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 35. Wenn $\langle c,\ \sigma\rangle \Downarrow \sigma'$, dann $\mathcal{D}\,[\![c]\!]\,\sigma = \sigma'$.

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

- Fall SKIP_{BS}: Zu zeigen: $\mathcal{D}\,[\![\texttt{skip}]\!]\,\sigma = \sigma$. Nach Definition.
- Fall ASS_{BS}: Zu zeigen: $\mathcal{D}\,[\![x\ \texttt{:=}\ a]\!]\,\sigma = \sigma[x \mapsto \mathcal{A}\,[\![a]\!]\,\sigma]$. Nach Definition.
- Fall SEQ_{BS}: Induktionsannahmen: $\mathcal{D}\,[\![c_1]\!]\,\sigma = \sigma'$ und $\mathcal{D}\,[\![c_2]\!]\,\sigma' = \sigma''$.
  Zu zeigen: $\mathcal{D}\,[\![c_1\,\texttt{;}\ c_2]\!]\,\sigma = \sigma''$

  Nach Defintion gilt $\mathcal{D}\,[\![c_1\,\texttt{;}\ c_2]\!]\,\sigma = (\mathcal{D}\,[\![c_2]\!] \circ \mathcal{D}\,[\![c_1]\!])(\sigma)$.
  Mit den Induktionsannahmen folgt $(\mathcal{D}\,[\![c_2]\!] \circ \mathcal{D}\,[\![c_1]\!])(\sigma) = \sigma''$.
- Fall IFTT_{BS}: Induktionsannahmen: $\mathcal{B}\,[\![b]\!]\,\sigma = \mathbf{tt}$ und $\mathcal{D}\,[\![c_1]\!]\,\sigma = \sigma'$.
  Zu zeigen: $\mathcal{D}\,[\![\texttt{if (}b\texttt{) then}\ c_1\ \texttt{else}\ c_2]\!]\,\sigma = \sigma'$.

$$\mathcal{D}\,[\![\texttt{if (}b\texttt{) then}\ c_1\ \texttt{else}\ c_2]\!]\,\sigma = \text{IF}\,(\mathcal{B}\,[\![b]\!],\ \mathcal{D}\,[\![c_1]\!],\ \mathcal{D}\,[\![c_2]\!])\,(\sigma) = \mathcal{D}\,[\![c_1]\!]\,\sigma = \sigma'$$

- Fall IFFF_{BS}: Analog zu IFTT_{BS}.
- Fall $\text{WHILETT}_{\text{BS}}$:
  Induktionsannahmen: $\mathcal{B}\,[\![b]\!]\,\sigma = \mathbf{tt}$, $\mathcal{D}\,[\![c]\!]\,\sigma = \sigma'$ und $\mathcal{D}\,[\![\texttt{while (}b\texttt{) do}\ c]\!]\,\sigma' = \sigma''$.
  Zu zeigen: $\mathcal{D}\,[\![\texttt{while (}b\texttt{) do}\ c]\!]\,\sigma = \sigma''$.

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

- Fall $\text{WHILEFF}_{\text{BS}}$: Induktionsannahme: $\mathcal{B}\,[\![b]\!]\,\sigma = \mathbf{ff}$. Zu zeigen: $\mathcal{D}\,[\![\texttt{while (}b\texttt{) do}\ c]\!]\,\sigma = \sigma$.

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

Lemma 36. Wenn $\mathcal{D}\,[\![c]\!]\,\sigma = \sigma'$, dann $\langle c,\ \sigma\rangle \Downarrow \sigma'$.

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

- Fall $c = \texttt{skip}$: Zu zeigen: Wenn $\mathcal{D}\,[\![\texttt{skip}]\!]\,\sigma = \sigma'$, dann $\langle \texttt{skip},\ \sigma\rangle \Downarrow \sigma'$.

