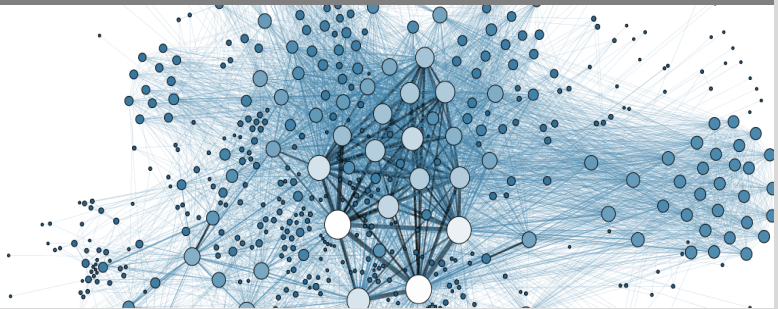
Karlsruher Institut für Technologie

SDQ Software Design and Quality

Grundbegriffe der Informatik
Tutorium 38

Algorithmen

Patrick Fetzer, uxkln@student.kit.edu | 13.12.2018



Algorithmen

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmen

- Es existiert eine **endliche** Beschreibung

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmen

- Es existiert eine **endliche** Beschreibung
- Die Anweisungen sind elementar

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmen

- Es existiert eine **endliche** Beschreibung
- Die Anweisungen sind elementar
- Es wird zu einer beliebig großen, aber **endlichen** Eingabe eine **endliche** Ausgabe berechnet

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmen

- Es existiert eine **endliche** Beschreibung
- Die Anweisungen sind elementar
- Es wird zu einer beliebig großen, aber **endlichen** Eingabe eine **endliche** Ausgabe berechnet
- Es finden **endlich** viele Schritte statt (der Algorithmus terminiert)

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmen

- Es existiert eine **endliche** Beschreibung
- Die Anweisungen sind elementar
- Es wird zu einer beliebig großen, aber **endlichen** Eingabe eine **endliche** Ausgabe berechnet
- Es finden **endlich** viele Schritte statt (der Algorithmus terminiert)
- Deterministisch (bei mehrmaliger Ausführung kommt immer das selbe raus)

Patrick Fetzer,
uxkln@student.kit.edu

Hier verwendeter Pseudocode

Patrick Fetzer,
uxkln@student.kit.edu

Hier verwendeter Pseudocode

- Zuweisungssymbol $\longleftarrow$

Patrick Fetzer,
uxkln@student.kit.edu

Hier verwendeter Pseudocode

- Zuweisungssymbol $\longleftarrow$
- Schlüsselwörter für Verzweigungen **if, then, else, fi**

Patrick Fetzer,
uxkln@student.kit.edu

Hier verwendeter Pseudocode

- Zuweisungssymbol $\leftarrow$
- Schlüsselwörter für Verzweigungen **if, then, else, fi**
- Schlüsselwörter für Schleifen **while, do, od, for, to**

Patrick Fetzer,
uxkln@student.kit.edu

Hier verwendeter Pseudocode

- Zuweisungssymbol $\leftarrow$
- Schlüsselwörter für Verzweigungen **if, then, else, fi**
- Schlüsselwörter für Schleifen **while, do, od, for, to**
- Symbole für Konstanten, Funktionen und Relationen

Patrick Fetzer,
uxkln@student.kit.edu

Eine **if**-Verzweigung

```
1 if x < y then
2     s ← x
3 else
4     s ← y
5 fi
```

Patrick Fetzer,
uxkln@student.kit.edu

Eine **if**-Verzweigung

```
1 if x < y then
2     s ← x
3 else
4     s ← y
5 fi
```

Eine **while**-Schleife

```
1 while x > 0 do
2     x ← x div 2
3     s ← s + x
4 od
```

Eine **if**-Verzweigung

```
1 if x < y then
2     s ← x
3 else
4     s ← y
5 fi
```

Eine **while**-Schleife

```
1 while x > 0 do
2     x ← x div 2
3     s ← s + x
4 od
```

Eine **for**-Schleife

```
1 for i ← 1 to n do
2     s ← s + i
3 od
```

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen, im Modul Algorithmen I

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen, im Modul Algorithmen I
- Laufzeitanalyse von Algorithmen

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen, im Modul Algorithmen I
- Laufzeitanalyse von Algorithmen, später.

