

## Korrekte Software: Grundlagen und Methoden

Vorlesung 4 vom 23.04.19

### Äquivalenz der Operationalen und Denotationalen Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2019

11:27:21 2019-07-04

1 [16]



### Operationale vs. denotationale Semantik

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

**Denotional**  $\mathcal{A}[a]$

$$\begin{array}{ll}
 m \in \mathbb{Z} & \langle m, \sigma \rangle \rightarrow_{Aexp} m \qquad \{(\sigma, m) | \sigma \in \Sigma\} \\
 x \in Loc & \frac{x \in Dom(\sigma)}{\langle x, \sigma \rangle \rightarrow_{Aexp} \sigma(x)} \qquad \{(\sigma, \sigma(x)) | \sigma \in \Sigma, x \in Dom(\sigma)\} \\
 & \frac{x \notin Dom(\sigma)}{\langle x, \sigma \rangle \rightarrow_{Aexp} \perp} \\
 a_1 \circ a_2 & \frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} m}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} n \circ^I m} \quad \{(\sigma, n \circ^I m) | \sigma \in \Sigma, (\sigma, n) \in \mathcal{A}[a_1], (\sigma, m) \in \mathcal{A}[a_2]\} \\
 & \frac{n = \perp \text{ oder } m = \perp}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} \perp} \\
 & \quad \circ \in \{+, *, -\}
 \end{array}$$

Korrekte Software

3 [16]



### Fahrplan

- ▶ Einführung
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Der Floyd-Hoare-Kalkül
- ▶ Invarianten und die Korrektheit des Floyd-Hoare-Kalküls
- ▶ Strukturierte Datentypen
- ▶ Verifikationsbedingungen
- ▶ Vorwärts mit Floyd und Hoare
- ▶ Modellierung
- ▶ Spezifikation von Funktionen
- ▶ Referenzen und Speichermodelle
- ▶ Funktionsaufrufe und das Framing-Problem
- ▶ Ausblick und Rückblick

Korrekte Software

2 [16]



### Operationale vs. denotationale Semantik

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

**Denotional**  $\mathcal{A}[a]$

$$\frac{a_1 / a_2}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} n \circ^I m} \quad \frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} m \quad m \neq 0 \quad m, n \neq \perp}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} n \circ^I m} \quad \{(\sigma, n/m) | \sigma \in \Sigma, (\sigma, n) \in \mathcal{A}[a_1], (\sigma, m) \in \mathcal{A}[a_2], m \neq 0\} \\
 \frac{n = \perp \text{ oder } m = \perp}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} \perp} \quad \frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} m \quad n = \perp, m = \perp \text{ oder } m = 0}{\langle a_1 \circ a_2, \sigma \rangle \rightarrow_{Aexp} \perp}$$

Korrekte Software

4 [16]



### Äquivalenz operationale und denotationale Semantik

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

$$\begin{aligned}
 \langle a, \sigma \rangle \rightarrow_{Aexp} n &\Leftrightarrow (\sigma, n) \in \mathcal{A}[a] \\
 \langle a, \sigma \rangle \rightarrow_{Aexp} \perp &\Leftrightarrow \sigma \notin Dom(\mathcal{A}[a])
 \end{aligned}$$

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

Korrekte Software

5 [16]



### Operationale vs. denotationale Semantik

**Operational**  $\langle b, \sigma \rangle \rightarrow_{Bexp} 0 \mid 1$

**Denotional**  $\mathcal{B}[b]$

$$\begin{array}{ll}
 1 & \langle 1, \sigma \rangle \rightarrow_{Bexp} 1 \qquad \{(\sigma, 1) | \sigma \in \Sigma\} \\
 0 & \langle 0, \sigma \rangle \rightarrow_{Bexp} 0 \qquad \{(\sigma, 0) | \sigma \in \Sigma\}
 \end{array}$$

Korrekte Software

6 [16]



