Grundbegriffe der Informatik

Kapitel 17: Der Begriff des Algorithmus (einige grundlegende Aspekte)

Mattias Ulbrich
(basierend auf Folien von Thomas Worsch)

KIT · Institut für Theoretische Informatik

Wintersemester 2023/2024

Wo sind wir?

Karlsruher Institut für Technologie

Es war einmal …

Lösen einer Sorte quadratischer Gleichungen

Zum informellen Algorithmusbegriff

Einführung des Hoare-Kalküls

Algorithmus zur Multiplikation nichtnegativer ganzer Zahlen

Eine Zeitreise

- Wie weit in die Vergangenheit kann man reisen und findet noch etwas, was mit Informatik zu tun hat? (jenseits von Zählen und Zahlen)

Eine Zeitreise

- Wie weit in die Vergangenheit kann man reisen und findet noch etwas, was mit Informatik zu tun hat? (jenseits von Zählen und Zahlen)
- Zeit: circa 825–830
- Ort: Bagdad, Haus der Weisheit
- Muhammad ibn Mūsā al-Khwārizmī
  - geboren ca. 780
    in Khiva (heute Usbekistan)
    oder Qutrubbull (heute Iran)
  - gestorben ca. 850

Bildquelle: `http://en.wikipedia.org/wiki/Image:Abu_Abdullah_Muhammad_bin_Musa_al-Khwarizmi.jpg`

Zwei wichtige Schriften von al-Khwārizmī (1)

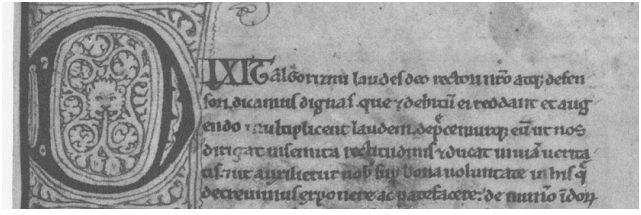
- «Al-Kitāb al-mukhtaṣar fī hīsāb al-ğabr wa'l-muqābala» oder «Al-Kitāb al-mukhtaṣar fī ḥisāb al-jabr wa-l-muqābala»
  - Buch von ca. 830 (?)
  - deutsch: «Das kurzgefasste Buch zum Rechnen durch Ergänzung und Ausgleich»
- aus «al-ğabr» bzw. «al-jabr» wurde später Algebra
- Inhalt des Buches unter anderem:
  Lösen quadratischer Gleichungen mit einer Unbekannten.

Zwei wichtige Schriften von al-Khwārizmī (2)

- Titel vielleicht «Kitāb al-Jamʿ wa-l-tafrīq bi-ḥisāb al-Hind»
  - ca. 825 (??)
  - «Über das Rechnen mit indischen Ziffern»
  - führt die Zahl Null in das arabische Zahlensystem ein …
  - nur noch Übersetzungen, z. B. auf Lateinisch, 12. Jhdt. (?):

Bildquelle:

commons.wikimedia.org/wiki/File:Dixit_algorizmi.png



  - Titel vielleicht etwa «Algorismi de numero Indorum» o. ä.
  - also ein Buch «von Al-gorismi über die indischen Zahlen».
  - Das «i» am Ende von «Algorismi» später fälschlicherweise als Pluralendung des Wortes Algorithmus angesehen.

Wo sind wir?

Es war einmal ...

Lösen einer Sorte quadratischer Gleichungen

Zum informellen Algorithmusbegriff

Einführung des Hoare-Kalküls

Algorithmus zur Multiplikation nichtnegativer ganzer Zahlen

Lösen einer Sorte quadratischer Gleichungen nach al-Khwārizmī (1)

- gegeben: quadratische Gleichung der Form

$$x^2 + bx = c \qquad \text{mit } b > 0 \text{ und } c > 0$$

- al-Khwārizmī: die positive Lösung findet man so:

$$h \leftarrow b/2 \tag{1}$$
$$q \leftarrow h^2 \tag{2}$$
$$s \leftarrow c + q \tag{3}$$
$$w \leftarrow \sqrt{s} \tag{4}$$
$$x \leftarrow w - h \tag{5}$$

- Behauptung: s nie negativ,
  am Ende $x > 0$ und Lösung von $x^2 + bx = c$

Lösen einer Sorte quadratischer Gleichungen nach al-Khwārizmī (1)

- gegeben: quadratische Gleichung der Form

$$x^2 + bx = c \qquad \text{mit } b > 0 \text{ und } c > 0$$

- al-Khwārizmī: die positive Lösung findet man so:

«Algorithmus»

$$\begin{aligned} h &\leftarrow b/2 && (1)\\ q &\leftarrow h^2 && (2)\\ s &\leftarrow c + q && (3)\\ w &\leftarrow \sqrt{s} && (4)\\ x &\leftarrow w - h && (5) \end{aligned}$$

- Behauptung: s nie negativ,
  am Ende $x > 0$ und Lösung von $x^2 + bx = c$

Wo sind wir?

Karlsruher Institut für Technologie

Es war einmal ...

Lösen einer Sorte quadratischer Gleichungen

Zum informellen Algorithmusbegriff

Einführung des Hoare-Kalküls

Algorithmus zur Multiplikation nichtnegativer ganzer Zahlen

Algorithmusbegriff informell

Karlsruher Institut für Technologie

Eigenschaften des eben gezeigten Algorithmus:

- endliche Beschreibung
- Abfolge von einzelnen Schritten
  - elementare Anweisungen: jede *offensichtlich effektiv* in einem Schritt ausführbar
  - endliche viele Schritte: nur endlich oft eine elementare Anweisung ausgeführt
- endliche Eingabe $\rightsquigarrow$ endliche Ausgabe
- beliebig große Eingaben bearbeitbar
- Determinismus: nächste elementare Anweisung immer eindeutig festgelegt
- Nachvollziehbarkeit/Verständlichkeit des Algorithmus:
  Ablauf *klar nachvollziehbar* für Eingeweihte

Algorithmusbegriff informell – Varianten

- auch Verallgemeinerungen sind interessant
  - randomisierte Algorithmen
    - Zufall beeinflusst Auswahl eines Schrittes
  - parallelisierte Algorithmen
    - Schritte werden nicht notwendigerweise in einer Reihe getätigt
  - Online-Algorithmen
    - Eingaben stehen erst nach und nach zur Verfügung
  - nicht terminierende Berechnungen
    - z. B. Ampelsteuerung
  - und noch mehr …

Diskussion des informellen Algorithmusbegriffs

- **Ziel heute:**
  - Eine generelle Methodik, wie man sich überzeugen kann, dass Algorithmen korrekt sind.
  - Die Idee eines Kalkül, um Beweise darüber zu führen

- **obige Forderungen sind plausibel aber informell:**
  - Was heißt «offensichtlich effektiv ausführbar» ?
  - Was heißt «nachvollziehbar» ?

