# Vorlesung vom 25.04.06: Einführung

## Organisatorisches

- *Veranstalter*: Christoph Lüth
  `cxl@informatik.uni-bremen.de`
  `http://www.informatik.uni-bremen.de/~cxl`
  MZH 8050, Tel. 7585
- *Termine*: Vorlesung: Dienstag 10– 12 MZH 8090
  Übung: Donnerstag 9 s.t. – 10 MZH 6240



## Struktur

- *Vorlesung* wie gehabt
- *Übungen*:
  - Beginn *04.05.06* (nächste Woche)
  - *Eine* längere Übung: Implementation eines Verifikationswerkzeuges
  - *Teilaufgaben* pro Woche
  - *Lösungen* vorstellen



## Inhalt

- Statt *breiter Übersicht* (FMSE I) *Schwerpunkte*.
- *Programmverifikation*:
- *Entwicklung korrekter Programme*:
- Blick in die Industrie: *Anwendungen*



## Inhalt — Programmverifikation

- Floyd-Hoare-Logik (zur Wiederholung),
- Prädikatentransformer und schwächste Vorbedingungen,
- Generation von Verifikationsbedingungen,
- Modellierung von imperativen Programmen,
- Verifikationswerkzeuge.



## Inhalt — Entwicklung korrekter Programme

- Entwicklung von korrekten Programmen durch Transformation,
- Algorithmendesign,
- Transformationswerkzeuge.



## Floyd-Hoare-Logik

- *Vor-* und *Nachbedingungen:*

$$\{A\}c\{B\}$$

  *Bedeutung*: jeder Zustand, der $A$ erfüllt, ist nach Termination von $c$ ein Zustand, der $B$ erfüllt.
- Hoare-Kalkül: *Beweisregeln*



## Regeln des Hoare-Kalküls

$$\frac{}{\{A\}\texttt{skip}\{A\}}\,Skip \qquad \frac{}{\{B[a/X]\}X := a\{B\}}\,Assgn$$

$$\frac{\{A\}c_0\{C\} \quad \{C\}c_1\{B\}}{\{A\}c_0; c_1\{B\}}\,Seq$$

$$\frac{\{A \wedge b\}c_0\{B\} \quad \{A \wedge \neg b\}c_1\{B\}}{\{A\}\texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1\ \texttt{fi}\{B\}}\,Cond$$

$$\frac{\{A \wedge b\}c\{A\}}{\{A\}\texttt{while}\ b\ \texttt{do}\ c\ \texttt{od}\{A \wedge \neg b\}}\,While$$

$$\frac{(A \Longrightarrow A') \quad \{A'\}c\{B'\} \quad (B' \Longrightarrow B)}{\{A\}c\{B\}}\,Impl$$

# Beispiel

```
int r= x;
int q= 0;
while (r >= y) do
  r= r-y;
  q= q+1
  od
```



# Fazit Hoare-Kalkül

- Umständlich zu ‘rechnen’
- Vermischung von Logik und Berechnung
  - z.B. Bedingung bei `if`, Invariante bei `while`
  - Dadurch: Seiteneffekte problematisch

```
if (x++ == 0) then p else q fi;
```



# Nächste Vorlesung

- *Korrektheit* des Hoare-Kalküls
- Dafür: *Semantik* durch Prädikatentransformer
- Damit: systematische Berechnung von Vorbedingungen