Einführung in die Theoretische Informatik

Christian Dalvit Manuel Eberl
Samuel Frontull **Cezary Kaliszyk** Daniel Ranalter

Wintersemester 2022/23

universität
innsbruck

Programmverifikation

Wintersemester 2022/23

Wozu Programmverifikation

Wozu Programmverifikation

- Ariane-5
- Fehler in der Datenkonvertierung
- USD 370 Millionen

Wozu Programmverifikation

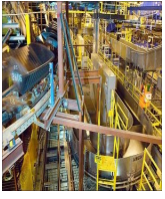
- Ariane-5
- Fehler in der Datenkonvertierung
- USD 370 Millionen

- Intel Pentium FDIV-Bug
- Falsche Berechnungen
- USD 475 Millionen

Wozu Programmverifikation

- Ariane-5
- Fehler in der Datenkonvertierung
- USD 370 Millionen

- Intel Pentium FDIV-Bug
- Falsche Berechnungen
- USD 475 Millionen



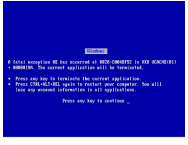
- Gepäckverteilung (Denver)
- Desaster
- USD 560 Millionen

Wozu Programmverifikation

- Ariane-5
- Fehler in der Datenkonvertierung
- USD 370 Millionen

- Intel Pentium FDIV-Bug
- Falsche Berechnungen
- USD 475 Millionen

- Gepäckverteilung (Denver)
- Desaster
- USD 560 Millionen



- Blue Screen of Death
- ziemlich lästig

Begutachtung

- Der Code wird von ähnlich qualifizierten Programmierern kontrolliert
- subtile Fehler werden leicht übersehen

Begutachtung

- Der Code wird von ähnlich qualifizierten Programmierern kontrolliert
- subtile Fehler werden leicht übersehen

Testen

- dynamische Technik, bei der das Programm ausgeführt wird
- Wie wird die richtige Testumgebung geschaffen?

Begutachtung

- Der Code wird von ähnlich qualifizierten Programmierern kontrolliert
- subtile Fehler werden leicht übersehen

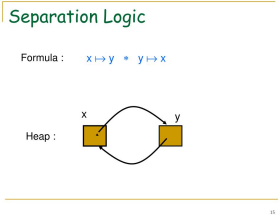
Testen

- dynamische Technik, bei der das Programm ausgeführt wird
- Wie wird die richtige Testumgebung geschaffen?

Formal Methoden

- erlauben die frühe Integration der Verifikation in die Softwareentwicklung
- sind effizienter als andere Methoden (höhere Erkennensrate von Fehlern)
- sind (im Besonderen wenn automatisierbar) schneller anwendbar

Beispiele formalen Softwareverifikation



universität innsbruck

Verifikation nach Hoare

Wintersemester 2022/23

Prädikatenlogik (informell)

- Die Prädikatenlogik ist eine Logik, deren Ausdruckskraft über die der Aussagenlogik weit hinausgeht
- Die wichtigste Erweiterung sind Prädikatensymbole und Quantoren
- Prädikatensymbole erlauben es uns, über Elemente einer Menge Aussagen zu treffen

Prädikatenlogik (informell)

- Die Prädikatenlogik ist eine Logik, deren Ausdruckskraft über die der Aussagenlogik weit hinausgeht
- Die wichtigste Erweiterung sind Prädikatensymbole und Quantoren
- Prädikatensymbole erlauben es uns, über Elemente einer Menge Aussagen zu treffen

Sprache einer Prädikatenlogik

Eine Prädikatenlogik durch eine Sprache beschrieben, diese Sprache enthält:

1. Funktionssymbole und Prädikatensymbole; Variablen
2. $\underbrace{=}_{\text{Gleichheit}}$, $\underbrace{\neg, \wedge, \vee, \rightarrow}_{\text{Junktoren}}$, $\underbrace{\forall, \exists}_{\text{Quantoren}}$

Beispiel

- Sei 7 eine Konstante und ist_prim ein Prädikatensymbol
- Wir schreiben ist_prim(7), um auszudrücken, dass 7 eine Primzahl

