

#### **Tutorium 42, #9, Weihnachtstutorium**

Max Göckel- uzkns@student.kit.edu

Institut für Theoretische Informatik - Grundbegriffe der Informatik

#### **Algorithmen**



Ein Algorithmus ist eine Vorschrift um ein Problem zu lösen, dabei überführt ein Algorithmus eine Eingabe in eine Ausgabe. Diese Vorschrift besteht dabei aus endlich vielen elementaren Einzelschritten, die dann zusammen den Algorithmus formen.

Man kann sich einen Algorithmus wie ein Kochrezept vorstellen das die Zutaten (Eingabe) in ein Endprodukt (Ausgabe) überführt.

### Hoare-Kalkühl: Grundlagen



Um die Korrektheit eines Algorithmus zu beweisen haben wir das Hoare-Kalkühl.

Dies beweist die Korrektheit eine Algorithmus indem es Aussagen (sog. "Zusicherungen") über den Zustand der Ausführung macht.

Eine Aussage zeigt dabei die Korrektheit eines Algorithmus bis zu dem jeweiligen Punkt.

### Hoare-Kalkühl: Tripel



Ein Hoare-Tripel besteht aus zwei Zusicherungen und einem Stück Programmcode:

- {P}: Vorbedingung
  - P soll wahr sein bevor der Programmcode beginnt
- S: Codesegment
  - Ein Stück Algorithmus eine oder mehrere Zeilen vom Code
- {Q}: Nachbedingung
  - Ist P wahr und wurde S ausgeführt so soll nun Q wahr sein.

Eine Hoare-Zusicherung hat also die  $\{P\}S\{Q\}$ 

## Hoare-Kalkühl: Tripel



Für die Tripel gelten bestimmte Regeln (HT<sub>1</sub>, HT<sub>2</sub>, HT<sub>3</sub>):

- $HT_1$  Gilt  $P' \to P \land Q \to Q'$  und ist  $\{P\}S\{Q\}$  korrekt, so ist auch  $\{P'\}S\{Q'\}$  korrekt
- $HT_2 \ \{P\}S_1\{Q\} \ \text{und} \ \{Q\}S_2\{R\} \ \text{sind korrekt} \to \{P\}S_1S_2\{R\} \ \text{ist korrekt}$
- $HT_3$  Ist eine eine Zuweisung  $x \leftarrow \beta$ , so kann man aus der Nachbed. eine Vorbed. erstellen indem man x mit  $\beta$  substituiert

#### Hoare-Kalkühl: HT<sub>1</sub>



Wir probieren immer eine möglich ungenaue Vorbedigung und eine mögichst genaue Nachbedingung finden. Wieso?

Unser Algorithmus soll für möglichst viele Eingaben das richtige ngung Ergebnis produzieren

- ⇒ Möglichst allgemeine Vorbedingung finden
- Über das Ergbnis der Überführung soll man so viele Aussagen wie ngung möglich treffen können
  - ⇒ Möglichst genaue Nachbedingung finden

## Hoare-Kalkühl: Beispiel



$$S_1 = b \leftarrow a, S_2 = c \leftarrow b, Q = \{c = 5\}.$$

$$\begin{cases}
?_1 \\
b \leftarrow a; \\
?_2 \\
c \leftarrow b; \\
Q
\end{cases}$$

Finde  $?_1$ ,  $?_2$  heraus.

### Hoare-Kalkühl: Beispiel



$$S_1 = b \leftarrow \textit{a}, \, S_2 = \textit{c} \leftarrow \textit{b}, \, \textit{Q} = \{\textit{c} = 5\}.$$

```
\begin{cases}
?_1 \\
b \leftarrow a; \\
?_2 \\
c \leftarrow b;
\end{cases}
```

Finde  $?_1$ ,  $?_2$  heraus.

$$?_1: \{a=5\} ?_2: \{b=5\}$$

# Hoare-Kalkühl: Aufgabe



$$S = x \leftarrow (x - y) \cdot (x + y), \ Q = \{x + y^2 \neq 0\}.$$

Bestimme die Vorbedigung *P*.

## Hoare-Kalkühl: Aufgabe



$$S = x \leftarrow (x - y) \cdot (x + y), \ Q = \{x + y^2 \neq 0\}.$$

Einsetzen ( $HT_3$ ) in Q:

$$\{(x-y) \cdot (x+y) + y^2 \neq 0\} \Leftrightarrow \{x^2 - y^2 + y^2 \neq 0\} \Leftrightarrow \{x^2 \neq 0\} = P$$

# Hoare-Kalkühl: Verzweigungen



In Algortihmen gibt es auch Verzweigungen der Form if B then S else T. Diese können mit Hoare-Tripeln ausgewertet werden.

