Grundbegriffe der Informatik WS 2011/12 Tutorium in der Woche 4

Gehalten in den Tutorien Nr. 10, Nr. 14

Philipp Basler (philippbasler@googlemail.com)
Nils Braun (area51.nils@googlemail.com)

KIT - Karlsruher Institut für Technologie

14.11.2011 & 15.11.2011

Inhaltsverzeichnis

1 Übungsblätter

2 div und mod Rechnung

3 Algorithmen

4 Schleifeninvarianzen

5 Schluss

1 **Übungsblätter**

2 div und mod Rechnung

3 Algorithmen

4 Schleifeninvarianzen

5 Schluss

Informationen zum nächsten Blatt

Blatt Nr. 4

Abgabetermin	18.1.1111 um 12:30 Uhr
Abgabeort	Briefkasten im UG
Themen	Schleifeninvarianten
Maximale Punkte	19

Statistik

- 21 von 26 Abgaben
- Durchschnittlich 13.8 von 21 Punkten

Häufige Fehler auf dem letzten Übungsblatt

Blatt Nr. 3

- 1. Aufgabe Induktionsanfang für alle 3 Fälle zeigen
- 2. Aufgabe Nicht den Rekursionsanker vergessen
- 3. Aufgabe keine
- 4. Aufgabe möglichst einfach
- 5. Aufgabe Mengenbeweis statt Definition, Distributivgesetz für Mengen hattet ihr noch nicht in der Vorlesung.

- Jede Sprache enthält Wörter

- Jede Sprache enthält Wörter

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift
- Das leere Wort liegt niemals in L^+

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift
- Das leere Wort liegt niemals in L^+

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift
- Das leere Wort liegt niemals in L^+
- Zwei Sprachen mit unterschiedlicher Definition sind immer verschieden

- Jede Sprache enthält Wörter
- Eine Sprache hat eine Abbildungsvorschrift
- Das leere Wort liegt niemals in L^+
- Zwei Sprachen mit unterschiedlicher Definition sind immer verschieden

1 Übungsblätter

2 div und mod Rechnung

3 Algorithmen

4 Schleifeninvarianzen

5 Schluss

Definition

$$\forall\ x \in \mathbb{N}_0 \forall\ y \in \mathbb{N}_+ : x = y \cdot (x \textbf{ div } y) + (x \textbf{ mod } y)$$

Definition

$$\forall\ x \in \mathbb{N}_0 \forall\ y \in \mathbb{N}_+ : x = y \cdot (x \ \mathbf{div}\ y) + (x \ \mathbf{mod}\ y)$$

$$a = x \ \mathbf{mod}\ y \iff \text{Teilung von } x \text{ durch } y \text{ gibt Rest } a$$

$$x \ \mathbf{div}\ y \iff \text{Ganzzahlige Teilung von } x \text{ durch } y$$

Definition

$$\forall\ x \in \mathbb{N}_0 \forall\ y \in \mathbb{N}_+ : x = y \cdot (x \ \mathbf{div} \ y) + (x \ \mathbf{mod} \ y)$$

$$a = x \ \mathbf{mod} \ y \iff \text{Teilung von } x \text{ durch } y \text{ gibt Rest } a$$

$$x \ \mathbf{div} \ y \iff \text{Ganzzahlige Teilung von } x \text{ durch } y$$

Folgerung

$$x \ \mathbf{div} \ y \in \mathbb{N}_0$$

$$x \ \mathbf{mod} \ y \in \{0, \ldots, y-1\}$$

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$		

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$	0	2

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$	0	2
$x = 5, y = 2$		

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$	0	2
$x = 5, y = 2$	2	1

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$	0	2
$x = 5, y = 2$	2	1
$x = 8, y = 2$		