Diskussion des informellen Algorithmusbegriffs

- **Ziel heute:**
  - Eine generelle Methodik, wie man sich überzeugen kann, dass Algorithmen korrekt sind.
  - Die Idee eines Kalkül, um Beweise darüber zu führen

- **obige Forderungen sind plausibel aber informell:**
  - Was heißt «offensichtlich effektiv ausführbar» ?
  - Was heißt «nachvollziehbar» ?

- **Belastbare Aussagen benötigen einen präziseren Algorithmusbegriff**

Wo sind wir?

Karlsruher Institut für Technologie

Es war einmal ...

Lösen einer Sorte quadratischer Gleichungen

Zum informellen Algorithmusbegriff

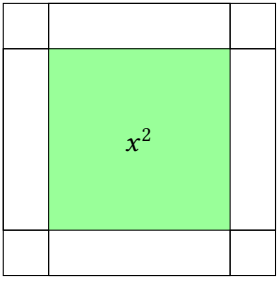
Einführung des Hoare-Kalküls

Algorithmus zur Multiplikation nichtnegativer ganzer Zahlen

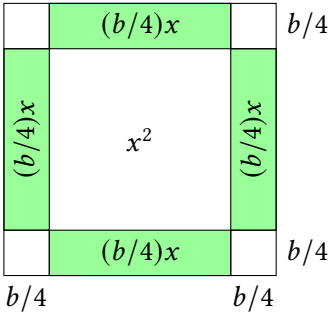
Korrektheit eines Algorithmus — wie beweist man sie?

- Betrachtung des Einzelfalls
  - Beispiel von al-Khwārizmī
- Systematische Betrachtung
  - formales Regelwerk
  - verschiedene Möglichkeiten
  - Beispiel Hoare-Tripel
- Nota bene:
  - Analyse der syntaktischen Beschreibung des Algorithmus
  - Nicht Testen oder Ausprobieren oder Ablaufen oder Beispiele
  - Verständnis durch systematisches Anschauen und Analysieren.

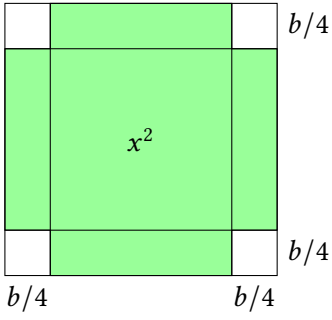
Beweis von al-Khwārizmī



Beweis von al-Khwārizmī

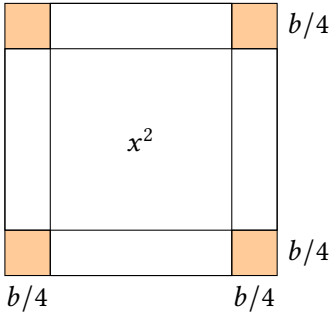


Beweis von al-Khwārizmī



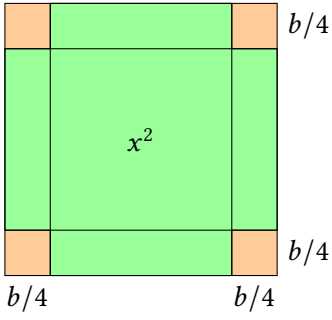
- $x^2 + bx = c$

Beweis von al-Khwārizmī



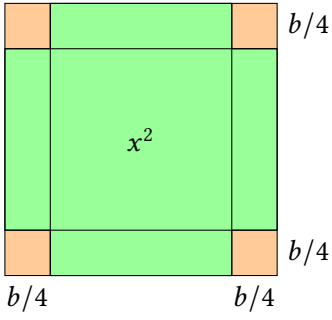
- $x^2 + bx = c$
- $4 \cdot b^2/16 = b^2/4 = q$

Beweis von al-Khwārizmī



- $x^2 + bx = c$
- $4 \cdot b^2/16 = b^2/4 = q$
- $c + q = (b/4 + x + b/4)^2$

Beweis von al-Khwārizmī



- $x^2 + bx = c$
- $4 \cdot b^2/16 = b^2/4 = q$
- $c + q = (b/4 + x + b/4)^2$
- $\sqrt{c+q} - b/2 = x$

Beweis durch Nachrechnen

$\{\ b > 0 \wedge c > 0\ \}$

$h \leftarrow b/2$

$\{\ h = b/2\ \}$

$q \leftarrow h^2$

$\{\ q = b^2/4\ \}$

$s \leftarrow c + q$

$\{\ s = c + b^2/4\ \}$

$w \leftarrow \sqrt{s}$

$\{\ w = \sqrt{c + b^2/4}\ \}$

$x \leftarrow w - h$

$\{\ x = \sqrt{c + b^2/4} - b/2\ \}$

$\{\ x^2 + bx = (\sqrt{c + b^2/4} - b/2)^2 + b(\sqrt{c + b^2/4} - b/2)\ \}$

$\{\ x^2 + bx = c\ \}$

Beweis durch Nachrechnen

$\{\ b > 0 \land c > 0\ \}$

$h \leftarrow b/2$

$\{\ h = b/2\ \}$

$q \leftarrow h^2$

$\{\ q = b^2/4 \land h = b/2\ \}$

$s \leftarrow c + q$

$\{\ s = c + b^2/4 \land h = b/2\ \}$

$w \leftarrow \sqrt{s}$

$\{\ w = \sqrt{c + b^2/4} \land h = b/2\ \}$

$x \leftarrow w - h$

$\{\ x = \sqrt{c + b^2/4} - b/2\ \}$

$\{\ x^2 + bx = (\sqrt{c + b^2/4} - b/2)^2 + b(\sqrt{c + b^2/4} - b/2)\ \}$

$\{\ x^2 + bx = c\ \}$

Eine einfache «Programmiersprache»