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen, im Modul Algorithmen I
- Laufzeitanalyse von Algorithmen, später.
- Korrektheitsbeweise

Patrick Fetzer,
uxkln@student.kit.edu

Was kann man mit Algorithmen machen?

- Komplexe Algorithmen mit Pseudocode definieren zu Sortierung, Graphen, Datenstrukturen, im Modul Algorithmen I
- Laufzeitanalyse von Algorithmen, später.
- Korrektheitsbeweise, jetzt.

Korrektheitsbeweise

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Korrektheitsbeweise

Wie findet man heraus, ob ein Algorithmus korrekt funktioniert?

Patrick Fetzer,
uxkln@student.kit.edu

Korrektheitsbeweise

Wie findet man heraus, ob ein Algorithmus korrekt funktioniert?

- Durch den Beweis von Zusicherungen, die an bestimmten Stellen des Algorithmus gelten.

Patrick Fetzer,
uxkln@student.kit.edu

Korrektheitsbeweise

Wie findet man heraus, ob ein Algorithmus korrekt funktioniert?

- Durch den Beweis von Zusicherungen, die an bestimmten Stellen des Algorithmus gelten.

Was sind Zusicherungen?

Patrick Fetzer,
uxkln@student.kit.edu

Korrektheitsbeweise

Wie findet man heraus, ob ein Algorithmus korrekt funktioniert?

- Durch den Beweis von Zusicherungen, die an bestimmten Stellen des Algorithmus gelten.

Was sind Zusicherungen?

- prädikatenlogische Formeln, die Aussagen über (Zusammenhänge zwischen) Variablen machen

Das Hoare-Tripel

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode
- P und Q sind Zusicherungen

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode
- P und Q sind Zusicherungen

- P nennt man Vorbedingung, Q Nachbedingung

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode
- P und Q sind Zusicherungen

- P nennt man Vorbedingung, Q Nachbedingung
- Prädikatenlogische Formeln

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode
- P und Q sind Zusicherungen

- P nennt man Vorbedingung, Q Nachbedingung
- Prädikatenlogische Formeln
- Beispiel (Vorausblick): $\{x \dot{=} 1\}x \leftarrow x + 1\{x \dot{=} 2\}$

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Definition

$\{P\}S\{Q\}$ heißt Hoare-Tripel. Dabei gilt:

- S ist ein Programmstück im Pseudocode
- P und Q sind Zusicherungen

- P nennt man Vorbedingung, Q Nachbedingung
- Prädikatenlogische Formeln
- Beispiel (Vorausblick): $\{x \dot{=} 1\}x \leftarrow x + 1\{x \dot{=} 2\}$
- Meistens in jeder Zeile nur eine Zeile Code oder ein Zusicherungsblock

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Gültigkeit von Hoare-Tripeln

$\{P\}S\{Q\}$ ist gültig, wenn für jede gültige Interpretation (D, I) und Variablenbelegung β gilt:

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Gültigkeit von Hoare-Tripeln

$\{P\}S\{Q\}$ ist gültig, wenn für jede gültige Interpretation (D, I) und Variablenbelegung β gilt:
Aus

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Gültigkeit von Hoare-Tripeln

$\{P\}S\{Q\}$ ist gültig, wenn für jede gültige Interpretation (D, I) und Variablenbelegung β gilt:
Aus

- $val_{D,I,\beta}(P) = w$

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Gültigkeit von Hoare-Tripeln

$\{P\}S\{Q\}$ ist gültig, wenn für jede gültige Interpretation (D, I) und Variablenbelegung β gilt:
Aus

- $val_{D,I,\beta}(P) = w$
- β' ist Variablenbelegung nach Ausführung von S

Patrick Fetzer,
uxkln@student.kit.edu

Das Hoare-Tripel

Gültigkeit von Hoare-Tripeln

$\{P\}S\{Q\}$ ist gültig, wenn für jede gültige Interpretation (D, I) und Variablenbelegung β gilt:
Aus