  Aus der Voraussetzung folgt $\sigma' = id(\sigma) = \sigma$, damit die Behauptung nach Regel SKIP_{BS}.
- Fall $c = x\ \texttt{:=}\ a$: Zu zeigen: Wenn $\mathcal{D}\,[\![x\ \texttt{:=}\ a]\!]\,\sigma = \sigma'$, dann $\langle x\ \texttt{:=}\ a,\ \sigma\rangle \Downarrow \sigma'$. Aus der Voraussetzung folgt $\sigma' = \sigma[x \mapsto \mathcal{A}\,[\![a]\!]\,\sigma]$. ASS_{BS} liefert dann die Behauptung.
- Fall $c = c_1\,\texttt{;}\ c_2$:
  Induktionsannahmen (für beliebige σ, σ'):
  Wenn $\mathcal{D}\,[\![c_1]\!]\,\sigma = \sigma'$, dann $\langle c_1,\ \sigma\rangle \Downarrow \sigma'$. Wenn $\mathcal{D}\,[\![c_2]\!]\,\sigma = \sigma'$, dann $\langle c_2,\ \sigma\rangle \Downarrow \sigma'$.

  Zu zeigen: Wenn $\mathcal{D}\,[\![c_1\,\texttt{;}\ c_2]\!]\,\sigma = \sigma'$, dann $\langle c_1\,\texttt{;}\ c_2,\ \sigma\rangle \Downarrow \sigma'$.

  Wegen $\mathcal{D}\,[\![c_1\,\texttt{;}\ c_2]\!]\,\sigma = (\mathcal{D}\,[\![c_2]\!] \circ \mathcal{D}\,[\![c_1]\!])(\sigma) = \sigma'$ gibt es ein σ^* mit $\mathcal{D}\,[\![c_1]\!]\,\sigma = \sigma^*$ und $\mathcal{D}\,[\![c_2]\!]\,\sigma^* = \sigma'$.
  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 SEQ_{BS}.

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

  Fallunterscheidung nach $\mathcal{B}[\![b]\!]\sigma$:
  - Fall $\mathcal{B}[\![b]\!]\sigma = \mathbf{tt}$: Dann gilt:

    $$\mathcal{D}[\![\texttt{if (}b\texttt{) then } c_1 \texttt{ else } c_2]\!]\sigma = \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ \mathcal{D}[\![c_1]\!],\ \mathcal{D}[\![c_2]\!])\,(\sigma) = \mathcal{D}[\![c_1]\!]\sigma = \sigma'$$

    Mit der Induktionsannahme folgt $\langle c_1,\ \sigma\rangle \Downarrow \sigma'$ und daraus zusammen mit $\mathcal{B}[\![b]\!]\sigma = \mathbf{tt}$ die Behauptung nach Regel $\textsc{IfTT}_{\mathrm{BS}}$.
  - Fall $\mathcal{B}[\![b]\!]\sigma = \mathbf{ff}$: Analog.
- Fall $c = \texttt{while (}b\texttt{) do } c$:
  Induktionsannahme I (für beliebige σ, σ'): Wenn $\mathcal{D}[\![c]\!]\sigma = \sigma'$, dann $\langle c,\ \sigma\rangle \Downarrow \sigma'$.
  Zu zeigen: Wenn $\mathcal{D}[\![\texttt{while (}b\texttt{) do } c]\!]\sigma = \sigma'$, dann $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$.

  Sei $F(f) = \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ f \circ \mathcal{D}[\![c]\!],\ id)$ das Funktional für die Schleife. Dann gilt nach Thm. 33:

  $$\mathcal{D}[\![\texttt{while (}b\texttt{) do } c]\!]\sigma = \mathrm{FIX}\,(F)\,(\sigma) = \left(\bigsqcup \{\, F^n(\bot) \mid n \in \mathbb{N}\,\}\right)(\sigma) = \sigma'$$