- Terminalsymbole
  - Zuweisungssymbol ←
  - Schlüsselwörter **if**, **then**, **else**, **fi**
  - Schlüsselwörter **while**, **do**, **od**, **for**, **to**
  - Symbole für Variablen, Konstanten, Funktionen und Relationen

Eine einfache «Programmiersprache»

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

Beispiel

«Eingaben» sind $a, b \geq 0$

$x \leftarrow a$
$y \leftarrow b$
$z \leftarrow x$
while $y > 0$ **do**
    $z \leftarrow z + 1$
    $y \leftarrow y - 1$
od

Eine einfache «Programmiersprache»

- Produktionen der kontextfreien Grammatik
  - $\langle Prog \rangle \longrightarrow \langle Stmt \rangle \mid \langle Stmt \rangle \ ; \ \langle Prog \rangle$

Beispiel

«Eingaben» sind
$a, b \geq 0$

$x \leftarrow a;$
$y \leftarrow b;$
$z \leftarrow x;$
while $y > 0$ **do**
    $z \leftarrow z + 1;$
    $y \leftarrow y - 1$
od

Eine einfache «Programmiersprache»

- Produktionen der kontextfreien Grammatik
  - $\langle \mathit{Prog} \rangle \to \langle \mathit{Stmt} \rangle \mid \langle \mathit{Stmt} \rangle$ **;** $\langle \mathit{Prog} \rangle$
  - $\langle \mathit{Stmt} \rangle \to \langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$
    $\mid$ **if** $\langle \mathit{Bool} \rangle$ **then** $\langle \mathit{Prog} \rangle$ **else** $\langle \mathit{Prog} \rangle$ **fi**
    $\mid$ **while** $\langle \mathit{Bool} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
    $\mid$ **for** $\langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$ **to** $\langle \mathit{Exp} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**

Beispiel

«Eingaben» sind
$a, b \geq 0$

$x \leftarrow a$;
$y \leftarrow b$;
$z \leftarrow x$;
while $y > 0$ **do**
  $z \leftarrow z + 1$;
  $y \leftarrow y - 1$
od

Eine einfache «Programmiersprache»

- Produktionen der kontextfreien Grammatik
  - $\langle \mathit{Prog} \rangle \to \langle \mathit{Stmt} \rangle \mid \langle \mathit{Stmt} \rangle$; $\langle \mathit{Prog} \rangle$
  - $\langle \mathit{Stmt} \rangle \to \langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$
    $\mid$ **if** $\langle \mathit{Bool} \rangle$ **then** $\langle \mathit{Prog} \rangle$ **else** $\langle \mathit{Prog} \rangle$ **fi**
    $\mid$ **while** $\langle \mathit{Bool} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
    $\mid$ **for** $\langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$ **to** $\langle \mathit{Exp} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
  - $\langle \mathit{Var} \rangle \to$ `x` $\mid$ `y` $\mid$ `z` $\mid$ `a` $\mid \ldots$

Beispiel

«Eingaben» sind
$a, b \geq 0$

$x \leftarrow a$;
$y \leftarrow b$;
$z \leftarrow x$;
while $y > 0$ **do**
  $z \leftarrow z + 1$;
  $y \leftarrow y - 1$
od

Eine einfache «Programmiersprache»

- Produktionen der kontextfreien Grammatik
  - $\langle \mathit{Prog} \rangle \to \langle \mathit{Stmt} \rangle \mid \langle \mathit{Stmt} \rangle$ **;** $\langle \mathit{Prog} \rangle$
  - $\langle \mathit{Stmt} \rangle \to \langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$
    $\mid$ **if** $\langle \mathit{Bool} \rangle$ **then** $\langle \mathit{Prog} \rangle$ **else** $\langle \mathit{Prog} \rangle$ **fi**
    $\mid$ **while** $\langle \mathit{Bool} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
    $\mid$ **for** $\langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$ **to** $\langle \mathit{Exp} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
  - $\langle \mathit{Var} \rangle \to$ `x` $\mid$ `y` $\mid$ `z` $\mid$ `a` $\mid \ldots$
  - $\langle \mathit{Exp} \rangle \to \langle \mathit{Exp} \rangle$`+`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Exp} \rangle$`*`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Literal} \rangle \mid \langle \mathit{Var} \rangle \mid \ldots$

Beispiel

«Eingaben» sind
$a, b \geq 0$

$x \leftarrow a$;
$y \leftarrow b$;
$z \leftarrow x$;
while $y > 0$ **do**
  $z \leftarrow z + 1$;
  $y \leftarrow y - 1$
od

Eine einfache «Programmiersprache»

- Produktionen der kontextfreien Grammatik
  - $\langle \mathit{Prog} \rangle \to \langle \mathit{Stmt} \rangle \mid \langle \mathit{Stmt} \rangle$; $\langle \mathit{Prog} \rangle$
  - $\langle \mathit{Stmt} \rangle \to \langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$
    $\mid$ **if** $\langle \mathit{Bool} \rangle$ **then** $\langle \mathit{Prog} \rangle$ **else** $\langle \mathit{Prog} \rangle$ **fi**
    $\mid$ **while** $\langle \mathit{Bool} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
    $\mid$ **for** $\langle \mathit{Var} \rangle \leftarrow \langle \mathit{Exp} \rangle$ **to** $\langle \mathit{Exp} \rangle$ **do** $\langle \mathit{Prog} \rangle$ **od**
  - $\langle \mathit{Var} \rangle \to$ `x` $\mid$ `y` $\mid$ `z` $\mid$ `a` $\mid \ldots$
  - $\langle \mathit{Exp} \rangle \to \langle \mathit{Exp} \rangle$`+`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Exp} \rangle$`*`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Literal} \rangle \mid \langle \mathit{Var} \rangle \mid \ldots$
  - $\langle \mathit{Bool} \rangle \to \langle \mathit{Exp} \rangle$`=`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Exp} \rangle$`>`$\langle \mathit{Exp} \rangle \mid \langle \mathit{Bool} \rangle \wedge \langle \mathit{Bool} \rangle \mid$
    $\langle \mathit{Bool} \rangle \vee \langle \mathit{Bool} \rangle \mid \neg \langle \mathit{Bool} \rangle \mid \ldots$

Beispiel

«Eingaben» sind
$a, b \geq 0$

$x \leftarrow a$;
$y \leftarrow b$;
$z \leftarrow x$;
while $y > 0$ **do**
  $z \leftarrow z + 1$;
  $y \leftarrow y - 1$
od