- $val_{D,I,\beta}(P) = w$
- β' ist Variablenbelegung nach Ausführung von S

folgt $val_{D,I,\beta'}(Q) = w$

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \longleftarrow E$ eine Zuweisung

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \longleftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \longleftarrow E$ und

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \longleftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \longleftarrow E$ und
- $\sigma_{\{x/E\}}$ kollisionsfrei für Q

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \leftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \leftarrow E$ und
- $\sigma_{\{x/E\}}$ kollisionsfrei für Q

Dann ist $\sigma_{\{x/E\}}(Q)x \leftarrow E\{Q\}$ ein gültiges Hoare-Tripel

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \leftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \leftarrow E$ und
- $\sigma_{\{x/E\}}$ kollisionsfrei für Q

Dann ist $\sigma_{\{x/E\}}(Q)x \leftarrow E\{Q\}$ ein gültiges Hoare-Tripel

Bemerkung

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \leftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \leftarrow E$ und
- $\sigma_{\{x/E\}}$ kollisionsfrei für Q

Dann ist $\sigma_{\{x/E\}}(Q)x \leftarrow E\{Q\}$ ein gültiges Hoare-Tripel

Bemerkung

- $\sigma_{\{x/E\}}$ ist die Substitution von x mit E

Patrick Fetzer,
uxkln@student.kit.edu

Zuweisung

Axiom HT-A

- Sei $x \leftarrow E$ eine Zuweisung
- Q eine Nachbedingung von $x \leftarrow E$ und
- $\sigma_{\{x/E\}}$ kollisionsfrei für Q

Dann ist $\sigma_{\{x/E\}}(Q)x \leftarrow E\{Q\}$ ein gültiges Hoare-Tripel

Bemerkung

- $\sigma_{\{x/E\}}$ ist die Substitution von x mit E
- Bei Anwendung der Regel rückwärts vorgehen

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Betrachte die Zuweisung

$x \leftarrow x + 1$

und die Nachbedingung

$\{x \doteq 1\}$

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Betrachte die Zuweisung

$x \leftarrow x + 1$

und die Nachbedingung

$\{x \doteq 1\}$

Nach HT-A gilt

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Betrachte die Zuweisung

$x \leftarrow x + 1$

und die Nachbedingung

$\{x \doteq 1\}$

Nach HT-A gilt

$\{x + 1 \doteq 1\}\ x \leftarrow x + 1\ \{x \doteq 1\}$ ist ein gültiges Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-E

- Verstärkung der Vorbedingung
- Abschwächung der Nachbedingung

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-E

- Verstärkung der Vorbedingung
- Abschwächung der Nachbedingung

HT-E

Wenn $\{P\}S\{Q\}$ ein gültiges Hoare-Tripel ist und $P' \vdash P$ und $Q \vdash Q'$ gelten, dann folgt:

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-E

- Verstärkung der Vorbedingung
- Abschwächung der Nachbedingung

HT-E

Wenn $\{P\}S\{Q\}$ ein gültiges Hoare-Tripel ist und $P' \vdash P$ und $Q \vdash Q'$ gelten, dann folgt:
$\{P'\}S\{Q'\}$ ist ein gültiges Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-E

- Verstärkung der Vorbedingung
- Abschwächung der Nachbedingung

HT-E

Wenn $\{P\}S\{Q\}$ ein gültiges Hoare-Tripel ist und $P' \vdash P$ und $Q \vdash Q'$ gelten, dann folgt:
$\{P'\}S\{Q'\}$ ist ein gültiges Hoare-Tripel.