Beispiel

	x **div** y	x **mod** y
$x = 2, y = 3$	0	2
$x = 5, y = 2$	2	1
$x = 8, y = 2$	4	2

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$		
$x = 2, y = 1$		
$x = 10, y = 3$		
$x = 8, y = 3$		
$x = 9, y = 2$		
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$		
$x = 10, y = 3$		
$x = 8, y = 3$		
$x = 9, y = 2$		
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$	2	0
$x = 10, y = 3$		
$x = 8, y = 3$		
$x = 9, y = 2$		
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$	2	0
$x = 10, y = 3$	3	1
$x = 8, y = 3$		
$x = 9, y = 2$		
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$	2	0
$x = 10, y = 3$	3	1
$x = 8, y = 3$	2	2
$x = 9, y = 2$		
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$	2	0
$x = 10, y = 3$	3	1
$x = 8, y = 3$	2	2
$x = 9, y = 2$	4	1
$x = 4, y = 3$		

Jetz ihr

	x **div** y	x **mod** y
$x = 3, y = 4$	0	3
$x = 2, y = 1$	2	0
$x = 10, y = 3$	3	1
$x = 8, y = 3$	2	2
$x = 9, y = 2$	4	1
$x = 4, y = 3$	1	1

größter gemeinsamer Teiler

Der größte gemeinsame Teiler zweier Zahlen a, b ist die größtmögliche Zahl $m \in \mathbb{N}_0$, für die gilt

$$a \textbf{ div } m = 0 \qquad b \textbf{ div } m = 0$$

größter gemeinsamer Teiler

Der größte gemeinsame Teiler zweier Zahlen a, b ist die größtmögliche Zahl $m \in \mathbb{N}_0$, für die gilt

$$a \textbf{ div } m = 0 \qquad b \textbf{ div } m = 0$$

Wie bestimmen wir den ggT?

größter gemeinsamer Teiler

Der größte gemeinsame Teiler zweier Zahlen a, b ist die größtmögliche Zahl $m \in \mathbb{N}_0$, für die gilt

$$a \textbf{ div } m = 0 \qquad b \textbf{ div } m = 0$$

Wie bestimmen wir den ggT?

Primzahlzerlegung

z.B. : $a = 3528, b = 3780$. Dann ergibt sich

$$a = 3528 = 2^3 \cdot 3^2 \cdot 7^2 \qquad b = 3780 = 2^2 \cdot 3^3 \cdot 5^1 \cdot 7^1$$

Es folgt

$$ggT(3528, 3780) = 2^2 \cdot 3^2 \cdot 7^1$$

Wie kann ich den ggT programmieren?

ggT rekursiv

$$ggT(a, b) = \begin{cases} a & \text{falls } b = 0 \\ ggT(b, a \textbf{ mod } b) \text{sonst} \end{cases}$$

1 Übungsblätter

2 div und mod Rechnung

3 Algorithmen

4 Schleifeninvarianzen

5 Schluss

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen
- Determinismus

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen
- Determinismus
- endliche Eingabe gibt endliche Ausgabe

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen
- Determinismus
- endliche Eingabe gibt endliche Ausgabe
- endlich viele Schritte

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen
- Determinismus
- endliche Eingabe gibt endliche Ausgabe
- endlich viele Schritte
- funktioniert für beliebig große Eingaben

Aus der Vorlesung

Algorithmen haben folgende Eigenschaften

- endliche Beschreibung
- elemantere Aussagen
- Determinismus
- endliche Eingabe gibt endliche Ausgabe
- endlich viele Schritte
- funktioniert für beliebig große Eingaben
- nachvollziehbar, verständlich

Warum Schleifen?

Wofür brauchen wir Schleifen?

Warum Schleifen?

Wofür brauchen wir Schleifen?
Endliche, immer gleich bleibende Vorgänge bekannter oder unbekannter Länge

Welche Schleifen gibt es?