 $\{P\}$  Verzweigung  $\{Q\}$  wahr  $\Leftrightarrow \{P \land B\}S\{Q\}$  u.  $\{P \land \neg B\}T\{Q\}$  wahr d.h. für eine Verzweigung sind beide Fälle wahr wenn man die Bedigung im if-Teil umdreht.

#### Hoare-Kalkühl: Schleifen



Neben Verzweigungen gibt es auch Schleifen. Wir reden hier nur von Schleifen der Form <code>while B do S</code>, wo das Tripel  $\{I \land B\}S\{I\}$  gelten soll. Dabei ist I die Schleifeninvariante. Eine Schleifeninvariante ist eine Bedigung, die vor und nch jedem Durchlauf der Schleife gültig ist.

Gilt die Invariante I so gilt auch  $\{I\}$  Loop  $\{I \land \neg B\}$ 

#### Hoare-Kalkühl: Schreibweise



#### if-then-else

 $\begin{array}{c} \{P\} \\ \text{if $B$ then} \\ \{P \wedge B\} \\ S \\ \{Q\} \\ \text{else} \\ \{P \wedge \neg B\} \\ T \end{array}$ 

#### while-do

while B do  $\{I \land B\}$  S  $\{I\}$   $\{I \land \neg B\}$ 

### Hoare-Kalkühl: Beispiele



#### if-then-else

if (x > y) then  $z \leftarrow y$  else

 $Z \leftarrow X$ 

Mit Eingabe a,b für x,y.

#### while-do

$$s \leftarrow a$$
  
 $y \leftarrow b$   
for  $i \leftarrow 0$  to  $b - 1$  do  
 $s \leftarrow s + 1$ 

$$y \leftarrow y - 1$$

#### Graphen: Grundlagen



Ein Graph ist eine Struktur, die eine Menge von Elementen und eine Verbindung dieser Elemente darstellt.

Dabei ist V die Menge aller Elemente auf dem Graph (die Knoten, engl. Vertex) und E die Verbindung zwischen den Elementen (die Kanten, engl. Edges)

#### Formal:

Ein Graph G ist ein Tupel (V, E) mit V Menge der Knoten und E Menge der Verbindungen/Kanten.

### Graphen: Gerichtet vs. Ungerichtet



#### Gerichtet

- G = (V, E)
- $E \subseteq V \times V$
- E ist also Teilmenge des karth.
   Produktes und besteht aus
   Tupeln der Form (A, B)
- (A, B) heißt damit "von A nach B"
- Die Relation wird durch einen Pfeil  $A \rightarrow B$  ausgdrückt

#### Ungerichtet

- G = (V, E)
- $E \subseteq \{\{x,y\}|x,y\in V\}$
- E Teilmenge einer Menge aus Mengen der Form {A, B}
- {A, B} heißt damit "A und B sind verbunden"
- Die Relation wird durch einen Strich A – B ausgdrückt

#### Beispielgraph





Figure: Ausweichfahrplan S5/S52 März 2017

## Graphen: Teilgraphen



Ein Teilgraph ist ähnlich einer Teilmenge nur ein bestimmter Abschnitt vom Graph. Dabei ist egal ob der Teilgraph zusammenhängt (Alle Knoten verbunden sind) oder nicht.

#### Formal:

Ist G = (V, E) ein gerichteter Graph so ist T = (V', E') mit  $V' \subseteq V$  und  $E' \subseteq E \cap (V' \times V')$  ein Teilgraph von G. Ist G ungerichtet so ist  $E' \subseteq E \cap \{\{x, y\} | x, y \in V'\}$ 

### Graphen: Wege und Pfade



Sind zwei Knoten mit einer Kante verbunden, so sind sie adjazent.

In einem gerichteten Graph ist ein Pfad von A nach B eine Liste aus zueinander adjazenten Knoten. Die Länge des Pfades ist dabei die Anzahl der Kanten zwischen den Knoten.

In einem ungerichteten Graph ist ein Weg von A nach B ein ungerichteter Pfad von A nach B.

- Gerichteter Graph G = (V, E) streng zusammenhängend  $\Leftrightarrow \forall A, B \in V \exists P \text{fad } A \to B$
- Ungerichteter Graph G = (V, E) zusammenhängend  $\Leftrightarrow$  G ist aus einem Stück

#### Graphen: Grade



Der Grad eines Knoten in einem gerichteten Graph ist die Anzahl der Pfeile die von ihm weggehen und zu ihm führen.

- Der Eingangsgrad d<sup>-</sup>(D) ist die Anzahl der Kanten die in D hineingehen
- Der Ausgangsgrad d<sup>+</sup>(D) ist die Anzahl an Kanten die aus D weggehen
- der Grad d(D) ist  $d^+(D) + d^-(D)$ , also alle Kanten kombiniert

Für ungerichtete Graphen gibt es keinen Eingangs-/Ausgangsgrad, also ist der Grad d(D) die Anzahl der Kanten in denen D vorkommt.