

6.4 Bezug zur operationalen Semantik

Für `While` haben wir nun zwei operationale Semantiken und eine denotationale. Von den operationalen ist bekannt, dass sie das gleiche terminierende Verhalten definieren (Kor. 10), für unendliche Ausführungen haben wir das in der Übung untersucht. Nun stellt sich natürlich die Frage, in welchem Verhältnis die denotationale Semantik zu diesen beiden steht.

Lemma 32. Wenn $\langle c, \sigma \rangle \Downarrow \sigma'$, dann $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma'$.

Beweis. Induktion über $\langle c, \sigma \rangle \Downarrow \sigma'$ (vgl. Def. 4)

- Fall `SKIPBS`: Zu zeigen: $\mathcal{D} \llbracket \text{skip} \rrbracket \sigma = \sigma$. Nach Definition.
- Fall `ASSBS`: Zu zeigen: $\mathcal{D} \llbracket x := a \rrbracket \sigma = \sigma[x \mapsto \mathcal{A} \llbracket a \rrbracket \sigma]$. Nach Definition.
- Fall `SEQBS`: Induktionsannahmen: $\mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$ und $\mathcal{D} \llbracket c_2 \rrbracket \sigma' = \sigma''$.
Zu zeigen: $\mathcal{D} \llbracket c_1 ; c_2 \rrbracket \sigma = \sigma''$
Nach Definition gilt $\mathcal{D} \llbracket c_1 ; c_2 \rrbracket \sigma = (\mathcal{D} \llbracket c_2 \rrbracket \circ \mathcal{D} \llbracket c_1 \rrbracket)(\sigma)$.
Mit den Induktionsannahmen folgt $(\mathcal{D} \llbracket c_2 \rrbracket \circ \mathcal{D} \llbracket c_1 \rrbracket)(\sigma) = \sigma''$.
- Fall `IFTTBS`: Induktionsannahmen: $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{tt}$ und $\mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$.
Zu zeigen: $\mathcal{D} \llbracket \text{if } (b) \text{ then } c_1 \text{ else } c_2 \rrbracket \sigma = \sigma'$.

$$\mathcal{D} \llbracket \text{if } (b) \text{ then } c_1 \text{ else } c_2 \rrbracket \sigma = \text{IF}(\mathcal{B} \llbracket b \rrbracket, \mathcal{D} \llbracket c_1 \rrbracket, \mathcal{D} \llbracket c_2 \rrbracket)(\sigma) = \mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$$

- Fall `IFFFBS`: Analog zu `IFTTBS`.
- Fall `WHILETTBS`:
Induktionsannahmen: $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{tt}$, $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma'$ und $\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma' = \sigma''$.
Zu zeigen: $\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma = \sigma''$.

$$\begin{aligned} \mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma &= \text{FIX}(\lambda f. \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id}))(\sigma) \\ (\text{Fixpunkteigenschaft}) &= \text{IF}(\mathcal{B} \llbracket b \rrbracket, \text{FIX}(\lambda f. \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id})) \circ \mathcal{D} \llbracket c \rrbracket, \text{id})(\sigma) \\ &= (\text{FIX}(\lambda f. \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id})) \circ \mathcal{D} \llbracket c \rrbracket)(\sigma) \\ &= (\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \circ \mathcal{D} \llbracket c \rrbracket)(\sigma) = \sigma'' \end{aligned}$$

- Fall `WHILEFFBS`: Induktionsannahme: $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{ff}$. Zu zeigen: $\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma = \sigma$.

$$\begin{aligned} \mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma &= \text{FIX}(\lambda f. \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id}))(\sigma) \\ &= \text{IF}(\mathcal{B} \llbracket b \rrbracket, \text{FIX}(\lambda f. \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id})) \circ \mathcal{D} \llbracket c \rrbracket, \text{id})(\sigma) = \text{id}(\sigma) = \sigma \square \end{aligned}$$

Lemma 33. Wenn $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma'$, dann $\langle c, \sigma \rangle \Downarrow \sigma'$.

