

# Korrekte Software: Grundlagen und Methoden

## Vorlesung 3 vom 20.04.17: Operationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2017



## Fahrplan

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



## Zutaten

```
// GGT(A,B)
if (a == 0) r = b;
else {
  while (b != 0) {
    if (a <= b)
      b = b - a;
    else a = a - b;
  }
  r = a;
}
```

- ▶ Programme berechnen **Werte**
- ▶ Basierend auf
  - ▶ Werte sind **Variablen** zugewiesen
  - ▶ Evaluation von **Ausdrücken**
- ▶ Folgt dem Programmablauf



## Unsere Programmiersprache

Wir betrachten einen Ausschnitt der Programmiersprache **C (C0)**.

Ausbaustufe 1 kennt folgende Konstrukte:

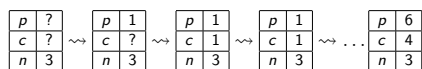
- ▶ Typen: **int**;
- ▶ Ausdrücke: Variablen, Literale (für ganze Zahlen), arithmetische Operatoren (für ganze Zahlen), Relationen (**==, !=, <=, ...**), boolesche Operatoren (**&&, ||**);
- ▶ Anweisungen:
  - ▶ Fallunterscheidung (**if... else...**), Iteration (**while**), Zuweisung, Blöcke;
  - ▶ Sequenzierung und leere Anweisung sind implizit



## Semantik von C0

- ▶ Die (operationale) Semantik einer imperativen Sprache wie C0 ist ein **Zustandsübergang**: das System hat einen impliziten Zustand, der durch Zuweisung von **Werten** an **Adressen** geändert werden kann.

- ▶ Konkretes Beispiel:  $n = 3$



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

### Systemzustände

- ▶ Ausdrücke werten zu **Werten V** (hier ganze Zahlen) aus.
- ▶ Adressen **Loc** sind hier Programmvariablen (Namen)
- ▶ Ein **Systemzustand** bildet Adressen auf Werte ab:  $\Sigma = \text{Loc} \rightarrow \mathbf{V}$
- ▶ Ein Programm bildet einen Anfangszustand **möglicherweise** auf einen Endzustand ab (wenn es **terminiert**).
- ▶ Zusicherungen sind Prädikate über dem Systemzustand.



## C0: Ausdrücke und Anweisungen

```
Aexp a ::= N | Loc | a1 + a2 | a1 - a2 | a1 * a2 | a1 / a2
Bexp b ::= 0 | 1 | a1 == a2 | a1 != a2
          | a1 <= a2 | !b | b1 && b2 | b1 || b2
Exp e ::= Aexp | Bexp
Stmt c ::= Loc = Exp;
          | if ( b ) c1 else c2
          | while ( b ) c
          | {c*}
```



## Eine Handvoll Beispiele

```
// {y = Y ∧ y ≥ 0}
x = 1;
while (y != 0) {
  y = y - 1;
  x = 2 * x;
}
// {x = 2^Y}
```

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

```
// {a ≥ 0 ∧ b ≥ 0}
r = b;
q = 0;
while (b <= r) {
  r = r - y;
  q = q + 1;
}
// {a = b * q + r ∧ r < b}
```

```
// {0 ≤ a}
t = 1;
s = 1;
i = 0;
while (s <= a) {
  t = t + 2;
  s = s + t;
  i = i + 1;
}
// {i^2 ≤ a ∧ a < (i + 1)^2}
```



## Operationale Semantik: Arithmetische Ausdrücke

Ein arithmetischer Ausdruck  $a$  wertet unter gegebenen Zustand  $\sigma$  zu einer ganzen Zahl  $n$  (Wert) aus oder zu einem Fehler  $\perp$ .

- ▶ **Aexp**  $a ::= \mathbf{N} \mid \text{Loc} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2$
- ▶ Zustände bilden Adressen/Programmvariablen auf **Werte** ab ( $\sigma$ )

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

Regeln:

$$\frac{}{\langle n, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{X \in \text{Loc}, X \in \text{Dom}(\sigma), \sigma(X) = v}{\langle X, \sigma \rangle \rightarrow_{Aexp} v} \quad \frac{X \in \text{Loc}, X \notin \text{Dom}(\sigma)}{\langle X, \sigma \rangle \rightarrow_{Aexp} \perp}$$



## Operationale Semantik: Arithmetische Ausdrücke

► **Aexp**  $a ::= \mathbf{N} \mid \mathbf{Loc} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2$   
 $\langle a, \sigma \rangle \rightarrow_{Aexp} n \mid \perp$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbf{N}, n \text{ Summe } n_1 \text{ und } n_2}{\langle a_1 + a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad \text{falls } n_1 = \perp \text{ oder } n_2 = \perp}{\langle a_1 + a_2, \sigma \rangle \rightarrow_{Aexp} \perp}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbf{N}, n \text{ Diff. } n_1 \text{ und } n_2}{\langle a_1 - a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad \text{falls } n_1 = \perp \text{ oder } n_2 = \perp}{\langle a_1 - a_2, \sigma \rangle \rightarrow_{Aexp} \perp}$$



