GBI Tutorium Nr. 2^5

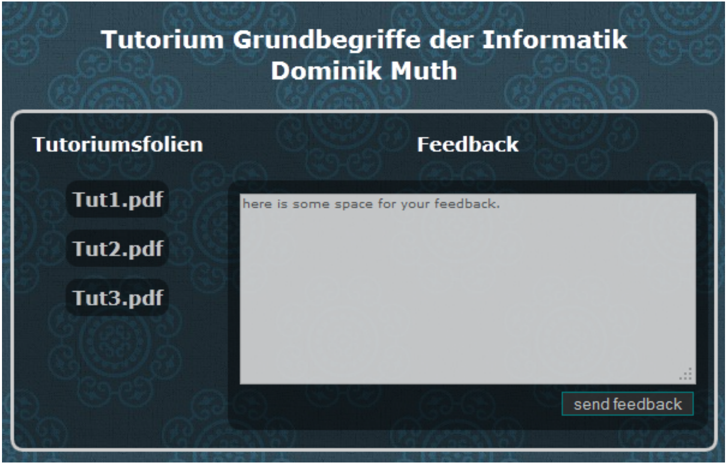
Tutorium 4

Dominik Muth - dominik.muth@student.kit.edu | 7. November 2012

INSTITUT FÜR INFORMATIK

Outline/Gliederung

Feedback und Folien



domi-individual.bplaced.de/tut/

Übungsblatt 3

Karlsruher Institut für Technologie

Aufgabe 3.3 a)

$|L_1^2| = |L_1 \times L_1|$

Aufgabe 3.3 c)

$(L_1 \cup L_2)^* = (L_1^* \cdot L_2^*)^*$

Übungsblatt 3

Aufgabe 3.3 a)

$$|L_1^2| = |L_1 \times L_1|$$

Aufgabe 3.3 c)

$$(L_1 \cup L_2)^* = (L_1^* \cdot L_2^*)^*$$

Wiederholung - Quiz

- A^* ist eine formale Sprache!
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$.
- Eine bijektive Relation ist eine Funktion.
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv.
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$.
- Eine bijektive Relation ist eine Funktion.
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv.
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$ X
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$.
- Eine bijektive Relation ist eine Funktion.
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv.
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

Karlsruher Institut für Technologie

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$ X
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$. $\surd$
- Eine bijektive Relation ist eine Funktion.
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv.
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$ X
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$. $\surd$
- Eine bijektive Relation ist eine Funktion. $\surd$
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv.
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$ X
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$. $\surd$
- Eine bijektive Relation ist eine Funktion. $\surd$
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv. X
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

Wiederholung - Quiz

- A^* ist eine formale Sprache! $\surd$
- $(L_1 \cdot L_2)^* = L_1^* \cdot L_2^*$ X
- $f(x) = x^3 - x^2$ ist rechtstotal für $x, f(x) \in \mathbb{R}$. $\surd$
- Eine bijektive Relation ist eine Funktion. $\surd$
- Wenn $f : A \to B$ injektiv $\Rightarrow f^{-1}$ ist surjektiv. X
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$ $\surd$

Wiederholung - Aufgaben

- Schreiben sie die Injektivität als Prädikatenlogische Formel.
- Es sei $L \subseteq A^*$ eine formale Sprache. Beweisen oder widerlegen Sie: $L^+ \cdot L^+ \subseteq L^+$

Wiederholung - Aufgaben

- Schreiben sie die Injektivität als Prädikatenlogische Formel.
- Es sei $L \subseteq A^*$ eine formale Sprache. Beweisen oder widerlegen Sie: $L^+ \cdot L^+ \subseteq L^+$

Wiederholung

Induktion

Alice und Bob feiern ihren Hochzeitstag. Auf ihrer Party befinden sich $n \in \mathbb{N}_+$ Paare. Dabei begrüßen sich alle Paare mit Ausnahme des eigenen Partners.

a) Geben Sie die Anzahl der Begrüßungen x_i für $i \in \{1, 2, 3, 4, 5\}$ Paare an.

b) Stellen Sie für x_n eine geschlossene Formel (d.h. einen arithmetischen Ausdruck, in dem nur Zahlen, n und die Grundrechenarten vorkommen) auf.

c) Beweisen Sie Ihre Aussage aus Teilaufgabe b) durch vollständige Induktion

Pseudocode

Was ist Pseudocode

Pseudocode ist Programmcode, der zur Darstellung von Algorithmen verwendet wird.

Konvention in GBI

- Zuweisung: $x \leftarrow 42$
- Kommentare: //Kommentar
- Schleifen:
  for $i \leftarrow 0$ *to* 42*do* .... *od*
  while $i < 42$ *do* .... *od*
- Abfrage: *if* $i = 42$*then*.......*endif*

Pseudocode

Karlsruher Institut für Technologie

Was ist Pseudocode

Pseudocode ist Programmcode, der zur Darstellung von Algorithmen verwendet wird.

Konvention in GBI

- Zuweisung: $x \leftarrow 42$
- Kommentare: $//$Kommentar
- Schleifen:
  for $i \leftarrow 0$ *to* 42*do* .... *od*
  while $i < 42$ *do* .... *od*
- Abfrage: *if* $i = 42$*then*.......*endif*

Pseudocode

Was ist Pseudocode

Pseudocode ist Programmcode, der zur Darstellung von Algorithmen verwendet wird.

Konvention in GBI

- Zuweisung: $x \leftarrow 42$
- Kommentare: $//$Kommentar
- Schleifen:
  for $i \leftarrow 0$ *to* 42*do* .... *od*
  while $i < 42$ *do* .... *od*
- Abfrage: *if* $i = 42$*then*.......*endif*

Pseudocode

Karlsruher Institut für Technologie

Was ist Pseudocode

Pseudocode ist Programmcode, der zur Darstellung von Algorithmen verwendet wird.