Bemerkung
$B \vdash A :\Leftrightarrow$ Aussage A ist syntaktisch aus Aussage B ableitbar

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es sei $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ ein gültiges Hoare-Tripel.
Es gilt $\{(y > 4)\} \vdash \{(y > 3)\}$ und $\{(x > 1)\} \vdash \{(x > 0)\}$.
Also folgt nach HT-E:

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es sei $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ ein gültiges Hoare-Tripel.
Es gilt $\{(y > 4)\} \vdash \{(y > 3)\}$ und $\{(x > 1)\} \vdash \{(x > 0)\}$.
Also folgt nach HT-E:
$\{y > 4\}\ x \leftarrow y - 1\ \{x > 0\}$ ist ein gültiges Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es sei $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ ein gültiges Hoare-Tripel.
Es gilt $\{(y > 4)\} \vdash \{(y > 3)\}$ und $\{(x > 1)\} \vdash \{(x > 0)\}$.
Also folgt nach HT-E:
$\{y > 4\}\ x \leftarrow y - 1\ \{x > 0\}$ ist ein gültiges Hoare-Tripel.

Bemerkung

Es müssen sich nicht unbedingt beide Bedingungen ändern!
Aus $\{(y > 3)\} \vdash \{(y > 3)\}$ und $\{(x > 1)\} \vdash \{(x > 0)\}$
folgt nach HT-E auch
$\{y > 3\}\ x \leftarrow y - 1\ \{x > 0\}$ ist ein gültiges Hoare-Tripel.

Ableitungsregeln: HT-S

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-S

Hintereinanderausführung von durch Hoare-Triple bewiesene Code Segmente sind selbst gültig.

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-S

Hintereinanderausführung von durch Hoare-Triple bewiesene Code Segmente sind selbst gültig.

HT-S

Wenn $\{P\}S_1\{Q\}$ und $\{Q\}S_2\{R\}$ gültige Hoare-Tripel sind

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-S

Hintereinanderausführung von durch Hoare-Triple bewiesene Code Segmente sind selbst gültig.

HT-S

Wenn $\{P\}S_1\{Q\}$ und $\{Q\}S_2\{R\}$ gültige Hoare-Tripel sind, dann folgt:

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-S

Hintereinanderausführung von durch Hoare-Triple bewiesene Code Segmente sind selbst gültig.

HT-S

Wenn $\{P\}S_1\{Q\}$ und $\{Q\}S_2\{R\}$ gültige Hoare-Tripel sind, dann folgt:$\{P\}S_1; S_2\{R\}$ ist ein gültiges Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Ableitungsregeln: HT-S

Hintereinanderausführung von durch Hoare-Triple bewiesene Code Segmente sind selbst gültig.

HT-S

Wenn $\{P\}S_1\{Q\}$ und $\{Q\}S_2\{R\}$ gültige Hoare-Tripel sind, dann folgt:$\{P\}S_1; S_2\{R\}$ ist ein gültiges Hoare-Tripel.

Bemerkung

";" trennt hier zwei Programmstücke

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es seien $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ und $\{x > 1\}\ z \leftarrow x - 1\ \{z > -1\}$ gültige Hoare-Tripel.

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es seien $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ und $\{x > 1\}\ z \leftarrow x - 1\ \{z > -1\}$ gültige Hoare-Tripel.
Dann folgt nach HT-S:

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

Angenommen es seien $\{y > 3\}\ x \leftarrow y - 1\ \{x > 1\}$ und
$\{x > 1\}\ z \leftarrow x - 1\ \{z > -1\}$ gültige Hoare-Tripel.
Dann folgt nach HT-S:
$\{y > 3\}\ x \leftarrow y - 1; z \leftarrow x - 1\ \{z > -1\}$ ein gültiges Hoare-Tripel.

Bedingte Anweisungen

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Bedingte Anweisungen

HT-I

Wenn $\{P \wedge B\} S_1 \{Q\}$ und $\{P \wedge \neg B\} S_2 \{Q\}$ gültige Hoare-Tripel sind

Patrick Fetzer,
uxkln@student.kit.edu

Bedingte Anweisungen

HT-I

Wenn $\{P \wedge B\} S_1 \{Q\}$ und $\{P \wedge \neg B\} S_2 \{Q\}$ gültige Hoare-Tripel sind, dann folgt:

$\{P\}$
if B **then** S_1
else S_2
fi
$\{Q\}$

ist ein gültiges Hoare-Tripel.