Hoare-Tripel —
Programmstück mit Zusicherungen

- benannt nach Sir Charles Antony Richard Hoare

- $\{P\}\ S\ \{Q\}$
  - S Programmstück
  - P Vorbedingung
  - Q Nachbedingung
- P, Q Zusicherungen
  - prädikatenlogische Formeln
  - frei vorkommende Variablen
    - Variablen eines Programms, von dem S ein Teil ist
    - nicht notwendig Variablen, die in S vorkommen

Hoare-Tripel —
Programmstück mit Zusicherungen

- benannt nach Sir Charles Antony Richard Hoare

Beispiel

$\{ x \geq 0 \}$
$x \leftarrow x + 1$
$\{ x \geq 1 \}$

- $\{P\}\ S\ \{Q\}$
  - S Programmstück
  - P Vorbedingung
  - Q Nachbedingung
- P, Q Zusicherungen
  - prädikatenlogische Formeln
  - frei vorkommende Variablen
    - Variablen eines Programms, von dem S ein Teil ist
    - nicht notwendig Variablen, die in S vorkommen

Eine fixierte Interpretation

Im folgenden ist die Interpretation festgelegt:

$$(D, I) = (\mathbb{N}_0, I_{\mathbb{N}})$$

mit der offensichtlichen Interpretation der Funktions- und Relationssymbole:

$$I_{\mathbb{N}}(+)(n, m) = n + m$$

$$I_{\mathbb{N}}(*)(n, m) = n \cdot m$$

$$I_{\mathbb{N}}(>)(n, m) = \mathbf{w} \text{ gdw. } n > m$$

$$\ldots$$

Bedeutung von Programmen

- Programme ändern die Variablenbelegung
- Zuweisung $x \leftarrow E$ Variablensymbol x, Term E
- wenn vorher Variablenbelegung β
  dann hinterher Variablenbelegung

  $\beta' = \beta_x^{\mathrm{val}_{D,I,\beta}(E)}$

  also
  - fast alles unverändert, nur
  - x hat nun den Wert $\mathrm{val}_{D,I,\beta}(E)$
- Bedeutung der *Kontrollstrukturen* **while** und **if**
  - hoffentlich aus Veranstaltungen zum Programmieren bekannt
  - hier keine formale Definition

Gültigkeit von Hoare-Tripeln

- $\{P\}\ S\ \{Q\}$ gültig, wenn
  - für die Interpretation $(D, I) = (\mathbb{N}_0, I_{\mathbb{N}})$ und
  - jede Variablenbelegung β

  gilt:

Wenn P vor S

  - wenn $val_{D,I,\beta}(P) = \mathbf{w}$ und
  - wenn die Ausführung von S für I und β endet und hinterher Variablenbelegung β' vorliegt

dann Q nach S

  - dann $val_{D,I,\beta'}(Q) = \mathbf{w}$

Gültigkeit von Hoare-Tripeln

- $\{P\}\ S\ \{Q\}$ gültig, wenn
  - für die Interpretation $(D, I) = (\mathbb{N}_0, I_{\mathbb{N}})$ und
  - jede Variablenbelegung β

  gilt:

Wenn P vor S
  - wenn $val_{D,I,\beta}(P) = \mathbf{w}$ und
  - wenn die Ausführung von S für I und β endet und hinterher Variablenbelegung β' vorliegt

dann Q nach S
  - dann $val_{D,I,\beta'}(Q) = \mathbf{w}$

Beispiele

- $\{\ x > 0\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 1\ \}$

  ist gültig

- $\{\ x + 1 > 2\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 2\ \}$

  ist gültig

Gültigkeit von Hoare-Tripeln

- $\{P\}\ S\ \{Q\}$ gültig, wenn
  - für die Interpretation $(D, I) = (\mathbb{N}_0, I_{\mathbb{N}})$ und
  - jede Variablenbelegung β

  gilt:

Wenn P vor S
  - wenn $val_{D,I,\beta}(P) = \mathbf{w}$ und
  - wenn die Ausführung von S für I und β endet und hinterher Variablenbelegung β' vorliegt

dann Q nach S
  - dann $val_{D,I,\beta'}(Q) = \mathbf{w}$

Beispiele

- $\{\ x > 0\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 1\ \}$

  ist gültig

- $\{\ x + 1 > 2\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 2\ \}$

  ist gültig

Gültigkeit von Hoare-Tripeln

- $\{P\}\ S\ \{Q\}$ gültig, wenn
  - für die Interpretation $(D, I) = (\mathbb{N}_0, I_{\mathbb{N}})$ und
  - jede Variablenbelegung β

  gilt:

Wenn P vor S
  - wenn $val_{D,I,\beta}(P) = \mathbf{w}$ und
  - wenn die Ausführung von S für I und β endet und hinterher Variablenbelegung β' vorliegt

dann Q nach S
  - dann $val_{D,I,\beta'}(Q) = \mathbf{w}$

Beispiele

- $\{\ x > 0\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 1\ \}$

  ist gültig

- $\{\ x + 1 > 2\ \}$
  $x \leftarrow x + 1$
  $\{\ x > 2\ \}$

  ist gültig

- $\{\ x + 1 > 2\ \}$
  $x \leftarrow x + 1$
  $\{\ x = 0\ \}$

  ist nicht gültig

Hoare-Kalkül — Schlussregeln

$$\frac{}{\{Q[x/E]\}\ x \leftarrow E\ \{Q\}}\ \text{(HT-A)}(*)$$

$$\frac{P' \to P \qquad \{P\}\ S\ \{Q\} \qquad Q \to Q'}{\{P'\}\ S\ \{Q'\}}\ \text{(HT-E)}$$

$$\frac{\{P\}\ S_1\ \{Q\} \qquad \{Q\}\ S_2\ \{R\}}{\{P\}\ S_1 ; S_2\ \{R\}}\ \text{(HT-E)}$$

$$\frac{\{P \wedge B\}\ S_1\ \{Q\} \qquad \{P \wedge \neg B\}\ S_2\ \{Q\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}\ \{Q\}}\ \text{(HT-I)}$$

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

(*) für quantorenfreie Formeln Q

Hoare-Kalkül —
Axiome und Ableitungsregeln für Hoare-Tripel

Ziel

- Axiome und Ableitungsregeln für Hoare-Tripel so, dass
- genau die gültigen Hoare-Tripel ableitbar sind