Konvention in GBI

- Zuweisung: $x \leftarrow 42$
- Kommentare: $//$Kommentar
- Schleifen:
  for $i \leftarrow 0$ *to* 42*do* .... *od*
  while $i < 42$ *do* .... *od*
- Abfrage: *if* $i = 42$*then*.......*endif*

div / mod

Erläuterung div

x div y entspricht der Ganzzahldivision ohne Rest:

$\lfloor \frac{x}{y} \rfloor$

Beispiel: 14 div 3 $=$ 2

Erläuterung mod

x mod y entspricht dem Rest der Ganzzahldivision.

Beispiel:

5 mod 3 $=$ 2

Mod lässt sich auch durch $\cdot$, $-$ und div darstellen:

$x\ mod\ y = x - (y \cdot (x\ div\ y))$

Übung

Fülle folgende Tabelle aus:

x	0	1	2	3	4	5	6	7	8	9	10	11
x **div** 4												
$4(x$ **div** $4)$												
x **mod** 4												

Algorithmen

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Algorithmen

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Algorithmen

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Algorithmen

Karlsruher Institut für Technologie

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Algorithmen

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Algorithmen

Eigenschaften

Ein Algorithmus...

- hat eine endliche Beschreibung,
- besteht aus elementaren Anweisungen,
- ist deterministisch,
- gibt endliche Ausgabe auf endliche Eingabe aus,
- hat endlich viele Schritte,
- ist skalierbar
- und ist nachvollziehbar

Schleifen

Karlsruher Institut für Technologie

while Schleife

Eine while Schleife wiederholt etwas, solange eine Bedingung erfüllt ist.

Pseudocode: **while** $x > 1$ **do** .... **od**

Jave: while($x > 1$) { .... }

do while Schleife

Eine do while Schleife tut erst etwas, prüft danach, ob die Bedingung erfüllt ist und wiederholt dann den schleifenrumpf.

Pseudocode: **do** .... **od while** $x > 1$

Jave: do { .... }while($x > 1$)

Schleifen

Beispiel 1

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

$i \leftarrow 0$

while $x > 1$ **do**

$x \leftarrow x \operatorname{div} 2$

$i \leftarrow i + 1$

od

Output: i

Schleifen

Beispiel 2

$k \leftarrow 0$
for $i \leftarrow 0$ **to** 20 **do**
    $k \leftarrow k + i$
od
Output: k

Schleifen

Beispiel 3

Gegeben sei ein Wort w der Länge $|w| = n$. Das Array W hat an i-ter Stelle den i-ten Buchstabe von w. w ist ϵ-frei.

$c \leftarrow 0$
for $i \leftarrow 0$ **to** $n - 1$ **do**
  if $A[i] =$'x' **then**
    $c \leftarrow c + 1$
  end if
od
Output: c

Schleifen

Karlsruher Institut für Technologie

Übung 1, Winter 2008/2009

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:

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

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

Schleifeninvarianz

Definition

Die Schleifeninvarianz bezeichnet eine Bedingung innerhalb einer Schleife, welche bei jedem Schleifendurchlauf gültig ist.

Schleifeninvarianz

Wofür?

Mit Schleifeninvarianten lässt sich die Korrektheit von Algorithmen beweisen.

Wie

Mit vollständiger Induktion

Schleifeninvarianz

Karlsruher Institut für Technologie

Beispiel

Input: $a, b \in \mathbb{N}_0$

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

Output: S

Übung

Algorithmus mit $a = 3$ und $b = 4$ ausprobieren und Werte für S und Y bei jedem Schleifendurchlauf finden.

Schleifeninvarianz

Beispiel

Input: $a, b \in \mathbb{N}_0$

$S \leftarrow a$

$Y \leftarrow b$

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

$S \leftarrow S - 1$

$Y \leftarrow Y - 1$

od

Output: S

Übung

Algorithmus mit $a = 3$ und $b = 4$ ausprobieren und Werte für S und Y bei jedem Schleifendurchlauf finden.

Schleifeninvarianz

Übung

Gegeben sei folgendes Programmstück:

$X_0 \leftarrow 2$
$Y_0 \leftarrow 5$
for $i \leftarrow 0$ **to** n **do**
  $j \leftarrow i$
  $Y_{j+1} \leftarrow 5Y_j - 6X_j$
  $X_{j+1} \leftarrow Y_j$
od

Beweisen Sie die Korrektheit der folgenden Aussage:

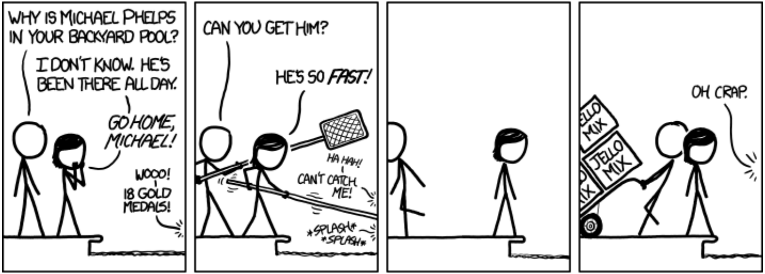
$Y_j = 2^{j+1} + 3^{j+1} \wedge X_j = 2^j + 3^j$

Fragen

Karlsruher Institut für Technologie

- Fragen zum Stoff?
- Fragen zum nächsten Übungsblatt?
- Generelle Fragen?

EOF



source : http : / / imgs.xkcd.com / comics / michael$_p$helps.png