Welche Schleifen gibt es?

while Tue solange bis Bedingung nichtmehr gilt

Welche Schleifen gibt es?

while Tue solange bis Bedingung nichtmehr gilt

Welche Schleifen gibt es?

while Tue solange bis Bedingung nichtmehr gilt

for Tue etwas n−mal

Welche Schleifen gibt es?

while Tue solange bis Bedingung nichtmehr gilt

for Tue etwas $n-$mal

Welche Schleifen gibt es?

while Tue solange bis Bedingung nichtmehr gilt

for Tue etwas n−mal

do while Tue etwas und dann überprüfe die Bedingung. Wenn die Bedingung erfüllt ist, tue es solange bis die Bedingung nichtmehr erfüllt ist.

Was tut das?

Input $x \in \mathbb{N}_+$

$i \leftarrow 0$

$r \leftarrow 0$

while $x > 1$ **do**

  $r \leftarrow x$ **mod** 2

  $x \leftarrow x$ **div** 2

  $i \leftarrow i + 1$

od

Output i

Und das?

$k \leftarrow 0$

for $i \leftarrow 0$ **to** 20 **do**

  $k \leftarrow i$

od

Output k

Weils so schön war

Sei w ein Wort der Länge n und der Array W hat am $i-$ten Eintrag den $i-$ten Buchstab von w, weiterhin kommt in w kein ε vor.

$$c \leftarrow 0$$

for $i = 0$ **to** $n - 1$ **do**

$$c \leftarrow \begin{cases} c+1 & \text{falls } W[i] = x \\ c & \text{sonst} \end{cases}$$

od

Aufgabe (WS 2008)

Es sei A ein Alphabet.
Schreiben Sie einen Algorithmus auf, der folgendes leistet: Als Eingaben erhält er ein Wort w über A und zwei Symbole $x \in A$ und $y \in A$. Am Ende soll eine Variable r den Wert 0 oder 1 haben, und zwar soll gelten:

$$r = \begin{cases} 1 & \text{falls irgendwo in w direkt hintereinander} \\ & \text{erst x und dann y vorkommen} \\ 0 & \text{sonst} \end{cases}$$

Benutzen Sie zum Zugriff auf das i-te Symbol von w die Schreibweise $w(i)$. Formulieren Sie den Algorithmus mit Hilfe einer for-Schleife.

Lösung

$$r = \begin{cases} 1 & \text{falls irgendwo in w direkt hintereinander} \\ & \text{erst x und dann y vorkommen} \\ 0 & \text{sonst} \end{cases}$$

Lösung

$$r = \begin{cases} 1 & \text{falls irgendwo in w direkt hintereinander} \\ & \text{erst x und dann y vorkommen} \\ 0 & \text{sonst} \end{cases}$$

Der Algorithmus:

$r \leftarrow 0$

for $i \leftarrow 0$ **to** $n-2$ **do**

$$r \leftarrow \begin{cases} 1 & \text{falls } w(i) = x \text{ und } w(i+1) = y \\ r & \text{sonst} \end{cases}$$

od

löst das Problem.

1 Übungsblätter

2 div und mod Rechnung

3 Algorithmen

4 **Schleifeninvarianzen**

5 Schluss

Sinn und Zweck

Schleifeninvarianten ...

- sind Aussagen, die bei jedem Schleifendurchgang gleich sind
- helfen, die Korrektheit eines Programmes zu beweisen
- beweist man durch vollständige Induktion

Beispiel

// Eingaben $a, b \in \mathbb{N}_0$

$$S \leftarrow a$$
$$Y \leftarrow b$$
for $i \leftarrow 0$ **to** $b - 1$ **do**
$$S \leftarrow S + 1$$
$$Y \leftarrow Y - 1$$
od
$$Output\ S$$

