Korrektheit und Hoare-Kalkül für Imperative Programme

Martin Wirsing

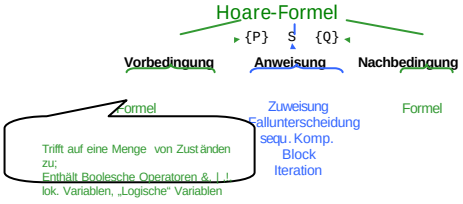
in Zusammenarbeit mit
Matthias Hölzl, Piotr Kosiuczenko, Dirk Pattinson

04/03

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

Hoare-Formel



Bedeutung von Hoare-Formeln

2 Interpretationen:

- partielle Korrektheit
- totale Korrektheit

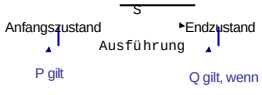
Partielle Korrektheit

{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

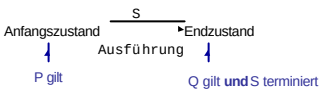
Totale Korrektheit

{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

Folgerung

- Totale Korrektheit = Partielle Korrektheit + Terminierung
- Für eine Anweisung S ohne Iteration stimmen totale und partielle Korrektheit überein.

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 ⇒

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

$$\{P[exp/x]\}\ x = exp;\ \{P\}$$

Ersetze x in P durch exp

Deklaration $\{P[exp/x]\}$ **type** x = 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

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

Hoare-Regeln

Fallunterscheidungsregel

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

Sequentielle Komposition

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

Hoare-Regel Fallunterschiedung

Beispiel

```
{x>=0 & x == A}      y = x;          {x == A & y ==|A|} *
{x< 0 & x == A}      y = -x;         {x == A & y ==|A|} **
```

```
{x == A}   if (x>=0) y = x; else y = -x; {x == A & y ==|A|}
```

* wegen
x>=0 & x == A ⇒ x == A & x ==|A|

```
{x == A & x ==|A|}   y = x;   {x == A & y ==|A|}
```
(Abschwchg.)

```
{x>=0 & x == A}      y = x;   {x == A & y ==|A|}
```

** wegen
x<0 & x == A ⇒ x == A & -x ==|A|

Zuweisgsaxiom

```
{x == A & -x ==|A|}  y = -x;  {x == A & y ==|A|}
```
(Abschwchg.)

```
{x<0 & x == A}       y = -x; {x == A & y ==|A|}
```


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}  final int C = 5; max = max – C;  {max == 35}
```

```
{max == 40} {final int C = 5; max = max – C;} {max == 35}
```


Iteration: Hoare-Regeln

Partielle Korrektheit:

Invariante

$$\frac{\{b\ \&\ I\}\ S\ \{I\}}{\{I\}\ \textbf{while}\ (b)\ \ S\ \{(!b)\ \&\ I\}}\quad (\text{Iteration}_{partiell})$$

Totale Korrektheit:

$$\frac{\begin{array}{l}\{b\ \&\ I\}\ S\ \{I\}\\ \{b\ \&\ I\ \&\ t == z\}\ S\ \{t < z\}\ \text{//t wird echt kleiner}\\ I \Rightarrow t >= 0\end{array}}{\{I\}\ \textbf{while}\ (b)\ \ S\ \{(!b)\ \&\ I\}}\quad \begin{array}{l}\text{//t nie negativ}\\ (\text{Iteration}_{total})\end{array}$$

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 ≡ (n <= end+1) (Invariante)

```
{n <= end & n <= end+1}      n = n+1;      {n <= end+1}
```

```
{n <= end+1 } while (n <= end) n = n+1; {n <= end+1 & !(n <= end)}
```

Beispiel totale Korrektheit Sei t ≡ end+1-n

```
{n <= end & n <= end+1}                  n = n+1;    {n <= end+1}
{n <= end & n <= end+1 & (end+1-n)==z}   n = n+1;    {(end+1-n)<z}
                n <= end+1   ⇒  (end+1-n)>= 0
```

```
{n <= end+1 } while (n <= end) n = n+1; {n <= end+1 & !(n <= 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.