

Grundlagen der Programmierung

Vorlesung 5
Sebastian Iwanowski
FH Wedel

Grundlagen der Programmierung

1. Einführung

Grundlegende Eigenschaften von Algorithmen und Programmen

2. Logik

Aussagenlogik

Prädikatenlogik

3. Programmentwicklung und –verifikation



Grundlagen der Programmverifikation, Zuweisungen

Verbundanweisungen, Verzweigungen

Schleifen

Modularisierung

Rekursion

4. Entwurf und Analyse von Algorithmen

Klassifikation von Algorithmen

Programmierung von Algorithmen

Bewertung von Algorithmen

Programmentwicklung

Konstruktionsproblem:

Gegeben eine Spezifikation:

Funktion, die einem Argument einen Funktionswert zuordnet

Entwirf ein Programm, das jedes Argument des Definitionsbereichs als Eingabe akzeptiert und den zugehörigen Funktionswert als Ausgabe produziert.

Verifikationsproblem:

Gegeben eine Spezifikation und ein Programm:

Beweise, dass das Programm für jedes Argument der Spezifikation als Eingabe den zugehörigen Funktionswert der Ausgabe berechnet.

- Zusatzaufgaben:**
1. Was wird bei anderen Eingaben als den zulässigen Argumenten berechnet ?
 2. Welche Bedingungen muss die Eingabe erfüllen, um bestimmte Ausgaben auszuschließen ?

Programmverifikation

Etwas allgemeiner:

Gegeben eine Spezifikation:

Funktion, die Startzuständen eindeutige Endzustände zuordnet.

Gegeben ein Programm:

Beweise, dass das Programm für jeden Startzustand den von der Spezifikation geforderten Endzustand erreicht und danach stoppt.

- Zusatzaufgaben:**
1. Welche Zustände werden bei anderen als den geforderten Startzuständen erreicht ?
 2. Welche Bedingungen müssen die Startzustände erfüllen, um bestimmte Endzustände auszuschließen ?

Anmerkung: Die gleichen Aufgaben können natürlich auch für beliebige Zwischenzustände untersucht werden.

Programmverifikation

Formalismus zum Lösen der Verifikationsaufgaben: Hoare-Tripel

$$\{ \varphi \} \quad P \quad \{ \psi \}$$

Vorbedingung Programm Nachbedingung

Da ein Programm eine Sequenz von Anweisungen ist, kann man dieses Vorgehen auf die einzelnen Anweisungen reduzieren:

$$\{ \varphi \} \quad S \quad \{ \psi \}$$

Vorbedingung Anweisung Nachbedingung

Fragestellungen:

1. Gegeben φ , finde stärkste Nachbedingung ψ
2. Gegeben ψ , finde schwächste Vorbedingung φ

Programmverifikation

Beispiel für das Arbeiten mit Hoare-Tripeln:

$\{ v \} \quad \varphi$

$z := x \cdot y ; \quad S$

$\{ z \geq 0 \} \quad \psi$

$w := \text{sqrt} (z)$

$\varphi_1 \Leftrightarrow (x > 0) \wedge (y > 0)$ ist eine Vorbedingung für ψ : $\{\varphi_1\} S \{\psi\}$

$\varphi_2 \Leftrightarrow (x < 0) \wedge (y < 0)$ ist auch eine Vorbedingung für ψ : $\{\varphi_2\} S \{\psi\}$

Welche ist die schwächste Vorbedingung V ?

$V \Leftrightarrow ((x > 0) \wedge (y > 0)) \vee ((x < 0) \wedge (y < 0)) \vee (x = 0) \vee (y = 0)$

ist die schwächste Vorbedingung für ψ :

$\{V\} S \{\psi\}$ Außerdem gilt: $\{\varphi\} S \{\psi\} \Rightarrow (\varphi \rightarrow V)$

Programmverifikation

Logischer Zusammenhang von Vorbedingungen:

$$(\{\varphi_1\} S \{\psi\}) \wedge (\varphi_2 \rightarrow \varphi_1) \quad \Rightarrow \quad \{\varphi_2\} S \{\psi\}$$

Die Vertauschung gilt nicht:

~~$$(\{\varphi_1\} S \{\psi\}) \wedge (\varphi_1 \rightarrow \varphi_2) \quad \Rightarrow \quad \{\varphi_2\} S \{\psi\}$$~~

Logischer Zusammenhang von Nachbedingungen:

$$(\{\varphi\} S \{\psi_1\}) \wedge (\psi_1 \rightarrow \psi_2) \quad \Rightarrow \quad \{\varphi\} S \{\psi_2\}$$

Die Vertauschung gilt nicht:

~~$$(\{\varphi\} S \{\psi_1\}) \wedge (\psi_2 \rightarrow \psi_1) \quad \Rightarrow \quad \{\varphi\} S \{\psi_2\}$$~~

Verifikation von Zuweisungen

Definition einer Zuweisung:

$x := \text{Ausdruck}$

Hierbei ist x ein beliebiger Variablenname und **Ausdruck** eine beliebige Funktion, die unter Umständen von Variablen abhängt. Die Variablen in **Ausdruck** müssen zum Zeitpunkt der Anweisung Werte haben.

Funktionsweise einer Zuweisung:

{ Prädikate für die Werte von x_1, \dots, x_k
und eventuell weitere Prädikate }

$x := \text{Ausdruck} (x_1, \dots, x_k)$

{ Neues Prädikat für den Wert von x und
eventuell weitere Prädikate }

Zunächst wird **Ausdruck** (x_1, \dots, x_k) ausgewertet.

Der sich ergebende Funktionswert wird danach in die Variable x geschrieben.

Verifikation von Zuweisungen

Beispiel für die Verifikation einer Zuweisung:

$\{ (x > 0) \wedge (y > 0) \}$ φ

$z := x - \text{sqrt}(y);$ S

$\{ N \}$ ψ

Welche ist die stärkste Nachbedingung N ?

Welche ist die schwächste Vorbedingung für $\psi \Leftrightarrow (z > 0)$?

Verifikation von Zuweisungen

Die Zuweisungsvariable darf auch im zu berechnenden Ausdruck vorkommen:

$\{ (x > 0) \}$ φ

$x := x - \text{sqrt}(x);$ S

$\{ N \}$ ψ

Welche ist die stärkste Nachbedingung N ?

Welche ist die schwächste Vorbedingung für $\psi \Leftrightarrow (x > 0)$?

Verifikation von Zuweisungen

Der allgemeine Fall einer Zuweisung:

$\{ V(x) \}$ φ

$x := f(x);$ S

$\{ N(x) \}$ ψ

Verfahren zur Berechnung der schwächsten Vorbedingung zu gegebener Nachbedingung:

- Ersetze in $N(x)$ jedes x durch $f(x)$ \rightarrow Das Ergebnis ist die schwächste Vorbedingung.

Verfahren zur Berechnung der stärksten Nachbedingung zu gegebener Vorbedingung:

- Ersetze in $V(x)$ jedes x durch $f(x)$ \rightarrow Das Ergebnis ist die stärkste Nachbedingung.

Beim nächsten Mal:

**Programmentwicklung und –verifikation:
Verbundanweisungen
Verzweigungen**