Beweis. Induktion über c (σ, σ' beliebig):

- Fall $c = \text{skip}$: Zu zeigen: Wenn $\mathcal{D} \llbracket \text{skip} \rrbracket \sigma = \sigma'$, dann $\langle \text{skip}, \sigma \rangle \Downarrow \sigma'$.
Aus der Voraussetzung folgt $\sigma' = \text{id}(\sigma) = \sigma$, damit die Behauptung nach Regel `SKIPBS`.
- Fall $c = x := a$: Zu zeigen: Wenn $\mathcal{D} \llbracket x := a \rrbracket \sigma = \sigma'$, dann $\langle x := a, \sigma \rangle \Downarrow \sigma'$. Aus der Voraussetzung folgt $\sigma' = \sigma[x \mapsto \mathcal{A} \llbracket a \rrbracket \sigma]$. `ASSBS` liefert dann die Behauptung.
- Fall $c = c_1 ; c_2$:
Induktionsannahmen (für beliebige σ, σ'):
Wenn $\mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$, dann $\langle c_1, \sigma \rangle \Downarrow \sigma'$. Wenn $\mathcal{D} \llbracket c_2 \rrbracket \sigma' = \sigma''$, dann $\langle c_2, \sigma' \rangle \Downarrow \sigma''$.
Zu zeigen: Wenn $\mathcal{D} \llbracket c_1 ; c_2 \rrbracket \sigma = \sigma''$, dann $\langle c_1 ; c_2, \sigma \rangle \Downarrow \sigma''$.
Wegen $\mathcal{D} \llbracket c_1 ; c_2 \rrbracket \sigma = (\mathcal{D} \llbracket c_2 \rrbracket \circ \mathcal{D} \llbracket c_1 \rrbracket)(\sigma) = \sigma''$ gibt es ein σ^* mit $\mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma^*$ und $\mathcal{D} \llbracket c_2 \rrbracket \sigma^* = \sigma''$.
Mit den Induktionsannahmen gilt also $\langle c_1, \sigma \rangle \Downarrow \sigma^*$ und $\langle c_2, \sigma^* \rangle \Downarrow \sigma''$. Damit folgt die Behauptung mit Regel `SEQBS`.

- Fall $c = \text{if } (b) \text{ then } c_1 \text{ else } c_2$:
 Induktionsannahmen: Wenn $\mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$, dann $\langle c_1, \sigma \rangle \Downarrow \sigma'$. Wenn $\mathcal{D} \llbracket c_2 \rrbracket \sigma = \sigma'$, dann $\langle c_2, \sigma \rangle \Downarrow \sigma'$.
 Zu zeigen: Wenn $\mathcal{D} \llbracket \text{if } (b) \text{ then } c_1 \text{ else } c_2 \rrbracket \sigma = \sigma'$, dann $\langle \text{if } (b) \text{ then } c_1 \text{ else } c_2, \sigma \rangle \Downarrow \sigma'$.

Fallunterscheidung nach $\mathcal{B} \llbracket b \rrbracket \sigma$:

- Fall $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{tt}$: Dann gilt:

$$\mathcal{D} \llbracket \text{if } (b) \text{ then } c_1 \text{ else } c_2 \rrbracket \sigma = \text{IF}(\mathcal{B} \llbracket b \rrbracket, \mathcal{D} \llbracket c_1 \rrbracket, \mathcal{D} \llbracket c_2 \rrbracket)(\sigma) = \mathcal{D} \llbracket c_1 \rrbracket \sigma = \sigma'$$

Mit der Induktionsannahme folgt $\langle c_1, \sigma \rangle \Downarrow \sigma'$ und daraus zusammen mit $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{tt}$ die Behauptung nach Regel IFTT_{BS} .

- Fall $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{ff}$: Analog.

- Fall $c = \text{while } (b) \text{ do } c$:

Induktionsannahme I (für beliebige σ, σ'): Wenn $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma'$, dann $\langle c, \sigma \rangle \Downarrow \sigma'$.

Zu zeigen: Wenn $\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma = \sigma'$, dann $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$.

Sei $F(f) = \text{IF}(\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket c \rrbracket, \text{id})$ das Funktional für die Schleife. Dann gilt nach Thm. 30:

$$\mathcal{D} \llbracket \text{while } (b) \text{ do } c \rrbracket \sigma = \text{FIX}(F)(\sigma) = \left(\bigsqcup \{ F^n(\perp) \mid n \in \mathbb{N} \} \right) (\sigma) = \sigma'$$

Nach Definition von $\bigsqcup \{ F^n(\perp) \mid n \in \mathbb{N} \}$ gibt es ein $n \in \mathbb{N}$ mit $F^n(\perp)(\sigma) = \sigma'$.

Behauptung: Wenn $F^n(\perp)(\sigma) = \sigma'$, dann $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$.

Beweis durch Induktion über n (σ, σ' beliebig):

- Basisfall $n = 0$: Zu zeigen: Wenn $F^0(\perp)(\sigma) = \sigma'$, dann $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$.

Trivial, da Voraussetzung nicht erfüllt: $F^0(\perp)(\sigma) = \perp(\sigma) = \perp$.

