

# Korrektheit und Hoare-Kalkül für Imperative Programme

Martin Wirsing

in Zusammenarbeit mit  
Matthias Hölzl, Piotr Kosciuzenko, Dirk Pattinson

04/03

Informatik II, SS 03

2

## Ziele

- Wiederholung des Begriffs der partiellen und totalen Korrektheit
- Wiederholung der Regeln des Hoare-Kalkül für **while**-Programme
- Lernen einfache **while**-Programme als korrekt zu beweisen

M. Wirsing: Korrektheit und HoareKalkül für imperative Programme 04/03

Informatik II, SS 03

3

## Hoare-Formel

Vorbedingung  $\{P\}$  Anweisung  $S$  Nachbedingung  $\{Q\}$

Formel

Zuweisung  
Fallunterscheidung  
sequ. Komp.  
Block  
Iteration

Formel

Trifft auf eine Menge von Zuständen zur.  
Enthält Boolesche Operatoren  $\&$ ,  $!$ ,  $\vee$ ,  
log. Variablen, „Logische“ Variablen

### Bedeutung von Hoare-Formeln

#### 2 Interpretationen:

- partielle Korrektheit
- totale Korrektheit**

M. Wirsing: Korrektheit und HoareKalkül für imperative Programme 04/03

Informatik II, SS 03

4

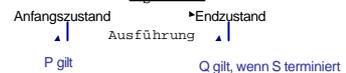
## Partielle Korrektheit

$\{P\} S \{Q\}$  ist gültig,

wenn S **partiell korrekt** ist bzgl.

Vorbedingung P und Nachbedingung Q,

d.h.



d.h. wenn folgendes gilt:

wenn P im Anfangszustand von S gilt und wenn S terminiert,  
dann gilt Q nach Ausführung von S

M. Wirsing: Korrektheit und HoareKalkül für imperative Programme 04/03

Informatik II, SS 03

5

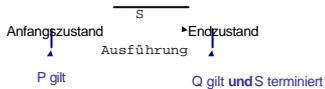
## Totale Korrektheit

$\{P\} S \{Q\}$  ist gültig,

wenn S **total korrekt** ist bzgl.

Vorbedingung P und Nachbedingung Q,

d.h.



d.h. wenn folgendes gilt:

wenn P im Anfangszustand von S gilt,

dann terminiert S und Q gilt nach Ausführung von S

M. Wirsing: Korrektheit und HoareKalkül für imperative Programme 04/03

Informatik II, SS 03

6

## Folgerung

- Totale Korrektheit = Partielle Korrektheit + Terminierung

- Für eine Anweisung S ohne Iteration stimmen totale und partielle Korrektheit überein.

M. Wirsing: Korrektheit und HoareKalkül für imperative Programme 04/03

## Partielle und totale Korrektheit

### Beispiele

• Totale Korrektheit (und partielle Korrektheit):

```
{true} if (y>0) x=y; else x=-y; {x == |y|}
{x>=0} if (y>0) x=y; else x=-y; {x>=0}
{x>1} x=x+1; y=x; {y>2 & x>2}
{x>=0} while (x!=0) x=x-1; {x==0}
```

• Partielle Korrektheit (aber nicht totale Korrektheit):

```
{true} while (x!=0) x=x-1; {x==0}
(terminiert nicht für x<0!)
```

## Beschreibung von Nichttermination

Die Hoare-Formel

```
{x>0} while (x>0) x=x+1; {false}
```

terminiert nie! Sie ist partiell korrekt, aber nicht total korrekt.

Allgemein,  $\{P\} S \{false\}$  drückt Nichtterminierung aus, d.h

$\{P\} S \{false\}$  partiell korrekt  $\Rightarrow$

S terminiert **nicht** für alle Anfangszustände, die P erfüllen.

## Partielle und totale Korrektheit: Hoare-Kalkül

▪ Der Hoare-Kalkül dient zum (konstruktiven) Beweisen von partieller und totaler Korrektheit

▪ Idee von Hoare:

Leite (rückwärts schreitend)  
ausgehend von der (gewünschten) Nachbedingung die  
Vorbedingung ab

## Hoare-Regel Zuweisung

Zuweisungsaxiom

```
 $\{P[x \mapsto \text{exp}]\} x = \text{exp}; \{P\}$ 
```

Ersetze x in P durch exp

Deklaration

```
 $\{P[x \mapsto x]\} \text{type } x = \text{exp}; \{P\}$ 
```

Beispiele

```
{max? - C == 35} max = max - C; {max == 35}
```

```
{max - ? == 35} int C = 5; {max - C == 35}
```

## Hoare-Regel Abschwächung

Abschwächungsregel

$$\frac{P1 \Rightarrow P, \{P\} S \{Q\}, Q \Rightarrow Q1}{\{P1\} S \{Q1\}}$$

Beispiel

```
n=3  $\Rightarrow$  2n>=6, {2n>=6} n = 2*n; {n>=6}, n>=6  $\Rightarrow$  n>5 (Abschwchg.)
{n=3} n = 2*n; {n>5}
```