## Operationale Semantik: Arithmetische Ausdrücke

► **Aexp**  $a ::= \mathbf{N} \mid \mathbf{Loc} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2$   
 $\langle a, \sigma \rangle \rightarrow_{Aexp} n \mid \perp$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbf{N}, n \text{ Produkt } n_1 \text{ und } n_2}{\langle a_1 * a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad \text{falls } n_1 = \perp \text{ oder } n_2 = \perp}{\langle a_1 * a_2, \sigma \rangle \rightarrow_{Aexp} \perp}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbf{N}, n_2 \neq 0, n \text{ Quotient } n_1, n_2}{\langle a_1 / a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad \text{falls } n_1 = \perp, n_2 = \perp \text{ oder } n_2 = 0}{\langle a_1 / a_2, \sigma \rangle \rightarrow_{Aexp} \perp}$$



## Beispiel-Ableitungen

Sei  $\sigma(X) = 6, \sigma(Y) = 5$ .

$$\frac{\langle X, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5 \quad \langle X, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle X + Y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\langle X, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle X - Y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\frac{\langle X + Y, \sigma \rangle \rightarrow_{Aexp} 11 \quad \langle X - Y, \sigma \rangle \rightarrow_{Aexp} 1}{\langle (X + Y) * (X - Y), \sigma \rangle \rightarrow_{Aexp} 11}$$

$$\frac{\langle X, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle X, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle X * X, \sigma \rangle \rightarrow_{Aexp} 36} \quad \frac{\langle Y, \sigma \rangle \rightarrow_{Aexp} 5 \quad \langle Y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle Y * Y, \sigma \rangle \rightarrow_{Aexp} 25}$$

$$\frac{\langle X * X, \sigma \rangle \rightarrow_{Aexp} 36 \quad \langle Y * Y, \sigma \rangle \rightarrow_{Aexp} 25}{\langle (X * X) - (Y * Y), \sigma \rangle \rightarrow_{Aexp} 11}$$



## Operationale Semantik: Boolesche Ausdrücke

► **Bexp**  $b ::= 0 \mid 1 \mid a_1 == a_2 \mid a_1 <= a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$   
**Regeln:**

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

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

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

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \neq \perp, n_1 \text{ und } n_2 \text{ gleich}}{\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} 1}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \neq \perp, n_1 \text{ und } n_2 \text{ ungleich}}{\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} 0}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_1 = \perp \text{ or } n_2 = \perp}{\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} \perp}$$



## Operationale Semantik: Boolesche Ausdrücke

► **Bexp**  $b ::= 0 \mid 1 \mid a_1 == a_2 \mid a_1 <= a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$   
**Regeln:**

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

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} 1}{\langle !b, \sigma \rangle \rightarrow_{Bexp} 0} \quad \frac{\langle b, \sigma \rangle \rightarrow_{Bexp} 0}{\langle !b, \sigma \rangle \rightarrow_{Bexp} 1} \quad \frac{\langle b, \sigma \rangle \rightarrow_{Bexp} \perp}{\langle !b, \sigma \rangle \rightarrow_{Bexp} \perp}$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} t_1 \quad \langle b_2, \sigma \rangle \rightarrow_{Bexp} t_2}{\langle b_1 \&\& b_2, \sigma \rangle \rightarrow_{Bexp} t}$$

wobei  $t = 1$  wenn  $t_1 = t_2 = 1$ ;  
 $t = 0$  wenn  $t_1 = 0$  oder  $(t_1 = 1 \text{ und } t_2 = 0)$ ;  
 $t = \perp$  sonst



## Operationale Semantik: Boolesche Ausdrücke

► **Bexp**  $b ::= 0 \mid 1 \mid a_1 == a_2 \mid a_1 <= a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$   
**Regeln:**

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

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} t_1 \quad \langle b_2, \sigma \rangle \rightarrow_{Bexp} t_2}{\langle b_1 \parallel b_2, \sigma \rangle \rightarrow_{Bexp} t}$$

wobei  $t = 0$  wenn  $t_1 = t_2 = 0$ ;  
 $t = 1$  wenn  $t_1 = 1$  oder  $(t_1 = 0 \text{ und } t_2 = 1)$ ;  
 $t = \perp$  sonst



## Operationale Semantik: Anweisungen

► **Stmt**  $c ::= \mathbf{Loc} = \mathbf{Exp}; \mid \{c^*\} \mid \mathbf{if} (b) c_1 \mathbf{else} c_2 \mid \mathbf{while} (b) c$   
**Beispiel:**