- Induktionsschritt $n + 1$:

Induktionsannahme II (σ, σ' beliebig): Wenn $F^n(\perp)(\sigma) = \sigma'$, dann $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$.

Zu zeigen: Wenn $F^{n+1}(\perp)(\sigma) = \sigma'$, dann $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$.

Beweis von $\langle \text{while } (b) \text{ do } c, \sigma \rangle \Downarrow \sigma'$ durch Fallunterscheidung nach $\mathcal{B} \llbracket b \rrbracket \sigma$:

- Fall $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{tt}$: Dann gilt:

$$F^{n+1}(\perp)(\sigma) = F(F^n(\perp))(\sigma) = \text{IF}(\mathcal{B} \llbracket b \rrbracket, F^n(\perp) \circ \mathcal{D} \llbracket c \rrbracket, \text{id})(\sigma) = (F^n(\perp) \circ \mathcal{D} \llbracket c \rrbracket)(\sigma) = \sigma'$$

Damit gibt es ein σ^* mit $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma^*$ und $F^n(\perp)(\sigma^*) = \sigma'$. Aus $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma^*$ folgt nach Induktionsannahme I, dass $\langle c, \sigma \rangle \Downarrow \sigma^*$. Aus $F^n(\perp)(\sigma^*) = \sigma'$ folgt nach Induktionsannahme II, dass $\langle \text{while } (b) \text{ do } c, \sigma^* \rangle \Downarrow \sigma'$. Zusammen folgt die Behauptung nach Regel $\text{WHILETT}_{\text{BS}}$.

- Fall $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{ff}$: Dann gilt:

$$F^{n+1}(\perp)(\sigma) = F(F^n(\perp))(\sigma) = \text{IF}(\mathcal{B} \llbracket b \rrbracket, F^n(\perp) \circ \mathcal{D} \llbracket c \rrbracket, \text{id})(\sigma) = \text{id}(\sigma) = \sigma$$

Also $\sigma' = \sigma$. Mit $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{ff}$ folgt die Behauptung nach Regel $\text{WHILEFF}_{\text{BS}}$. \square

Theorem 34 (Adäquatheit, Äquivalenz von operationaler und denotationaler Semantik).

Für alle c, σ und σ' gilt $\langle c, \sigma \rangle \Downarrow \sigma'$ gdw. $\langle c, \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle$ gdw. $\mathcal{D} \llbracket c \rrbracket \sigma = \sigma'$.

Was hat man nun von der denotationalen Semantik? Viele Aussagen über die operationale Semantik sind (dank des Adäquatheitstheorems) in der denotationalen Semantik viel einfacher zu beweisen. Man braucht für vieles nicht mehr Induktionen über Ableitungsbäume zu führen! Beispielsweise werden die Determinismusbeweise (Thm. 2 und Kor. 5) hinfällig, da dies direkt aus dem Funktionscharakter von $\mathcal{D} \llbracket - \rrbracket$ folgt. Auch das Schleifenabwicklungslemma 1 wird trivial.

Aus Kompositionalität und Adäquatheit kann man noch mehr Nutzen ziehen: Zwei Programme c_1 und c_2 mit der gleichen denotationalen Semantik ($\mathcal{D} \llbracket c_1 \rrbracket = \mathcal{D} \llbracket c_2 \rrbracket$) können jederzeit gegeneinander ausgetauscht werden – sogar beliebig innerhalb anderer Sprachkonstrukte.

Definition 40 (Kontext). Ein Kontext ist ein While-Programm mit einem Loch \square . Formal ist ein Kontext (Variablenkonvention K) durch folgende Grammatik gegeben:

$$\begin{aligned} \text{Context } K ::= & \square \mid K; c \mid c; K \mid \text{if } (b) \text{ then } K \text{ else } c \mid \\ & \text{if } (b) \text{ then } c \text{ else } K \mid \text{while } (b) \text{ do } K \end{aligned}$$

Die Kontextfüll-Funktion $[-]$ ersetzt das Loch \square im Kontext K durch das übergebene Programm c' :

$$\begin{aligned} \square [c'] &= c' \\ (K; c) [c'] &= K [c']; c \\ (c; K) [c'] &= c; (K [c']) \\ (\text{if } (b) \text{ then } K \text{ else } c) [c'] &= \text{if } (b) \text{ then } K [c'] \text{ else } c \\ (\text{if } (b) \text{ then } c \text{ else } K) [c'] &= \text{if } (b) \text{ then } c \text{ else } (K [c']) \\ (\text{while } (b) \text{ do } K) [c'] &= \text{while } (b) \text{ do } (K [c']) \end{aligned}$$

