sigma, \sigma') \in \emptyset \implies \text{false}$$

Implikation trivialerweise erfüllt da $\text{false} \implies P$ immer wahr

Unterbeweis: $\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \implies \langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmnt} \sigma'$

Es gilt die Induktionsannahme für c :

$$\forall \rho, \rho'. (\rho, \rho') \in \llbracket c \rrbracket_C \implies \langle c, \rho \rangle \rightarrow_{Stmnt} \rho' \quad (*)$$

Beweis per Induktion über i :

- ▶ Induktionsschritt $i \rightarrow i + 1$:
- ▶ Induktionsannahme (UB) gilt für i

$$(\sigma, \sigma') \in \Gamma^{i+1}(\emptyset) \implies (\sigma, \sigma') \in \Gamma(\Gamma^i(\emptyset))$$

$$\stackrel{\text{Def. } \Gamma}{\implies} (\sigma, \sigma') \in \{(\sigma, \sigma'') \mid (\sigma, \text{true}) \in \llbracket b \rrbracket_B, (\sigma, \sigma') \in \llbracket c \rrbracket_C, (\sigma', \sigma'') \in \Gamma^i(\emptyset)\} \\ \cup \{(\sigma, \sigma) \mid (\sigma, \text{false}) \in \llbracket b \rrbracket_B\}$$

- ▶ Fallunterscheidung über Zugehörigkeit zur Teilmenge

Unterbeweis: $\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{Stmt} \sigma'$

Es gilt die Induktionsannahme für c :

$$\forall \rho, \rho'. (\rho, \rho') \in \llbracket c \rrbracket_c \implies \langle c, \rho \rangle \rightarrow_{Stmt} \rho' \quad (*)$$

Beweis per Induktion über i :

- ▶ Induktionsschritt $i \rightarrow i + 1$:
- ▶ Induktionsannahme (UB) gilt für i
- ▶ Fall $(\sigma, true) \in \llbracket b \rrbracket_B$ mit $(\sigma, \sigma') \in \llbracket c \rrbracket_c, (\sigma', \sigma'') \in \Gamma^i(\emptyset)$

$$\begin{aligned} (\sigma, \sigma'') \in \Gamma(\Gamma^i(\emptyset)) &\implies \underbrace{(\sigma, true) \in \llbracket b \rrbracket_B}_{\text{Lemma Bexp}} \wedge \underbrace{(\sigma, \sigma') \in \llbracket c \rrbracket_c}_{\text{IA } (*)} \wedge \underbrace{(\sigma', \sigma'') \in \Gamma^i(\emptyset)}_{\text{IA (UB) für } i} \\ &\implies \langle b, \sigma \rangle \rightarrow_{Bexp} true \wedge \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \wedge \langle \mathbf{while} (b) c, \sigma' \rangle \rightarrow_{Stmt} \sigma'' \\ &\implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{Stmt} \sigma'' \end{aligned}$$

Unterbeweis: $\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{Stmnt} \sigma'$

Es gilt die Induktionsannahme für c :

$$\forall \rho, \rho'. (\rho, \rho') \in \llbracket c \rrbracket_c \implies \langle c, \rho \rangle \rightarrow_{Stmnt} \rho' \quad (*)$$

Beweis per Induktion über i :

- ▶ Induktionsschritt $i \rightarrow i + 1$:
- ▶ Induktionsannahme (UB) gilt für i
- ▶ Fall $(\sigma, false) \in \llbracket b \rrbracket_B$

$$\begin{aligned} (\sigma, \sigma') \in \Gamma(\Gamma^i(\emptyset)) &\implies (\sigma, false) \in \llbracket b \rrbracket_B \wedge \sigma' = \sigma \\ &\implies \langle b, \sigma \rangle \rightarrow_{Bexp} false \wedge \sigma' = \sigma \\ &\implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{Stmnt} \sigma' \end{aligned}$$

Lemma für **Bexp**



Beweis: $\forall c \in \mathbf{Stmt}. \forall \sigma, \sigma'. (\sigma, \sigma') \in \llbracket c \rrbracket_c \implies \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'$

Induktionsschritt:

► Fall **while** (b) c

$$\llbracket \mathbf{while} (b) c \rrbracket_c = \mathit{fix}(\Gamma)$$

$$\begin{aligned} \text{mit } \Gamma(s) = & \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in \llbracket b \rrbracket_{\mathcal{B}} \wedge (\sigma, \sigma') \in \llbracket c \rrbracket_c \circ s\} \\ & \cup \{(\sigma, \sigma) \mid (\sigma, \mathit{false}) \in \llbracket b \rrbracket_{\mathcal{B}}\} \end{aligned}$$

Induktionsannahme gilt für c

$$\begin{aligned} (\sigma, \sigma') \in \llbracket \mathbf{while} (b) c \rrbracket_c &\implies (\sigma, \sigma') \in \mathit{fix}(\Gamma) && \text{nach Def. } \llbracket . \rrbracket_c \\ &\implies (\sigma, \sigma') \in \bigcup_{i \in \mathbb{N}} \Gamma^i(\emptyset) && \text{nach Def. } \mathit{fix}(\Gamma) \\ &\implies (\sigma, \sigma') \in \Gamma^i(\emptyset) \text{ für ein } i \in \mathbb{N} \\ &\implies \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' && \text{nach (UB)} \end{aligned}$$

Unterbeweis:

$$\forall i \in \mathbb{N}. (\sigma, \sigma') \in \Gamma^i(\emptyset) \Rightarrow \langle \mathbf{while} (b) c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \quad (\text{UB})$$

Zusammenfassung: Äquivalenz der Semantiken

- Wir haben gezeigt: für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ'

$$\langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma' \iff (\sigma, \sigma') \in \llbracket c \rrbracket_c$$

- Das ist äquivalent zu (für alle $c \in \mathbf{Stmt}$, für alle Zustände σ, σ'):

$$\llbracket c \rrbracket_c = \{(\sigma, \sigma') \mid \langle c, \sigma \rangle \rightarrow_{\mathbf{Stmt}} \sigma'\}$$

- Insbesondere ist die Undefiniertheit gleich:
wenn es keine Ableitung für c, σ gibt, dann ist auch $\sigma \notin \llbracket c \rrbracket_c$.

Fahrplan

- ▶ Einführung
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Der Floyd-Hoare-Kalkül I
- ▶ Der Floyd-Hoare-Kalkül II: Invarianten
- ▶ Korrektheit des Floyd-Hoare-Kalküls
- ▶ Strukturierte Datentypen
- ▶ Verifikationsbedingungen
- ▶ Vorwärts mit Floyd und Hoare
- ▶ Funktionen und Prozeduren I
- ▶ Funktionen und Prozeduren II
- ▶ Referenzen und Speichermodelle
- ▶ Ausblick und Rückblick