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

  Behauptung: Wenn $F^n(\bot)(\sigma) = \sigma'$, dann $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$.
  Beweis duch Induktion über n (σ, σ' beliebig):
  - Basisfall $n = 0$: Zu zeigen: Wenn $F^0(\bot)(\sigma) = \sigma'$, dann $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$.
    Trivial, da Voraussetzung nicht erfüllt: $F^0(\bot)(\sigma) = \bot(\sigma) = \bot$.
  - Induktionsschritt $n + 1$:
    Induktionsannahme II (σ, σ' beliebig): Wenn $F^n(\bot)(\sigma) = \sigma'$, dann $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$.
    Zu zeigen: Wenn $F^{n+1}(\bot)(\sigma) = \sigma'$, dann $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$.
    Beweis von $\langle \texttt{while (}b\texttt{) do } c,\ \sigma\rangle \Downarrow \sigma'$ durch Fallunterscheidung nach $\mathcal{B}[\![b]\!]\sigma$:
    - Fall $\mathcal{B}[\![b]\!]\sigma = \mathbf{tt}$: Dann gilt:

      $$F^{n+1}(\bot)(\sigma) = F(F^n(\bot))(\sigma) = \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ F^n(\bot) \circ \mathcal{D}[\![c]\!],\ id)\,(\sigma) = (F^n(\bot) \circ \mathcal{D}[\![c]\!])(\sigma) = \sigma'$$

      Damit gibt es ein σ^* mit $\mathcal{D}[\![c]\!]\sigma = \sigma^*$ und $F^n(\bot)(\sigma^*) = \sigma'$. Aus $\mathcal{D}[\![c]\!]\sigma = \sigma^*$ folgt nach Induktionsannahme I, dass $\langle c,\ \sigma\rangle \Downarrow \sigma^*$. Aus $F^n(\bot)(\sigma^*) = \sigma'$ folgt nach Induktionsannahme II, dass $\langle \texttt{while (}b\texttt{) do } c,\ \sigma^*\rangle \Downarrow \sigma'$. Zusammen folgt die Behauptung nach Regel $\textsc{WhileTT}_{\mathrm{BS}}$.
    - Fall $\mathcal{B}[\![b]\!]\sigma = \mathbf{ff}$: Dann gilt:

      $$F^{n+1}(\bot)(\sigma) = F(F^n(\bot))(\sigma) = \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ F^n(\bot) \circ \mathcal{D}[\![c]\!],\ id)\,(\sigma) = id(\sigma) = \sigma$$

      Also $\sigma' = \sigma$. Mit $\mathcal{B}[\![b]\!]\sigma = \mathbf{ff}$ folgt die Behauptung nach Regel $\textsc{WhileFF}_{\mathrm{BS}}$. □

Theorem 37 (Adäquatheit, Äquivalenz von operationaler und denotationaler Semantik). Für alle c, σ und σ' gilt $\langle c,\ \sigma\rangle \Downarrow \sigma'$ gdw. $\langle c,\ \sigma\rangle \xrightarrow{*}_1 \langle \texttt{skip},\ \sigma'\rangle$ gdw. $\mathcal{D}[\![c]\!]\sigma = \sigma'$.

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 $\mathcal{D}[\![\_]\!]$ 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 ($\mathcal{D}[\![c_1]\!] = \mathcal{D}[\![c_2]\!]$) können jederzeit gegeneinander ausgetauscht werden – sogar beliebig innerhalb anderer Sprachkonstrukte.

Definition 41 (Kontext). Ein Kontext ist ein While-Programm mit einem Loch []. Formal ist ein Kontext (Variablenkonvention K) durch folgende Grammatik gegeben:

$$\begin{array}{lrcl} \mathsf{Context} & K & ::= & [] \mid K\texttt{;}\ c \mid c\texttt{;}\ K \mid \texttt{if (}b\texttt{) then }K\texttt{ else }c \mid \\ & & & \texttt{if (}b\texttt{) then }c\texttt{ else }K \mid \texttt{while (}b\texttt{) do }K \end{array}$$

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

$$\begin{array}{rcl} [][c'] & = & c' \\ (K\texttt{;}\ c)[c'] & = & K[c']\texttt{;}\ c \\ (c\texttt{;}\ K)[c'] & = & c\texttt{;}\ \texttt{(}K[c']\texttt{)} \\ (\texttt{if (}b\texttt{) then }K\texttt{ else }c)[c'] & = & \texttt{if (}b\texttt{) then }K[c']\texttt{ else }c \\ (\texttt{if (}b\texttt{) then }c\texttt{ else }K)[c'] & = & \texttt{if (}b\texttt{) then }c\texttt{ else (}K[c']\texttt{)} \\ (\texttt{while (}b\texttt{) do }K)[c'] & = & \texttt{while (}b\texttt{) do (}K[c']\texttt{)} \end{array}$$