### Operationale vs. denotationale Semantik

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

**Denotional**  $\mathcal{B}[b]$

$$\begin{array}{ll}
 a_0 == a_1 & \frac{\langle a_0, \sigma \rangle \rightarrow_{Aexp} n \quad \langle a_1, \sigma \rangle \rightarrow_{Aexp} m \quad n, m \neq \perp \quad n = m}{\langle a_0 == a_1, \sigma \rangle \rightarrow_{Bexp} 1} \\
 & \qquad \{(\sigma, 1) | \sigma \in \Sigma, (\sigma, n) \in \mathcal{A}[a_0], (\sigma, m) \in \mathcal{A}[a_1], n_0 = n_1\} \\
 & \qquad \cup \\
 & \qquad \{(\sigma, 0) | \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{A}[a_0], (\sigma, n_1) \in \mathcal{A}[a_1], n_0 \neq n_1\} \\
 & \frac{n, m \neq \perp \quad n \neq m}{\langle a_0 == a_1, \sigma \rangle \rightarrow_{Bexp} 0} \\
 & \qquad \{(\sigma, 0) | \sigma \in \Sigma, (\sigma, n_0) \in \mathcal{A}[a_0], (\sigma, n_1) \in \mathcal{A}[a_1], n_0 \neq n_1\} \\
 & \frac{n = \perp \text{ oder } m = \perp}{\langle a_0 == a_1, \sigma \rangle \rightarrow_{Bexp} \perp} \\
 & \qquad \text{analog}
 \end{array}$$

$a_1 <= a_2$

analog

Korrekte Software

7 [16]



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

**Denotional**  $\mathcal{B}[b]$

$$\begin{array}{ll}
 b_1 \&& b_0 & \frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} 0}{\langle b_1 \&& b_2, \sigma \rangle \rightarrow 0} \qquad \{(\sigma, 0) | (\sigma, 0) \in \mathcal{B}[b_1]\} \\
 & \frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} 1}{\langle b_1 \&& b_2, \sigma \rangle \rightarrow 1} \\
 & \frac{\langle b_2, \sigma \rangle \rightarrow_{Bexp} b}{\langle b_1 \&& b_2, \sigma \rangle \rightarrow b} \qquad \{(\sigma, b) | (\sigma, b) \in \mathcal{B}[b_2]\} \\
 & \frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} \perp}{\langle b_1 \&& b_2, \sigma \rangle \rightarrow \perp} \\
 b_1 || b_2 & \text{analog} \\
 !n & \dots
 \end{array}$$

Korrekte Software

8 [16]



## Äquivalenz operationale und denotationale Semantik

- ▶ Für alle  $b \in \mathbf{Bexp}$ , für alle  $t \in \mathbb{B}$ , für alle Zustände  $\sigma$ :

$$\begin{aligned} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} t &\Leftrightarrow (\sigma, t) \in \mathcal{B}[\![b]\!] \\ \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \perp &\Leftrightarrow \sigma \notin \text{Dom}(\mathcal{B}[\![b]\!]) \end{aligned}$$

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

Korrekte Software

9 [16]



## Operationale vs. denotationale Semantik

$$\text{Operational} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \mid \perp$$

$$\text{Denotational } \mathcal{C}[\![c]\!]$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \perp}{\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}$$

**if** ( $b$ )  $c_0$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{1} \quad \langle c_0, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

$$\{(\sigma, \sigma') \mid (\sigma, \mathbf{1}) \in \mathcal{B}[\![b]\!], (\sigma, \sigma') \in \mathcal{C}[\![c_0]\!]\}$$

**else**  $c_1$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{0} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

$$\{(\sigma, \sigma') \mid (\sigma, \mathbf{0}) \in \mathcal{B}[\![b]\!], (\sigma, \sigma') \in \mathcal{C}[\![c_1]\!]\}$$

Korrekte Software

11 [16]



## Operationale vs. denotationale Semantik

$$\text{Operational} \quad \frac{\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \mid \perp}{\langle \{ \}, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma}$$

$$\text{Denotational } \mathcal{C}[\![c]\!] \quad \mathcal{C}[\![\{ \}] \!] = \text{Id}$$

$$\frac{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'' \neq \perp}{\begin{aligned} \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \\ \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma'' \\ \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \perp \\ \langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \perp \end{aligned}}$$

$$\mathcal{C}[\![c_2]\!] \circ \mathcal{C}[\![c_1]\!]$$

$$x = a \quad \frac{\begin{aligned} \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n \\ \langle x = a, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma[n/x] \\ \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} \perp \end{aligned}}{\langle x = a, \sigma \rangle \rightarrow_{\text{Stmt}} \perp} \quad \{(\sigma, \sigma[n/x]) | (\sigma, n) \in \mathcal{A}[\![a]\!]\}$$

Korrekte Software

10 [16]