Beispiel

- Sei 7 eine Konstante und ist_prim ein Prädikatensymbol
- Wir schreiben ist_prim(7), um auszudrücken, dass 7 eine Primzahl

Definition

Ein Ausdruck der mit Hilfe von Variablen und Funktionssymbolen gebildet wird, heißt Term

Beispiel

- Sei 7 eine Konstante und ist_prim ein Prädikatensymbol
- Wir schreiben ist_prim(7), um auszudrücken, dass 7 eine Primzahl

Definition

Ein Ausdruck der mit Hilfe von Variablen und Funktionssymbolen gebildet wird, heißt Term

Definition

1. Sei P ein Prädikatensymbol
2. Seien $t_1, \ldots, t_n$ Terme

Beispiel

- Sei 7 eine Konstante und ist_prim ein Prädikatensymbol
- Wir schreiben ist_prim(7), um auszudrücken, dass 7 eine Primzahl

Definition

Ein Ausdruck der mit Hilfe von Variablen und Funktionssymbolen gebildet wird, heißt Term

Definition

1. Sei P ein Prädikatensymbol
2. Seien $t_1, \ldots, t_n$ Terme

Dann nennen wir die Ausdrücke $P(t_1, \ldots, t_n)$ und $t_1 = t_2$ Atome oder atomare Formel

Definition (Zusicherungen)

Wir definieren Zusicherungen induktiv:

1. Atome sind Zusicherungen
2. Wenn A und B Zusicherungen sind, dann sind auch die folgenden Ausdrücke, Zusicherungen:

$$\neg A \quad (A \wedge B) \quad (A \vee B) \quad (A \to B)$$

Definition (Zusicherungen)

Wir definieren Zusicherungen induktiv:

1. Atome sind Zusicherungen
2. Wenn A und B Zusicherungen sind, dann sind auch die folgenden Ausdrücke, Zusicherungen:

$$\neg A \quad (A \wedge B) \quad (A \vee B) \quad (A \to B)$$

Konvention

Zusicherungen werden Formeln genannt

Definition (Zusicherungen)

Wir definieren Zusicherungen induktiv:

1. Atome sind Zusicherungen
2. Wenn A und B Zusicherungen sind, dann sind auch die folgenden Ausdrücke, Zusicherungen:

$$\neg A \quad (A \wedge B) \quad (A \vee B) \quad (A \to B)$$

Konvention

Zusicherungen werden Formeln genannt

Definition

Interpretationen $\mathcal{I}$ werden verwendet, um den Ausdrücken der Prädikatenlogik eine Bedeutung zu geben

Beispiel

- Wir betrachten die Konstante 7 und das Prädikat ist_prim
- Interpretation $\mathcal{I}$ legt fest, dass 7 als die Zahl sieben zu verstehen ist
- $\mathcal{I}$ legt fest, dass das Atom ist_prim(n) genau dann wahr ist, wenn n eine Primzahl

Beispiel

- Wir betrachten die Konstante 7 und das Prädikat ist_prim
- Interpretation $\mathcal{I}$ legt fest, dass 7 als die Zahl sieben zu verstehen ist
- $\mathcal{I}$ legt fest, dass das Atom ist_prim(n) genau dann wahr ist, wenn n eine Primzahl

Beobachtung

1. Mit Hilfe von Interpretationen wird der Wahrheitsgehalt von Atomen bestimmt
2. Ist die Wahrheit von Atomen in $\mathcal{I}$ definiert, wird die Wahrheit einer beliebigen Formel durch die Bedeutung der Junktoren bestimmt

Beispiel

- Wir betrachten die Konstante 7 und das Prädikat ist_prim
- Interpretation $\mathcal{I}$ legt fest, dass 7 als die Zahl sieben zu verstehen ist
- $\mathcal{I}$ legt fest, dass das Atom ist_prim(n) genau dann wahr ist, wenn n eine Primzahl