Definition 42 (Semantik eines Kontexts). Die Semantik $\mathcal{K}[\![K]\!]$ eines Kontexts K ist vom Typ $(\Sigma \rightharpoonup \Sigma) \Rightarrow (\Sigma \rightharpoonup \Sigma)$ und rekursiv über K definiert:

$$\begin{array}{rcl} \mathcal{K}[\![[]]\!]\, f & = & f \\ \mathcal{K}[\![K\texttt{;}\ c]\!]\, f & = & \mathcal{D}[\![c]\!] \circ \mathcal{K}[\![K]\!]\, f \\ \mathcal{K}[\![c\texttt{;}\ K]\!]\, f & = & \mathcal{K}[\![K]\!]\, f \circ \mathcal{D}[\![c]\!] \\ \mathcal{K}[\![\texttt{if (}b\texttt{) then }K\texttt{ else }c]\!]\, f & = & \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ \mathcal{K}[\![K]\!]\, f,\ \mathcal{D}[\![c]\!]) \\ \mathcal{K}[\![\texttt{if (}b\texttt{) then }c\texttt{ else }K]\!]\, f & = & \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ \mathcal{D}[\![c]\!],\ \mathcal{K}[\![K]\!]\, f) \\ \mathcal{K}[\![\texttt{while (}b\texttt{) do }K]\!]\, f & = & \mathrm{FIX}\,(\lambda g.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ g \circ \mathcal{K}[\![K]\!]\, f,\ id)) \end{array}$$

Theorem 38 (Kompositionalität).
Für alle Kontexte K und Programme c gilt $\mathcal{D}[\![K[c]]\!] = \mathcal{K}[\![K]\!](\mathcal{D}[\![c]\!])$

Beweis. Induktion über K und ausrechnen. Beispielhaft der Fall für `while (`b`) do `K:
Induktionsannahmen: $\mathcal{D}[\![K[c]]\!] = \mathcal{K}[\![K]\!](\mathcal{D}[\![c]\!])$ für alle c.
Zu zeigen: $\mathcal{D}[\![(\texttt{while (}b\texttt{) do }K)[c]]\!] = \mathcal{K}[\![\texttt{while (}b\texttt{) do }K]\!](\mathcal{D}[\![c]\!])$

$$\begin{array}{rcl} \mathcal{D}[\![(\texttt{while (}b\texttt{) do }K)[c]]\!] = \mathcal{D}[\![\texttt{while (}b\texttt{) do (}K[c]\texttt{)}]\!] & = & \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ f \circ \mathcal{D}[\![K[c]]\!],\ id)) \\ & = & \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ f \circ \mathcal{K}[\![K]\!](\mathcal{D}[\![c]\!]),\ id)) \\ & = & \mathcal{K}[\![\texttt{while (}b\texttt{) do }K]\!](\mathcal{D}[\![c]\!]) \end{array}$$

□

Korollar 39. Zwei Programme c_1 und c_2 mit gleicher Semantik sind in allen Kontexten austauschbar: Wenn $\mathcal{D}[\![c_1]\!] = \mathcal{D}[\![c_2]\!]$, dann $\mathcal{D}[\![K[c_1]]\!] = \mathcal{D}[\![K[c_2]]\!]$.

Beweis. Nach Thm. 38 gilt $\mathcal{D}[\![K[c_1]]\!] = \mathcal{K}[\![K]\!](\mathcal{D}[\![c_1]\!]) = \mathcal{K}[\![K]\!](\mathcal{D}[\![c_2]\!]) = \mathcal{D}[\![K[c_2]]\!]$. □

Beispiel 35. $c_1 =$ `while (`b`) do `c und $c_2 =$ `if (`b`) then `c`; while (`b`) do `c` 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.