Beispiel

Wertetabelle für a=3 und b=4

i	S	Y
	3	4
0	4	3
1	5	2
2	6	1
3	7	0

Beweis durch Vollständige Induktion über i

Behauptung:

$$\forall\ i \in \{0, b-1\} : S + Y = a + b$$

Beweis durch Vollständige Induktion über i

Behauptung:

$$\forall\ i \in \{0, b-1\} : S + Y = a + b$$

Induktionsanfang

Für $i = 0$ gilt :

$$S_0 + Y_0 = a + 1 + b - 1 = a + b$$

Beweis durch Vollständige Induktion über i

Behauptung:

$$\forall\ i \in \{0, b-1\} : S + Y = a + b$$

Induktionsanfang

Für $i = 0$ gilt :

$$S_0 + Y_0 = a + 1 + b - 1 = a + b$$

Induktionsvorrausetzung

Für ein beliebig aber festes $i \in \{0, b-1\}$ gelte die Behauptung

Induktionsschluß

Zu Zeigen

$$S_{i+1} + Y_{i+1} = a + b$$

Induktionsschluß

Zu Zeigen

$$S_{i+1} + Y_{i+1} = a + b$$

$$\begin{aligned} S_{i+1} + Y_{i+1} &= S_i + 1 + Y_i - 1 \\ &= S_i + Y_i \\ &\overset{IV}{=} a + b \end{aligned}$$

Aufgabe (WS 2008)

Gegeben sei für zwei Eingaben $a, b \in \mathbb{N}_+$ folgender Algorithmus.

$X_0 \leftarrow a$

$Y_0 \leftarrow b$

$P_0 \leftarrow 1$

$x_0 \leftarrow X_0$ **mod** 2

$n \leftarrow 1 + \lceil \log_2 a \rceil$

for $i \leftarrow 0$ **to** $n - 1$ **do**

  $P_{i+1} \leftarrow P_i \cdot Y_i^{x_i}$

  $X_{i+1} \leftarrow X_i$ **div** 2

  $Y_{i+1} \leftarrow Y_i^2$

  $x_{i+1} \leftarrow X_{i+1}$ **mod** 2

od

Aufgabe (WS 2008)

Gegeben sei für zwei Eingaben $a, b \in \mathbb{N}_+$ folgender Algorithmus.

$X_0 \leftarrow a$
$Y_0 \leftarrow b$
$P_0 \leftarrow 1$
$x_0 \leftarrow X_0 \textbf{ mod } 2$
$n \leftarrow 1 + \lceil \log_2 a \rceil$
for $i \leftarrow 0$ **to** $n - 1$ **do**
    $P_{i+1} \leftarrow P_i \cdot Y_i^{x_i}$
    $X_{i+1} \leftarrow X_i \textbf{ div } 2$
    $Y_{i+1} \leftarrow Y_i^2$
    $x_{i+1} \leftarrow X_{i+1} \textbf{ mod } 2$
od

Beweisen Sie durch vollständige Induktion über i die Schleifeninvariante

$$\forall\ i \in \mathbb{N}_0 : P_i \cdot Y_i^{X_i} = b^a$$

1 Übungsblätter

2 div und mod Rechnung

3 Algorithmen

4 Schleifeninvarianzen

5 Schluss

- Was **div** und **mod** bedeuten.
- Wie man mit **mod** 2 überprüfen kann, ob eine Zahl gerade ist.
- Wie man unnötig langen Code erkennt.
- Was eine Schleifeninvariante ist.
- Wie man eine Schleifeninvariante beweisen kann.

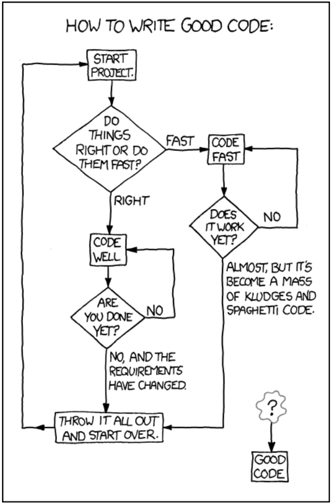

Abbildung: `http://www.xkcd.com`

Kontakt via E-Mail an Philipp Basler oder Nils Braun
gbi.ugroup.hostzi.com