Beobachtung

1. Mit Hilfe von Interpretationen wird der Wahrheitsgehalt von Atomen bestimmt
2. Ist die Wahrheit von Atomen in $\mathcal{I}$ definiert, wird die Wahrheit einer beliebigen Formel durch die Bedeutung der Junktoren bestimmt

Beispiel

Die Formel ist_prim(x) $\wedge$ $x = 7$ bedeutet, dass x die Primzahl 7 ist

Definition

Sei $\mathcal{I}$ eine Interpretation und F eine Formel, wir schreiben $\mathcal{I} \models F$, wenn die Formel F in der Interpretation $\mathcal{I}$ wahr ist

Definition

Sei $\mathcal{I}$ eine Interpretation und F eine Formel, wir schreiben $\mathcal{I} \models F$, wenn die Formel F in der Interpretation $\mathcal{I}$ wahr ist

Beispiel

Wenn x die Primzahl 7 ist, dann gilt $\mathcal{I} \models \text{ist\_prim}(x) \wedge x = 7$

Definition

Sei $\mathcal{I}$ eine Interpretation und F eine Formel, wir schreiben $\mathcal{I} \models F$, wenn die Formel F in der Interpretation $\mathcal{I}$ wahr ist

Beispiel

Wenn x die Primzahl 7 ist, dann gilt $\mathcal{I} \models \text{ist\_prim}(x) \wedge x = 7$

Definition

Die Konsequenzrelation $A \models B$ gilt, gdw. für alle Interpretationen $\mathcal{I}$:

$$\mathcal{I} \models A \text{ impliziert } \mathcal{I} \models B$$

Definition

Sei $\mathcal{I}$ eine Interpretation und F eine Formel, wir schreiben $\mathcal{I} \models F$, wenn die Formel F in der Interpretation $\mathcal{I}$ wahr ist

Beispiel

Wenn x die Primzahl 7 ist, dann gilt $\mathcal{I} \models \text{ist\_prim}(x) \wedge x = 7$

Definition

Die Konsequenzrelation $A \models B$ gilt, gdw. für alle Interpretationen $\mathcal{I}$:

$$\mathcal{I} \models A \text{ impliziert } \mathcal{I} \models B$$

Beispiel

Seien $x_1 > 4$ und $x_1 + 1 > 5$ Atome, es gilt:

$$x_1 > 4 \models x_1 + 1 > 5$$

Hoare-Tripel

Definition

- Sei P ein while-Programm (ein Programm einer Registermaschine)
- Seien Q und R Zusicherungen
- Ein Hoare-Tripel ist wie folgt definiert:

$$\{Q\}\ P\ \{R\}$$

- Q wird Vorbedingung
- R wird Nachbedingung genannt

Hoare-Tripel

Definition

- Sei P ein while-Programm (ein Programm einer Registermaschine)
- Seien Q und R Zusicherungen
- Ein Hoare-Tripel ist wie folgt definiert:

$$\{Q\}\ P\ \{R\}$$

- Q wird Vorbedingung
- R wird Nachbedingung genannt

Beispiel

Seien $x_1 > 4$, $x_1 > 5$ Zusicherungen und $x_1 := x_1 + 1$ ein Programm,
dann ist $\{x_1 > 4\}\ x_1 := x_1 + 1\ \{x_1 > 5\}$ ein Hoare-Tripel

Definition

- Ein Hoare-Tripel $\{Q\}\ P\ \{R\}$ ist wahr, wenn
  1. Q vor der Ausführung von P gilt
  2. R nach der Ausführung von P gilt
  3. unter der Voraussetzung, dass P terminiert

Definition

- Ein Hoare-Tripel $\{Q\}\ P\ \{R\}$ ist wahr, wenn
  1. Q vor der Ausführung von P gilt
  2. R nach der Ausführung von P gilt
  3. unter der Voraussetzung, dass P terminiert
- Wenn $\{Q\}\ P\ \{R\}$ wahr, dann ist P korrekt in Bezug auf Q und R
- Dann sagen wir auch P ist partiell korrekt