Axiome für Hoare-Kalkül — für Zuweisungen

- wenn
  - Zuweisung $x \leftarrow E$ von Ausdruck $E \in \mathit{Ter}$
  - Q Nachbedingung zu $x \leftarrow E$
  - Substitution $\sigma_{\{x/E\}}$ sei kollisionsfrei für Q (z. B. Q quantorenfrei)
- dann ist Axiom

  HT-A: $\{\sigma_{\{x/E\}}(Q)\}\ x \leftarrow E\ \{Q\}$
- beachte:

HT steht für Hoare-Tripel

Axiome für Hoare-Kalkül — für Zuweisungen

- wenn
  - Zuweisung $x \leftarrow E$ von Ausdruck $E \in \mathit{Ter}$
  - Q Nachbedingung zu $x \leftarrow E$
  - Substitution $\sigma_{\{x/E\}}$ sei kollisionsfrei für Q (z. B. Q quantorenfrei)
- dann ist Axiom

  HT-A: $\{\sigma_{\{x/E\}}(Q)\}\ x \leftarrow E\ \{Q\}$

HT steht für Hoare-Tripel

- beachte:
  - diese Tripel sind gültig
  - rückwärts: Vorbedingung aus Nachbedingung (vorwärts «geht nicht»)
  - andere Schreibweisen
    - $\{Q[E/x]\}\ x \leftarrow E\ \{Q\}$ oder $\{[E/x]Q\}\ x \leftarrow E\ \{Q\}$
    - $\{Q[x/E]\}\ x \leftarrow E\ \{Q\}$ (das erlauben wir auch)

Ableitungsregeln HT-E und HT-S im Hoare-Kalkül

- Ableitungsregel für Verstärkung der Vorbedingung und Abschwächung der Nachbedingung

  wenn $P' \longrightarrow P$ und $Q \longrightarrow Q'$ gelten, dann ist eine Regel:

  HT-E: $$\frac{\{P\}\ S\ \{Q\}}{\{P'\}\ S\ \{Q'\}}$$

- Ableitungsregel für Hintereinanderausführung

  HT-S: $$\frac{\{P\}\ S_1\ \{Q\} \qquad \{Q\}\ S_2\ \{R\}}{\{P\}\ S_1; S_2\ \{R\}}$$

- beachte: Gültigkeit bleibt erhalten

Beispiel

$\{ x = a \}$ zeige Ableitbarkeit von $\{x = a\}\ y \leftarrow x; z \leftarrow y\ \{z = a\}$

$y \leftarrow x; z \leftarrow y$

$\{ z = a \}$

Beispiel

$\{\ x = a\ \}$

$y \leftarrow x$

$\{\qquad\qquad\}$

$\{\qquad\qquad\}$

$z \leftarrow y$

$\{\ z = a\ \}$

zeige Ableitbarkeit von $\{x = a\}\ y \leftarrow x; z \leftarrow y\ \{z = a\}$

- auseinander ziehen

Beispiel

$\{\ x = a\ \}$

$y \leftarrow x$

$\{\qquad\qquad\}$

$\{\ y = a\ \}$

$z \leftarrow y$

$\{\ z = a\ \}$

zeige Ableitbarkeit von $\{x = a\}\ y \leftarrow x; z \leftarrow y\ \{z = a\}$

- auseinander ziehen
- HT-A: $\{y = a\}\ z \leftarrow y\ \{z = a\}$ ist ableitbar

Beispiel

$\{ x = a \}$

$y \leftarrow x$

$\{ y = a \}$

$\{ y = a \}$

$z \leftarrow y$

$\{ z = a \}$

zeige Ableitbarkeit von $\{x = a\}\ y \leftarrow x; z \leftarrow y\ \{z = a\}$

- auseinander ziehen
- HT-A: $\{y = a\}\ z \leftarrow y\ \{z = a\}$ ist ableitbar
- HT-A: $\{x = a\}\ y \leftarrow x\ \{y = a\}$ ist ableitbar

Beispiel

$\{ x = a \}$

$y \leftarrow x$

$\{ y = a \}$

$\{ y = a \}$

$z \leftarrow y$

$\{ z = a \}$

zeige Ableitbarkeit von $\{x = a\}\ y \leftarrow x; z \leftarrow y\ \{z = a\}$

- auseinander ziehen
- HT-A: $\{y = a\}\ z \leftarrow y\ \{z = a\}$ ist ableitbar
- HT-A: $\{x = a\}\ y \leftarrow x\ \{y = a\}$ ist ableitbar
- HT-S: fertig

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{ \ b > 0 \land c > 0 \ \}$

$\{ \quad \}$

$\{ \quad \}$

$h \leftarrow b/2$

$\{ \quad \}$

$q \leftarrow h^2$

$\{ \quad \}$

$s \leftarrow c + q$

$\{ \quad \}$

$w \leftarrow \sqrt{s}$

$\{ \quad \}$

$x \leftarrow w - h$

1 $\{ \ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \ \}$

$\{ \quad \}$

$\{ \ x^2 + bx = c \land x > 0 \ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{ \; b > 0 \wedge c > 0 \; \}$

$\{ \quad \}$

$\{ \quad \}$

$h \leftarrow b/2$

$\{ \quad \}$

$q \leftarrow h^2$

$\{ \quad \}$

$s \leftarrow c + q$

$\{ \quad \}$

$w \leftarrow \sqrt{s}$

2 $\{ \; w - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A} \; \}$

$x \leftarrow w - h$

1 $\{ \; x = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A} \; \}$

$\{ \quad \}$

$\{ \; x^2 + bx = c \wedge x > 0 \; \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{ \; b > 0 \land c > 0 \; \}$

$\{ \quad \}$

$\{ \quad \}$

$h \leftarrow b/2$

$\{ \quad \}$

$q \leftarrow h^2$

$\{ \quad \}$

$s \leftarrow c + q$

3 $\{ \; \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \; \}$

$w \leftarrow \sqrt{s}$

2 $\{ \; w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \; \}$

$x \leftarrow w - h$

1 $\{ \; x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \; \}$

$\{ \quad \}$

$\{ \; x^2 + bx = c \land x > 0 \; \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{\ b > 0 \land c > 0\ \}$