Definition 41 (Semantik eines Kontexts). Die Semantik $\mathcal{K} \llbracket K \rrbracket$ eines Kontexts K ist vom Typ $(\Sigma \rightarrow \Sigma) \Rightarrow (\Sigma \rightarrow \Sigma)$ und rekursiv über K definiert:

$$\begin{aligned} \mathcal{K} \llbracket \square \rrbracket f &= f \\ \mathcal{K} \llbracket K; c \rrbracket f &= \mathcal{D} \llbracket c \rrbracket \circ \mathcal{K} \llbracket K \rrbracket f \\ \mathcal{K} \llbracket c; K \rrbracket f &= \mathcal{K} \llbracket K \rrbracket f \circ \mathcal{D} \llbracket c \rrbracket \\ \mathcal{K} \llbracket \text{if } (b) \text{ then } K \text{ else } c \rrbracket f &= \text{IF} (\mathcal{B} \llbracket b \rrbracket, \mathcal{K} \llbracket K \rrbracket f, \mathcal{D} \llbracket c \rrbracket) \\ \mathcal{K} \llbracket \text{if } (b) \text{ then } c \text{ else } K \rrbracket f &= \text{IF} (\mathcal{B} \llbracket b \rrbracket, \mathcal{D} \llbracket c \rrbracket, \mathcal{K} \llbracket K \rrbracket f) \\ \mathcal{K} \llbracket \text{while } (b) \text{ do } K \rrbracket f &= \text{FIX} (\lambda g. \text{IF} (\mathcal{B} \llbracket b \rrbracket, g \circ \mathcal{K} \llbracket K \rrbracket f, id)) \end{aligned}$$

Theorem 35 (Kompositionalität).

Für alle Kontexte K und Programme c gilt $\mathcal{D} \llbracket K [c] \rrbracket = \mathcal{K} \llbracket K \rrbracket (\mathcal{D} \llbracket c \rrbracket)$

Beweis. Induktion über K und ausrechnen. Beispielhaft der Fall für **while** (b) **do** K :

Induktionsannahmen: $\mathcal{D} \llbracket K [c] \rrbracket = \mathcal{K} \llbracket K \rrbracket (\mathcal{D} \llbracket c \rrbracket)$ für alle c .

Zu zeigen: $\mathcal{D} \llbracket (\text{while } (b) \text{ do } K) [c] \rrbracket = \mathcal{K} \llbracket \text{while } (b) \text{ do } K \rrbracket (\mathcal{D} \llbracket c \rrbracket)$

$$\begin{aligned} \mathcal{D} \llbracket (\text{while } (b) \text{ do } K) [c] \rrbracket &= \mathcal{D} \llbracket \text{while } (b) \text{ do } (K [c]) \rrbracket = \text{FIX} (\lambda f. \text{IF} (\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{D} \llbracket K [c] \rrbracket, id)) \\ &= \text{FIX} (\lambda f. \text{IF} (\mathcal{B} \llbracket b \rrbracket, f \circ \mathcal{K} \llbracket K \rrbracket (\mathcal{D} \llbracket c \rrbracket), id)) \\ &= \mathcal{K} \llbracket \text{while } (b) \text{ do } K \rrbracket (\mathcal{D} \llbracket c \rrbracket) \quad \square \end{aligned}$$

Korollar 36. Zwei Programme c_1 und c_2 mit gleicher Semantik sind in allen Kontexten austauschbar: Wenn $\mathcal{D} \llbracket c_1 \rrbracket = \mathcal{D} \llbracket c_2 \rrbracket$, dann $\mathcal{D} \llbracket K [c_1] \rrbracket = \mathcal{D} \llbracket K [c_2] \rrbracket$.

Beweis. Nach Thm. 35 gilt $\mathcal{D} \llbracket K [c_1] \rrbracket = \mathcal{K} \llbracket K \rrbracket (\mathcal{D} \llbracket c_1 \rrbracket) = \mathcal{K} \llbracket K \rrbracket (\mathcal{D} \llbracket c_2 \rrbracket) = \mathcal{D} \llbracket K [c_2] \rrbracket$. □

Beispiel 35. $c_1 = \text{while } (b) \text{ do } c$ und $c_2 = \text{if } (b) \text{ then } c; \text{while } (b) \text{ do } c \text{ else skip}$ haben die gleiche denotationale Semantik. Damit kann ein Compiler in allen Programmen c_2 durch c_1 ersetzen (oder umgekehrt), ohne das Verhalten zu ändern.

Übung: Beweisen Sie ohne Verwendung der denotationalen Semantik direkt, dass man jederzeit Schleifen abwickeln darf.