Beispiel

Patrick Fetzer,
uxkln@student.kit.edu

$\{ x = a \wedge y = b \}$

if $x > y$

then

$\{ \dots \}$

$z \leftarrow y$

$\{ \dots \}$

else

$\{ \dots \}$

$z \leftarrow x$

$\{ \dots \}$

fi

$\{ z = \min(a, b) \}$

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

$\{ x = a \wedge y = b \}$

if $x > y$

then

$\{ x = a \wedge y = b \wedge x > y \}$

$\{ y = \min(a,b) \}$

$z \leftarrow y$

$\{ z = \min(a,b) \}$

else

$\{ x = a \wedge y = b \wedge \neg(x > y) \}$

$\{ x = \min(a,b) \}$

$z \leftarrow x$

$\{ z = \min(a,b) \}$

fi

$\{ z = \min(a,b) \}$

Patrick Fetzer,
uxkln@student.kit.edu

Schleifen

HT-W

Wenn $\{I \wedge B\}S\{I\}$ ein gültiges Hoare-Tripel ist

Patrick Fetzer,
uxkln@student.kit.edu

Schleifen

HT-W

Wenn $\{I \wedge B\}S\{I\}$ ein gültiges Hoare-Tripel ist, dann folgt:

Patrick Fetzer,
uxkln@student.kit.edu

Schleifen

HT-W

Wenn $\{I \wedge B\}S\{I\}$ ein gültiges Hoare-Tripel ist, dann folgt:

$\{I\}$
while B **do** S
od
$\{I \wedge \neg B\}$

ist ein gültiges Hoare-Tripel.

Schleifeninvariante

Patrick Fetzer,
uxkln@student.kit.edu

Patrick Fetzer,
uxkln@student.kit.edu

Schleifeninvariante

- Eine spezielle Zusicherung

Patrick Fetzer,
uxkln@student.kit.edu

Schleifeninvariante

- Eine spezielle Zusicherung
- Schleifeninvarianten müssen **vor**, **während** und **nach** jedem Schleifendurchlauf gelten

Patrick Fetzer,
uxkln@student.kit.edu

Schleifeninvariante

- Eine spezielle Zusicherung
- Schleifeninvarianten müssen **vor**, **während** und **nach** jedem Schleifendurchlauf gelten
- Garantiert, dass die Schleife nicht während einem beliebigen Durchlauf “kaputt” geht.

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

$\{ x = a \wedge y = b \}$

$\{ \dots \}$

while $y \neq 0$

do

$\{ \dots \}$

$y \leftarrow y - 1$

$\{ \dots \}$

$x \leftarrow x + 1$

$\{ \dots \}$

od

$\{ \dots \}$

$\{ x = a + b \}$

Patrick Fetzer,
uxkln@student.kit.edu

Beispiel

$\{ x = a \wedge y = b \}$

$\{ x + y = a + b \}$

while $y \neq 0$

do

$\{ x + y = a + b \wedge y \neq 0 \}$

$\{ x + 1 + y - 1 = a + b \}$

$y \leftarrow y - 1$

$\{ x + 1 + y = a + b \}$

$x \leftarrow x + 1$

$\{ x + y = a + b \}$

od

$\{ x + y = a + b \wedge \neg(y \neq 0) \}$

$\{ x = a + b \}$

Patrick Fetzer,
uxkln@student.kit.edu

Bestimmt für den folgenden Algorithmus was er berechnet und eine Schleifeninvariante:

$\{a \in \mathbb{N}_+, b \in \mathbb{N}_+\}$
$X_0 \leftarrow a$
$Y_0 \leftarrow b$
$P_0 \leftarrow 1$
$\alpha_0 \leftarrow X_0 mod 2$
$n \leftarrow 1 + \lceil log_2 a \rceil$
$fori \leftarrow 0$ to $n - 1 do$
    $P_{i+1} \leftarrow P_i * {Y_i}^{\alpha_i}$
    $X_{i+1} \leftarrow X_i div 2$
    $Y_{i+1} \leftarrow {Y_i}^2$
    $\alpha_{i+1} \leftarrow X_{i+1} mod 2$
od

Tipp: Beispieleingabe $a = 5$ und $b = 2$

Patrick Fetzer,
uxkln@student.kit.edu