$\{\quad\}$

$\{\qquad\qquad\qquad\}$

$h \leftarrow b/2$

$\{\qquad\qquad\qquad\}$

$q \leftarrow h^2$

4 $\{\ \sqrt{c+q} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$s \leftarrow c + q$

3 $\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$w \leftarrow \sqrt{s}$

2 $\{\ w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$x \leftarrow w - h$

1 $\{\ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$\{\qquad\qquad\qquad\}$

$\{\ x^2 + bx = c \land x > 0\ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{\ b > 0 \land c > 0\ \}$

$\{\quad\}$

$\{\qquad\qquad\qquad\qquad\}$

$h \leftarrow b/2$

5 $\{\ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$q \leftarrow h^2$

4 $\{\ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$s \leftarrow c + q$

3 $\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$w \leftarrow \sqrt{s}$

2 $\{\ w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$x \leftarrow w - h$

1 $\{\ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$\{\qquad\qquad\qquad\qquad\}$

$\{\ x^2 + bx = c \land x > 0\ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{ b > 0 \land c > 0 \}$

$\{ \quad \}$

6 $\{ \sqrt{c + (b/2)^2} - b/2 = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$h \leftarrow b/2$

5 $\{ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$q \leftarrow h^2$

4 $\{ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$s \leftarrow c + q$

3 $\{ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$w \leftarrow \sqrt{s}$

2 $\{ w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$x \leftarrow w - h$

1 $\{ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A} \}$

$\{ \quad \}$

$\{ x^2 + bx = c \land x > 0 \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

$\{\ b > 0 \land c > 0\ \}$

7 $\{\ \mathcal{A} : \sqrt{c + (b/2)^2} \text{ definiert und } \sqrt{c + (b/2)^2} - b/2 > 0\ \}$

6 $\{\ \sqrt{c + (b/2)^2} - b/2 = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$h \leftarrow b/2$

5 $\{\ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$q \leftarrow h^2$

4 $\{\ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$s \leftarrow c + q$

3 $\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$w \leftarrow \sqrt{s}$

2 $\{\ w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$x \leftarrow w - h$

1 $\{\ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$\{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\}$

$\{\ x^2 + bx = c \land x > 0\ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

8 $\{\ b > 0 \land c > 0\ \}$

7 $\{\ \mathcal{A} : \sqrt{c + (b/2)^2}$ definiert und $\sqrt{c + (b/2)^2} - b/2 > 0\ \}$

6 $\{\ \sqrt{c + (b/2)^2} - b/2 = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$h \leftarrow b/2$

5 $\{\ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$q \leftarrow h^2$

4 $\{\ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$s \leftarrow c + q$

3 $\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$w \leftarrow \sqrt{s}$

2 $\{\ w - h = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$x \leftarrow w - h$

1 $\{\ x = \sqrt{c + b^2/4} - b/2 \land \mathcal{A}\ \}$

$\{\qquad\qquad\qquad\qquad\}$

$\{\ x^2 + bx = c \land x > 0\ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

8	$\{\ b > 0 \wedge c > 0\ \}$
7	$\{\ \mathcal{A} : \sqrt{c + (b/2)^2}$ definiert und $\sqrt{c + (b/2)^2} - b/2 > 0\ \}$
6	$\{\ \sqrt{c + (b/2)^2} - b/2 = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$h \leftarrow b/2$
5	$\{\ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$q \leftarrow h^2$
4	$\{\ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$s \leftarrow c + q$
3	$\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$w \leftarrow \sqrt{s}$
2	$\{\ w - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$x \leftarrow w - h$
1	$\{\ x = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
9	$\{\ x^2 + bx = (\sqrt{c + b^2/4} - b/2)^2 + b(\sqrt{c + b^2/4} - b/2) \wedge x > 0\ \}$
	$\{\ x^2 + bx = c \wedge x > 0\ \}$

Beispiel (2) — Algorithmus von al-Khwārizmī

8	$\{\ b > 0 \wedge c > 0\ \}$
7	$\{\ \mathcal{A} : \sqrt{c + (b/2)^2}$ definiert und $\sqrt{c + (b/2)^2} - b/2 > 0\ \}$
6	$\{\ \sqrt{c + (b/2)^2} - b/2 = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$h \leftarrow b/2$
5	$\{\ \sqrt{c + h^2} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$q \leftarrow h^2$
4	$\{\ \sqrt{c + q} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$s \leftarrow c + q$
3	$\{\ \sqrt{s} - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$w \leftarrow \sqrt{s}$
2	$\{\ w - h = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
	$x \leftarrow w - h$
1	$\{\ x = \sqrt{c + b^2/4} - b/2 \wedge \mathcal{A}\ \}$
9	$\{\ x^2 + bx = (\sqrt{c + b^2/4} - b/2)^2 + b(\sqrt{c + b^2/4} - b/2) \wedge x > 0\ \}$
10	$\{\ x^2 + bx = c \wedge x > 0\ \}$

Regel HT-I — für bedingte Anweisungen

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

- HT-I:

$$\frac{\{P \wedge B\}\ S_1\ \{Q\} \qquad \{P \wedge \neg B\}\ S_2\ \{Q\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}\ \{Q\}}$$

- beachte: Gültigkeit «bleibt erhalten»

Beispiel für HT-I

$\{\ true\ \}$
if $x < 0$
then
$\quad x \leftarrow -x$
else
$\quad x \leftarrow x$
fi
$\{\ x \geq 0\ \}$

- Domäne sei für diese Folie $D = \mathbb{Z}$

Beispiel für HT-I

$\{\ true\ \}$
if $x < 0$
then
$\{\quad\}$
$\{\quad\}$
$x \leftarrow -x$
$\{\quad\}$
else
$\{\quad\}$
$\{\quad\}$
$x \leftarrow x$
$\{\quad\}$
fi
$\{\ x \geq 0\ \}$

- Domäne sei für diese Folie $D = \mathbb{Z}$
- **if** B **then** S_1 **else** S_2 **fi**
  - wenn $\{P \wedge B\}\ S_1\ \{Q\}$ ableitbar und
  - wenn $\{P \wedge \neg B\}\ S_2\ \{Q\}$ ableitbar
  - dann $\{P\}$ **if** B **then** S_1 **else** S_2 **fi** $\{Q\}$ ableitbar