Definition

- Ein Hoare-Tripel $\{Q\}\ P\ \{R\}$ ist wahr, wenn
  1. Q vor der Ausführung von P gilt
  2. R nach der Ausführung von P gilt
  3. unter der Voraussetzung, dass P terminiert
- Wenn $\{Q\}\ P\ \{R\}$ wahr, dann ist P korrekt in Bezug auf Q und R
- Dann sagen wir auch P ist partiell korrekt
- Das Programm P ist total korrekt, wenn es partiell korrekt ist und terminiert

Definition

- Ein Hoare-Tripel $\{Q\}\ P\ \{R\}$ ist wahr, wenn
  1. Q vor der Ausführung von P gilt
  2. R nach der Ausführung von P gilt
  3. unter der Voraussetzung, dass P terminiert
- Wenn $\{Q\}\ P\ \{R\}$ wahr, dann ist P korrekt in Bezug auf Q und R
- Dann sagen wir auch P ist partiell korrekt
- Das Programm P ist total korrekt, wenn es partiell korrekt ist und terminiert

Beispiel

Die folgenden Hoare-Tripel

$$\{x_1 > 4\}\ x_1 := x_1 + 1\ \{x_1 > 5\} \qquad \{x_2 = 0\}\ x_2 := x_2 - 1\ \{x_2 = 0\}$$

sind wahr und die jeweiligen Programme total korrekt

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$[z] \quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}}$$

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$\text{[z]} \quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad \text{[a]} \quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$\text{[z]} \quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad \text{[a]} \quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

$$\text{[s]} \quad \frac{\{Q\}\ P_1\ \{R\} \quad \{R\}\ P_2\ \{S\}}{\{Q\}\ P_1; P_2\ \{S\}}$$

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$\text{[z]}\quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad \text{[a]}\quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

$$\text{[s]}\quad \frac{\{Q\}\ P_1\ \{R\} \quad \{R\}\ P_2\ \{S\}}{\{Q\}\ P_1; P_2\ \{S\}} \qquad \text{[w]}\quad \frac{\{I \wedge B\}\ P\ \{I\}}{\{I\}\ \text{while } B \text{ do } P \text{ end } \{I \wedge \neg B\}}$$

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$\text{[z]}\quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad \text{[a]}\quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

$$\text{[s]}\quad \frac{\{Q\}\ P_1\ \{R\} \quad \{R\}\ P_2\ \{S\}}{\{Q\}\ P_1; P_2\ \{S\}} \qquad \text{[w]}\quad \frac{\{I \wedge B\}\ P\ \{I\}}{\{I\}\ \text{while } B \text{ do } P \text{ end } \{I \wedge \neg B\}}$$

Ist ein Hoare-Tripel in diesem Kalkül ableitbar, dann ist es wahr

Hoare-Kalkül

Definition

Die Regeln des Hoare-Kalkül sind wie folgt definiert:

$$\text{[z]}\quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad \text{[a]}\quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

$$\text{[s]}\quad \frac{\{Q\}\ P_1\ \{R\} \quad \{R\}\ P_2\ \{S\}}{\{Q\}\ P_1; P_2\ \{S\}} \qquad \text{[w]}\quad \frac{\{I \wedge B\}\ P\ \{I\}}{\{I\}\ \text{while } B \text{ do } P \text{ end } \{I \wedge \neg B\}}$$

Ist ein Hoare-Tripel in diesem Kalkül ableitbar, dann ist es wahr

Beispiel

$$\frac{\dfrac{}{\{x_1 + 1 > 5\}\ x_1 := x_1 + 1\ \{x_1 > 5\}}\ \text{[z]}}{\{x_1 > 4\}\ x_1 := x_1 + 1\ \{x_1 > 5\}}\ \text{[a]}, x_1 > 4 \models x_1 + 1 > 5$$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

$$\dfrac{\dfrac{\dfrac{}{\{x_i - 1 \geqslant 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[z]}{\{x_i \geqslant 0 \wedge x_i \neq 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[a]}{\{x_i \geqslant 0\}\ P\ \{x_i \geqslant 0 \wedge x_i = 0\}}\,[w]}{\{x_i \geqslant 0\}\ P\ \{x_i = 0\}}\,[a]$$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