## Hoare-Regeln

Fallunterscheidungsregel

$$\frac{\{b \& P\} S1 \{Q\} \quad \{\neg b \& P\} S2 \{Q\}}{\{P\} \text{if } (b) S1 \text{ else } S2 \{Q\}} \text{ (if)}$$

Sequentielle Komposition

$$\frac{\{P\} S1 \{R\} \quad \{R\} S2 \{Q\}}{\{P\} S1 S2 \{Q\}} \text{ (seq. Komp.)}$$

## Hoare-Regel Fallunterscheidung

### Beispiel

$\{x>=0 \ \& \ x == A\} \quad y = x; \quad \{x == A \ \& \ y == |A|\} *$   
 $\{x < 0 \ \& \ x == A\} \quad y = -x; \quad \{x == A \ \& \ y == |A|\} **$

$\{x == A\} \quad \text{if } (x>=0) \ y = x; \ \text{else } y = -x; \ \{x == A \ \& \ y == |A|\}$   
 wegen  
 $x>=0 \ \& \ x == A \Rightarrow x == A \ \& \ x == |A|$   
 $\{x == A \ \& \ x == |A|\} \quad y = x; \quad \{x == A \ \& \ y == |A|\}$  (Abschw.)  
 $x < 0 \ \& \ x == A \Rightarrow x == A \ \& \ -x == |A|$   
 $\{x == A \ \& \ -x == |A|\} \quad y = -x; \quad \{x == A \ \& \ y == |A|\}$  (Abschw.)  
 \*\* wegen  
 $x < 0 \ \& \ x == A \Rightarrow x == A \ \& \ -x == |A|$   
 $\{x == A \ \& \ -x == |A|\} \quad y = -x; \quad \{x == A \ \& \ y == |A|\}$  (Abschw.)  
 $\{x < 0 \ \& \ x == A\} \quad y = -x; \quad \{x == A \ \& \ y == |A|\}$  (Abschw.)

## Hoare-Regel Block

### Block-Regel

$\frac{\{P\} S \{Q\}}{\{P\} \{S\} \{Q\}}$  (falls P und Q keine in S deklarierten lokalen Variablen enthalten)

### Beispiel

$\{max == 40\} \ \text{final int } C = 5; \ max = max - C; \ \{max == 35\}$   
 $\{max == 40\} \ \{\text{final int } C = 5; \ max = max - C;\} \ \{max == 35\}$

## Iteration: Hoare-Regeln

### Partielle Korrektheit:

$\frac{\{b \ \& \ I\} \ S \ \{I\}}{\{I\} \ \text{while } (b) \ S \ \{(!b) \ \& \ I\}}$  (Iteration<sub>partiell</sub>)

### Totale Korrektheit:

$\frac{\{b \ \& \ I\} \ S \ \{I\} \quad \{b \ \& \ I \ \& \ t == z\} \ S \ \{t < z\} \quad //t \ \text{wird echt kleiner}}{I \Rightarrow t \geq 0} \quad //t \ \text{nie negativ}$  (Iteration<sub>total</sub>)  
 $\{I\} \ \text{while } (b) \ S \ \{(!b) \ \& \ I\}$

t – ein Integer-Ausdruck für die Terminierung der while-Schleife  
 z – eine „logische“ Variable, die nicht in I, b, S oder t vorkommt, also durch S nicht verändert wird.

## Hoare-Regel While

### Beispiel partielle Korrektheit

Sei  $I \equiv (n \leq end+1)$  (Invariante)

$\frac{\{n \leq end \ \& \ n \leq end+1\} \quad n = n+1; \quad \{n \leq end+1\}}{\{n \leq end+1\} \ \text{while } (n \leq end) \ n = n+1; \ \{n \leq end+1 \ \& \ !(n \leq end)\}}$

### Beispiel totale Korrektheit

Sei  $t \equiv end+1-n$

$\frac{\{n \leq end \ \& \ n \leq end+1\} \quad n = n+1; \quad \{n \leq end+1\} \quad \{n \leq end \ \& \ n \leq end+1 \ \& \ (end+1-n) == z\} \quad n = n+1; \quad \{(end+1-n) < z\}}{n \leq end+1 \Rightarrow (end+1-n) \geq 0} \quad //t \ \text{nie negativ}$   
 $\{n \leq end+1\} \ \text{while } (n \leq end) \ n = n+1; \ \{n \leq end+1 \ \& \ !(n \leq end)\}$

## Zusammenfassung

- Partielle und totale Korrektheit sind wichtige Begriffe zur Beschreibung des Ein- / Ausgabe-Verhaltens eines Programms.
- Der Hoare-Kalkül erlaubt den Beweis der partiellen und totalen Korrektheit (kleiner) Programme.