Beispiel für HT-I

$\{\ true\ \}$
if $x < 0$
then
$\{\ true \wedge x < 0\ \}$
$\{\quad\quad\}$
$x \leftarrow -x$
$\{\ x \geq 0\ \}$
else
$\{\quad\quad\quad\quad\quad\quad\}$
$\{\quad\quad\}$
$x \leftarrow x$
$\{\quad\quad\}$
fi
$\{\ x \geq 0\ \}$

- Domäne sei für diese Folie $D = \mathbb{Z}$
- **if** B **then** S_1 **else** S_2 **fi**
  - wenn $\{P \wedge B\}\ S_1\ \{Q\}$ ableitbar und
  - wenn $\{P \wedge \neg B\}\ S_2\ \{Q\}$ ableitbar
  - dann $\{P\}$ **if** B **then** S_1 **else** S_2 **fi** $\{Q\}$ ableitbar

Beispiel für HT-I

$\{\ true\ \}$
if $x < 0$
then
$\{\ true \wedge x < 0\ \}$
$\{\qquad\qquad\}$
$x \leftarrow -x$
$\{\ x \geq 0\ \}$
else
$\{\ true \wedge \neg(x < 0)\ \}$
$\{\qquad\qquad\}$
$x \leftarrow x$
$\{\ x \geq 0\ \}$
fi
$\{\ x \geq 0\ \}$

- Domäne sei für diese Folie $D = \mathbb{Z}$
- **if** B **then** S_1 **else** S_2 **fi**
  - wenn $\{P \wedge B\}\ S_1\ \{Q\}$ ableitbar und
  - wenn $\{P \wedge \neg B\}\ S_2\ \{Q\}$ ableitbar
  - dann $\{P\}$ **if** B **then** S_1 **else** S_2 **fi** $\{Q\}$ ableitbar

Beispiel für HT-I

$\{\ true\ \}$
if $x < 0$
then
$\{\ true \wedge x < 0\ \}$
$\{\ -x \geq 0\ \}$
$x \leftarrow -x$
$\{\ x \geq 0\ \}$
else
$\{\ true \wedge \neg(x < 0)\ \}$
$\{\ x \geq 0\ \}$
$x \leftarrow x$
$\{\ x \geq 0\ \}$
fi
$\{\ x \geq 0\ \}$

- Domäne sei für diese Folie $D = \mathbb{Z}$
- **if** B **then** S_1 **else** S_2 **fi**
  - wenn $\{P \wedge B\}\ S_1\ \{Q\}$ ableitbar und
  - wenn $\{P \wedge \neg B\}\ S_2\ \{Q\}$ ableitbar
  - dann $\{P\}$ **if** B **then** S_1 **else** S_2 **fi** $\{Q\}$ ableitbar

Regel HT-W — für **while**-Schleifen

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

- HT-W: $$\frac{\{I \wedge B\}\ S\ \{I\}}{\{I\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{od}\ \{I \wedge \neg B\}}$$
- Zusicherung I heißt Schleifeninvariante
- beachte: Gültigkeit «bleibt erhalten»

Wo sind wir?

Karlsruher Institut für Technologie

Es war einmal ...

Lösen einer Sorte quadratischer Gleichungen

Zum informellen Algorithmusbegriff

Einführung des Hoare-Kalküls

Algorithmus zur Multiplikation nichtnegativer ganzer Zahlen

Erinnerung: **div** und **mod**

- x **div** y
  - Ergebnis der ganzzahligen Division von x durch y
- x **mod** y
  - Rest der ganzzahligen Division von x durch y
  - $0 \leq x \bmod y < y$
- für alle $x, y \in \mathbb{N}_0$ gilt

$$x = y \cdot (x \operatorname{div} y) + (x \bmod y)$$

z. B. $\quad x = 2 \cdot (x \operatorname{div} 2) + (x \bmod 2)$

- Beispiele
  - $6 \operatorname{div} 2 = 3 \quad$ und $\quad 6 \bmod 2 = 0$
  - $7 \operatorname{div} 2 = 3 \quad$ und $\quad 7 \bmod 2 = 1$
  - $8 \operatorname{div} 2 = 4 \quad$ und $\quad 8 \bmod 2 = 0$

Algorithmus für die Multiplikation

$i \leftarrow 0$
$X \leftarrow a$
$Y \leftarrow b$
$P \leftarrow 0$
while $X > 0$ **do**
  $i \leftarrow i + 1$
  $P \leftarrow P + (X \textbf{ mod } 2) \cdot Y$
  $X \leftarrow X \textbf{ div } 2$
  $Y \leftarrow 2 \cdot Y$
od

- Grundbereich $\mathbb{N}_0$, also $I(a), I(b) \in \mathbb{N}_0$
- Angenommen: Multiplikation beliebiger Zahlen nicht verfügbar
- Ziel: Baue Multiplikation mit Addition nach.
- $n \cdot m = \underbrace{n + n + \ldots + n}_{m \text{ mal}}$ ... aufwändig
- Hier: Verfahren mit weniger Additionen ($\log(m)$)

Beispielrechnung

$i \leftarrow 0$
$X \leftarrow a$
$Y \leftarrow b$
$P \leftarrow 0$
while $X > 0$ **do**
  $i \leftarrow i + 1$
  $P \leftarrow P + (X \bmod 2) \cdot Y$
  $X \leftarrow X \operatorname{div} 2$
  $Y \leftarrow 2 \cdot Y$
od

- Es sei $a = 6$ und $b = 9$
- schreibe v_i für Wert von v «nach i Schleifendurchläufen»

	P_i	X_i	Y_i
$i = 0$	0	6	9
$i = 1$	0	3	18
$i = 2$	18	1	36
$i = 3$	54	0	72

Beispielrechnung

$i \leftarrow 0$
$X \leftarrow a$
$Y \leftarrow b$
$P \leftarrow 0$
while $X > 0$ **do**
  $i \leftarrow i + 1$
  $P \leftarrow P + (X \bmod 2) \cdot Y$
  $X \leftarrow X \operatorname{div} 2$
  $Y \leftarrow 2 \cdot Y$
od

- Es sei $a = 6$ und $b = 9$
- schreibe v_i für Wert von v «nach i Schleifendurchläufen»

	P_i	X_i	Y_i
$i = 0$	0	6	9
$i = 1$	0	3	18
$i = 2$	18	1	36
$i = 3$	54	0	72

- am Ende: $P = 54 = a \cdot b$
- wollen beweisen: Das klappt immer!