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

$$\langle X = 5, \sigma \rangle \rightarrow_{Stmt} \sigma'$$

wobei  $\sigma'(X) = 5$  und  $\sigma'(Y) = \sigma(Y)$  für alle  $Y \neq X$



## Operationale Semantik: Anweisungen

► **Stmt**  $c ::= \mathbf{Loc} = \mathbf{Exp}; \mid \{c^*\} \mid \mathbf{if} (b) c_1 \mathbf{else} c_2 \mid \mathbf{while} (b) c$   
**Regeln:**

**Definiere:**

$$\sigma[m/X](Y) := \begin{cases} m & \text{if } X = Y \\ \sigma(Y) & \text{sonst} \end{cases}$$

$$\langle X = 5, \sigma \rangle \rightarrow_{Stmt} \sigma[5/X]$$

**Es gilt:**

$$\forall \sigma, n, m, \forall X, Y. X \neq Y \Rightarrow \sigma[n/X][m/Y] = \sigma[m/Y][n/X]$$

$$\forall \sigma, n, m, \forall X. \sigma[n/X][m/X] = \sigma[m/X]$$



## Operationale Semantik: Anweisungen

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

Regeln:

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

$$\frac{\langle a, \sigma \rangle \rightarrow_{\text{Aexp}} n \in \mathbf{N}}{\langle X = a, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma[n/X]} \quad \frac{\langle a, \sigma \rangle \rightarrow_{\text{Aexp}} \perp}{\langle X = a, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}$$

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

$$\frac{\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}{\langle \{c c_s\}, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}$$

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

Korrekte Software

17 [24]



## Operationale Semantik: Anweisungen

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

Regeln:

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

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} 1 \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if}(b) c_1 \text{ else } c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} 0 \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if}(b) c_1 \text{ else } c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

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

Korrekte Software

18 [24]



## Operationale Semantik: Anweisungen

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

Regeln:

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

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} 0}{\langle \text{while}(b) c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} 1 \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle \text{while}(b) c, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle \text{while}(b) c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} 1 \quad \langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}{\langle \text{while}(b) c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp} \quad \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \perp}{\langle \text{while}(b) c, \sigma \rangle \rightarrow_{\text{Stmt}} \perp}$$

Korrekte Software

19 [24]



## Beispiel

```
x = 1;
while (y != 0) {
  y = y - 1;
  x = 2 * x;
}
// x = 2^y
σ(y) = 3
```

Korrekte Software

20 [24]



## Äquivalenz arithmetischer Ausdrücke

Gegeben zwei Aexp  $a_1$  and  $a_2$

► Sind sie gleich?

$$a_1 \sim_{\text{Aexp}} a_2 \text{ gdw } \forall \sigma, n. \langle a_1, \sigma \rangle \rightarrow_{\text{Aexp}} n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow_{\text{Aexp}} n$$

$$(X * X) + 2 * X * Y + (Y * Y) \quad \text{und} \quad (X + Y) * (X + Y)$$

► Wann sind sie gleich?

$$\exists \sigma, n. \langle a_1, \sigma \rangle \rightarrow_{\text{Aexp}} n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow_{\text{Aexp}} n$$

$$\begin{array}{ll} X * X & \text{und} \quad 9 * X + 22 \\ X * X & \text{und} \quad X * X + 1 \end{array}$$

Korrekte Software

21 [24]



## Äquivalenz Boolescher Ausdrücke

Gegeben zwei Bexp-Ausdrücke  $b_1$  and  $b_2$

► Sind sie gleich?

$$b_1 \sim_{\text{Bexp}} b_2 \text{ iff } \forall \sigma, b. \langle b_1, \sigma \rangle \rightarrow_{\text{Bexp}} b \Leftrightarrow \langle b_2, \sigma \rangle \rightarrow_{\text{Bexp}} b$$

$$A \mid \mid (A \ \&\& \ B) \quad \text{und} \quad A$$

Korrekte Software

22 [24]



## Beweisen

Zwei Programme  $c_0, c_1$  sind äquivalent gdw. sie die gleichen Zustandsveränderungen bewirken. Formal definieren wir

Definition

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

Ein einfaches Beispiel:

Lemma

Sei  $w \equiv \text{while}(b) c$  mit  $b \in \text{Bexp}$ ,  $c \in \text{Stmt}$ .

Dann gilt:  $w \sim \text{if}(b) \{c; w\} \text{ else } \{\}$

Beweis an der Tafel

Korrekte Software

23 [24]



## Zusammenfassung

- Operationale Semantik als ein Mittel für Beschreibung der Semantik
- Auswertungsregeln arbeiten entlang der syntaktischen Struktur
- Werten Ausdrücke zu Werten aus und Programme zu Zuständen (zu gegebenen Zustand)
- Fragen zu Programmen: Gleichheit

Korrekte Software

24 [24]

