Hoare Logik

Prinzip

Schrittweiser Nachweis von Eigenschaften eines Programmes
mit Vor- und Nachbedingungen

Technik

Hoare-Kalkül = Transformationsregeln zwischen Vor- und Nachbedingung

Typische Regeln

$$\frac{\{p\}\ S_1\ \{r\},\quad \{r\}\ S_2\ \{q\}}{\{p\}\ S_1\texttt{;}\ S_2\ \{q\}}$$

$$\{p[v/e]\}\ v\ \texttt{:=}\ e\ \{p\}$$

$$\frac{\{p \wedge B\}\ S\ \{p\}}{\{p\}\ \texttt{while}\ B\ \texttt{do}\ S\ \texttt{end}\ \{p \wedge \neg B\}}$$

mit Schleifeninvariante p

Beispiel Hoare Logik: Summe der ersten n Zahlen

$\{n > 0\}$
`i := 0; s := 0`
$\{n > 0,\ i = 0,\ s = 0\}$
$\{i \leq n,\ s = \sum_{j=0}^{i} j\}$ Schleifeninvariante p
`while i < n do` Bedingung $B \equiv i < n$
  $\{i < n,\ i \leq n,\ s = \sum_{j=0}^{i} j\}$
  $\{i < n,\ s = \sum_{j=0}^{i} j\}$
  `i := i + 1;`
  $\{i \leq n,\ s = \sum_{j=0}^{i-1} j\}$
  `s := s + i;`
  $\{i \leq n,\ s = i + \sum_{j=0}^{i-1} j\}$
  $\{i \leq n,\ s = \sum_{j=0}^{i} j\}$
`end;` Terminierungsfunktion $n - i$
$\{\neg i < n,\ i \leq n,\ s = \sum_{j=0}^{i} j\}$
$\{s = \sum_{j=0}^{n} j\}$

Hoare Logik Einschränkungen

Halteproblem
Es gibt kein Programm, das die Terminierung von Programmen berechnen kann!

Schleifen lassen sich automatisch nur begrenzt aufrollen

- Schleifeninvarianten sind schwer zu finden (entspricht Induktionsinvarianten)
- Nach- bzw. Vorbedingungen von Schleifen lassen sich nicht automatisch berechnen
- Terminierungsfunktionen sind schwer zu finden
- Pointer und Heap-Speicher problematisch, z.B. Aliasing

Ähnliches gilt für Rekursion statt Schleifen.