## Operationale vs. denotationale Semantik

$$\text{Operational} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \mid \perp$$

$$\text{Denotational } \mathcal{C}[\![c]\!]$$

$$\frac{\text{while } (b) c}{w} \quad \frac{\begin{aligned} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{0} \\ \langle w, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma \end{aligned}}{\begin{aligned} \langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \perp \\ \langle w, \sigma \rangle \rightarrow_{\text{Stmt}} \perp \end{aligned}} \quad \text{fix}(\Gamma)$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{1} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \neq \perp \quad \langle w, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle w, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{1} \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}{\langle w, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}$$

mit

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

Korrekte Software

12 [16]



## Äquivalenz operationale und denotationale Semantik

- ▶ Für alle  $c \in \text{Stmt}$ , für alle Zustände  $\sigma, \sigma'$ :

$$\begin{aligned} \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' &\Leftrightarrow (\sigma, \sigma') \in \mathcal{C}[\![c]\!] \\ \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp &\Rightarrow \sigma \notin \text{Dom}(\mathcal{C}[\![c]\!]) \end{aligned}$$

- ▶ ⇒ Beweis Prinzip? per Induktion über die Ableitung in der operationalen Semantik (Warum?)

- ◀ ⇐ Beweis Prinzip? 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?)

Korrekte Software

13 [16]



## Knackpunkt

$$\begin{aligned} \mathcal{C}[\![w]\!] = \text{fix}(\Gamma) = \Gamma(\text{fix}(\Gamma)) = \Gamma\left(\bigcup_{i \geq 0} \Gamma^i(\emptyset)\right) = \bigcup_{i \geq 0} \Gamma(\Gamma^i(\emptyset)) \\ = \bigcup_{i \geq 0} \{(\sigma, \sigma') \mid (\sigma, \mathbf{1}) \in \mathcal{B}[\![b]\!], (\sigma, \sigma'') \in \mathcal{C}[\![c]\!], (\sigma'', \sigma') \in \Gamma^i(\emptyset)\} \\ \cup \{(\sigma, \sigma) \mid (\sigma, \mathbf{0}) \in \mathcal{B}[\![b]\!]\} \end{aligned}$$

mit  $w \equiv \text{while } (b) c$  Induktion über  $i \geq 0$

$$\frac{\{(\sigma, \sigma') \mid (\sigma, \mathbf{1}) \in \mathcal{B}[\![b]\!], (\sigma, \sigma'') \in \mathcal{C}[\![c]\!] \quad , \quad (\sigma'', \sigma') \in \Gamma^i(\emptyset)\} \cup \{(\sigma, \sigma) \mid (\sigma, \mathbf{0}) \in \mathcal{B}[\![b]\!]\}}{\langle b, \sigma \rangle \rightarrow_{\mathbf{Bexp}} \mathbf{1} \quad (\text{strukt. IH})(\sigma, \sigma) \rightarrow_{\text{Stmt}} \sigma' \quad (\leq i \text{ IH})(w, \sigma'') \rightarrow_{\text{Stmt}} \sigma' \quad \langle w, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

Korrekte Software

14 [16]



## Äquivalenz operationale und denotationale Semantik

- ▶ Für alle  $c \in \text{Stmt}$ , für alle Zustände  $\sigma, \sigma'$ :

$$\begin{aligned} \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' &\Leftrightarrow (\sigma, \sigma') \in \mathcal{C}[\![c]\!] \\ \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp &\Rightarrow \sigma \notin \text{Dom}(\mathcal{C}[\![c]\!]) \end{aligned}$$

- ▶ Gegenbeispiel für ⇐ in der zweiten Aussage: wähle  $c \equiv \text{while}(1)\{\}$ :  $\mathcal{C}[\![c]\!] = \emptyset$  aber  $\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp$  gilt nicht (sondern?).

Korrekte Software

15 [16]



## Fahrplan

- ▶ Einführung
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Der Floyd-Hoare-Kalkül
- ▶ Invarianten und die Korrektheit des Floyd-Hoare-Kalküls
- ▶ Strukturierte Datentypen
- ▶ Verifikationsbedingungen
- ▶ Vorwärts mit Floyd und Hoare
- ▶ Modellierung
- ▶ Spezifikation von Funktionen
- ▶ Referenzen und Speichermodelle
- ▶ Funktionsaufrufe und das Framing-Problem
- ▶ Ausblick und Rückblick

Korrekte Software

16 [16]