Algorithmus für die Multiplikation

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ \mathit{true}\ \}$
$i \leftarrow 0$
$X \leftarrow a$
$Y \leftarrow b$
$P \leftarrow 0$

while $X > 0$ **do**
    $i \leftarrow i + 1$
    $P \leftarrow P + (X \bmod 2) \cdot Y$
    $X \leftarrow X \operatorname{div} 2$
    $Y \leftarrow 2 \cdot Y$
od

$\{\ P = a \cdot b\ \}$

Algorithmus für die Multiplikation

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

Schleifeninvariante

$\{\ true\ \}$
$i \leftarrow 0$
$X \leftarrow a$
$Y \leftarrow b$
$P \leftarrow 0$
$\{\ X \cdot Y + P = a \cdot b\ \}$
while $X > 0$ **do**
    $i \leftarrow i + 1$
    $P \leftarrow P + (X \bmod 2) \cdot Y$
    $X \leftarrow X \operatorname{div} 2$
    $Y \leftarrow 2 \cdot Y$
od
$\{\ X \cdot Y + P = a \cdot b \wedge \neg (X > 0)\ \}$
$\{\ P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (1)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b\ \}$
while $X > 0$ **do**
    $\{\ \}$
    $i \leftarrow i + 1$
    $P \leftarrow P + (X \bmod 2) \cdot Y$
    $X \leftarrow X \operatorname{div} 2$
    $Y \leftarrow 2 \cdot Y$
    $\{\ \}$
od
$\{\ \}$
$\{\ P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (1)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b\ \}$
while $X > 0$ **do**
  $\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
  $i \leftarrow i + 1$
  $P \leftarrow P + (X \bmod 2) \cdot Y$
  $X \leftarrow X \operatorname{div} 2$
  $Y \leftarrow 2 \cdot Y$
  $\{\ X \cdot Y + P = a \cdot b\ \}$
od
$\{\ \}$
$\{\ P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (1)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b\ \}$
while $X > 0$ **do**
  $\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
  $i \leftarrow i + 1$
  $P \leftarrow P + (X \bmod 2) \cdot Y$
  $X \leftarrow X \operatorname{div} 2$
  $Y \leftarrow 2 \cdot Y$
  $\{\ X \cdot Y + P = a \cdot b\ \}$
od
$\{\ X \cdot Y + P = a \cdot b \wedge \neg(X > 0)\ \}$
$\{\ P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ \}$
$\{\ \}$
$i \leftarrow i + 1$
$\{\ \}$
$P \leftarrow P + (X \ \mathbf{mod}\ 2) \cdot Y$
$\{\ \}$
$X \leftarrow X \ \mathbf{div}\ 2$
$\{\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ \}$
$\{\ \}$
$i \leftarrow i + 1$
$\{\ \}$
$P \leftarrow P + (X \bmod 2) \cdot Y$
$\{\ \}$
$X \leftarrow X \operatorname{div} 2$
$\{\ X \cdot (2Y) + P = a \cdot b\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ \}$
$\{\ \}$
$i \leftarrow i + 1$
$\{\ \}$
$P \leftarrow P + (X \bmod 2) \cdot Y$
$\{\ (X \operatorname{div} 2) \cdot (2Y) + P = a \cdot b\ \}$
$X \leftarrow X \operatorname{div} 2$
$\{\ X \cdot (2Y) + P = a \cdot b\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ \}$
$\{\ \}$
$i \leftarrow i + 1$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P + (X \ \mathbf{mod}\ 2) \cdot Y = a \cdot b\ \}$
$P \leftarrow P + (X \ \mathbf{mod}\ 2) \cdot Y$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P = a \cdot b\ \}$
$X \leftarrow X \ \mathbf{div}\ 2$
$\{\ X \cdot (2Y) + P = a \cdot b\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ \}$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P + (X \ \mathbf{mod}\ 2) \cdot Y = a \cdot b\ \}$
$i \leftarrow i + 1$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P + (X \ \mathbf{mod}\ 2) \cdot Y = a \cdot b\ \}$
$P \leftarrow P + (X \ \mathbf{mod}\ 2) \cdot Y$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P = a \cdot b\ \}$
$X \leftarrow X \ \mathbf{div}\ 2$
$\{\ X \cdot (2Y) + P = a \cdot b\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Schleifeninvariante für Multiplikationsalgorithmus (2)

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ X \cdot Y + P = a \cdot b \wedge X > 0\ \}$
$\{\ ((X \ \mathbf{div}\ 2) \cdot 2 + (X \ \mathbf{mod}\ 2)) \cdot Y + P = a \cdot b\ \}$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P + (X \ \mathbf{mod}\ 2) \cdot Y = a \cdot b\ \}$
$i \leftarrow i + 1$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P + (X \ \mathbf{mod}\ 2) \cdot Y = a \cdot b\ \}$
$P \leftarrow P + (X \ \mathbf{mod}\ 2) \cdot Y$
$\{\ (X \ \mathbf{div}\ 2) \cdot (2Y) + P = a \cdot b\ \}$
$X \leftarrow X \ \mathbf{div}\ 2$
$\{\ X \cdot (2Y) + P = a \cdot b\ \}$
$Y \leftarrow 2 \cdot Y$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Algorithmus für die Multiplikation: Initialisierungen

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ true\ \}$
$\{\ \}$
$i \leftarrow 0$
$\{\ \}$
$X \leftarrow a$
$\{\ \}$
$Y \leftarrow b$
$\{\ \}$
$P \leftarrow 0$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Algorithmus für die Multiplikation: Initialisierungen

Grundbereich $\mathbb{N}_0$
also $I(a), I(b) \in \mathbb{N}_0$

$\{\ true\ \}$
$\{\ a \cdot b + 0 = a \cdot b\ \}$
$i \leftarrow 0$
$\{\ a \cdot b + 0 = a \cdot b\ \}$
$X \leftarrow a$
$\{\ X \cdot b + 0 = a \cdot b\ \}$
$Y \leftarrow b$
$\{\ X \cdot Y + 0 = a \cdot b\ \}$
$P \leftarrow 0$
$\{\ X \cdot Y + P = a \cdot b\ \}$

Was ist wichtig

- Das sollten Sie mitnehmen:
  - informeller *Algorithmusbegriff*
  - *Schleifeninvarianten*
- Das sollten Sie üben:
  - Schleifeninvarianten finden
    - Wertetabellen können helfen
  - Korrektheitsbeweise finden mit Hoare-Kalkül