$$\dfrac{\dfrac{\dfrac{}{\{x_i - 1 \geqslant 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[\text{z}]}{\{x_i \geqslant 0 \wedge x_i \neq 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[\text{a}]}{\{x_i \geqslant 0\}\ P\ \{x_i \geqslant 0 \wedge x_i = 0\}}\,[\text{w}]}{\{x_i \geqslant 0\}\ P\ \{x_i = 0\}}\,[\text{a}]$$

wir verwenden:

1. $x_i \geqslant 0 \wedge x_i = 0 \models x_i = 0$
2. die Schleifeninvariante $x_i \geqslant 0$
3. $x_i \geqslant 0 \wedge x_i \neq 0 \models x_i - 1 \geqslant 0$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

$$\dfrac{\dfrac{\dfrac{}{\{x_i - 1 \geqslant 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[z]}{\{x_i \geqslant 0 \wedge x_i \neq 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[a]}{\{x_i \geqslant 0\}\ P\ \{x_i \geqslant 0 \wedge x_i = 0\}}\,[w]}{\{x_i \geqslant 0\}\ P\ \{x_i = 0\}}\,[a]$$

wir verwenden:

1. $x_i \geqslant 0 \wedge x_i = 0 \models x_i = 0$
2. die Schleifeninvariante $x_i \geqslant 0$
3. $x_i \geqslant 0 \wedge x_i \neq 0 \models x_i - 1 \geqslant 0$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

$$\dfrac{\dfrac{\dfrac{}{\{x_i - 1 \geqslant 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[z]}{\{x_i \geqslant 0 \wedge x_i \neq 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[a]}{\{x_i \geqslant 0\}\ P\ \{x_i \geqslant 0 \wedge x_i = 0\}}\,[w]}{\{x_i \geqslant 0\}\ P\ \{x_i = 0\}}\,[a]$$

wir verwenden:

1. $x_i \geqslant 0 \wedge x_i = 0 \models x_i = 0$
2. die Schleifeninvariante $x_i \geqslant 0$
3. $x_i \geqslant 0 \wedge x_i \neq 0 \models x_i - 1 \geqslant 0$

Beispiel

Wir betrachten das folgende einfache while-Programm P:

$$\begin{array}{l}\text{while } x_i \neq 0 \text{ do}\\ \quad x_i := x_i - 1\\ \text{end}\end{array}$$

und zeigen $\{x_i \geqslant 0\}\ P\ \{x_i = 0\}$

$$\dfrac{\dfrac{\dfrac{}{\{x_i - 1 \geqslant 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[z]}{\{x_i \geqslant 0 \wedge x_i \neq 0\}\ x_i := x_i - 1\ \{x_i \geqslant 0\}}\,[a]}{\{x_i \geqslant 0\}\ P\ \{x_i \geqslant 0 \wedge x_i = 0\}}\,[w]$$
$$\{x_i \geqslant 0\}\ P\ \{x_i = 0\}\ [a]$$

wir verwenden:

1. $x_i \geqslant 0 \wedge x_i = 0 \models x_i = 0$
2. die Schleifeninvariante $x_i \geqslant 0$
3. $x_i \geqslant 0 \wedge x_i \neq 0 \models x_i - 1 \geqslant 0$

universität innsbruck

Prüfungsorganisation und -vorbereitung

Wintersemester 2022/23

VO Exams

- Registration Necessary, with official deadlines
- ID controlled, without registration and ID no entry
- Content checked until today
- Closed book. Natural deduction rules will be reminded, rest you should know.

SL Exams

- Only 2 dates. Registration needed
  (but emails until previous day ok)
- Closed book

(Possible) Content

Induction, Natural Deduction, Grammars, Automata/TM, Yes/no

universität innsbruck

Prüfungsvorbereitung

Wintersemester 2022/23

Vielen Dank für die Aufmerksamkeit!