

# Grundbegriffe der Informatik - Tutorium 21

Christian Jülg  
Wintersemester 2012/13  
16. November 2011

<http://gbi-tutor.blogspot.com>

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

Eine formale Sprache  $L$ ...

1. ... ist eine Menge von Wörtern
2. ... basiert immer auf einem Alphabet  $A = \{a, b\}$
3. ... kann gleich einem Wort  $w$  sein

Das Produkt zweier formaler Sprachen...

1. ... ist kommutativ
2. ... ist nicht für die Sprachen  $L_1 = \{\}$  und  $L_2 = \{\epsilon\}$  definiert
3. ... ist gleich ihrer Konkatenation

Vollständige Induktion...

1. ... ist für jeden Beweis das beste Beweisverfahren
2. ... besteht immer aus einem Anfang, der Voraussetzung und einem Schritt
3. ... sollte ich mittlerweile im Schlaf beherrschen.

Eine formale Sprache  $L$ ...

1. ... ist eine Menge von Wörtern
2. ... basiert immer auf einem Alphabet  $A = \{a, b\}$
3. ... kann gleich einem Wort  $w$  sein

Das Produkt zweier formaler Sprachen...

1. ... ist kommutativ
2. ... ist nicht für die Sprachen  $L_1 = \{\}$  und  $L_2 = \{\epsilon\}$  definiert
3. ... ist gleich ihrer Konkatenation

Vollständige Induktion...

1. ... ist für jeden Beweis das beste Beweisverfahren
2. ... besteht immer aus einem Anfang, der Voraussetzung und einem Schritt
3. ... sollte ich mittlerweile im Schlaf beherrschen.

Eine formale Sprache  $L$ ...

1. ... ist eine Menge von Wörtern
2. ... basiert immer auf einem Alphabet  $A = \{a, b\}$
3. ... kann gleich einem Wort  $w$  sein

Das Produkt zweier formaler Sprachen...

1. ... ist kommutativ
2. ... ist nicht für die Sprachen  $L_1 = \{\}$  und  $L_2 = \{\epsilon\}$  definiert
3. ... ist gleich ihrer Konkatenation

Vollständige Induktion...

1. ... ist für jeden Beweis das beste Beweisverfahren
2. ... besteht immer aus einem Anfang, der Voraussetzung und einem Schritt
3. ... sollte ich mittlerweile im Schlaf beherrschen.

Eine formale Sprache  $L$ ...

1. ... ist eine Menge von Wörtern
2. ... basiert immer auf einem Alphabet  $A = \{a, b\}$
3. ... kann gleich einem Wort  $w$  sein

Das Produkt zweier formaler Sprachen...

1. ... ist kommutativ
2. ... ist nicht für die Sprachen  $L_1 = \{\}$  und  $L_2 = \{\epsilon\}$  definiert
3. ... ist gleich ihrer Konkatenation

Vollständige Induktion...

1. ... ist für jeden Beweis das beste Beweisverfahren
2. ... besteht immer aus einem Anfang, der Voraussetzung und einem Schritt
3. ... sollte ich mittlerweile im Schlaf beherrschen.

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss



# leider erst nächste Woche...

etwas Statistik

- 23 von 26 Abgaben!
- durchschnittliche Punktzahl: 15,6/21 Punkten

häufige Fehler...

3.2: in die IV gehört die komplette Behauptung

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

## Blatt 4

- Abgabe: 18.11.2011 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 19

## Themen

- Algorithmen
- Schleifen-Invarianten

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

## Aus der Vorlesung

Algorithmen haben folgende Eigenschaften:

- endliche Beschreibung (Wort über einem Alphabet)
- elementare Aussagen (effektiv in einem Schritt ausführbar)
- Determinismus (nächste Anweisung ist festgelegt)
- endliche Eingabe errechnet endliche Ausgabe
- endlich viele Schritte
- funktioniert für beliebig große Eingaben
- nachvollziehbar/verständlich

## DIV und MOD

- $a \text{ div } b$  ist das Ergebnis der ganzzahligen Division  $a/b$
- $a \text{ mod } b$  ist der Rest der ganzzahligen Division  $a/b$

## Ziel

- Bestimmte Berechnungen sollen wiederholt ausgeführt werden, bis eine bestimmte Bedingung eintritt.
- Dies wird u.a. durch eine for-Schleife (Zählschleife) erreicht.

## Ziel

- Bestimmte Berechnungen sollen wiederholt ausgeführt werden, bis eine bestimmte Bedingung eintritt.
- Dies wird u.a. durch eine for-Schleife (Zählschleife) erreicht.

## Syntax der for-Schleife in Pseudocode

```
for  $i \leftarrow 0$  to 10 do //zählt von 0 bis 10  
... (Schleifeninhalt)
```

## Ziel

- Bestimmte Berechnungen sollen wiederholt ausgeführt werden, bis eine bestimmte Bedingung eintritt.
- Dies wird u.a. durch eine for-Schleife (Zählschleife) erreicht.

## Syntax der for-Schleife in Pseudocode

```
for  $i \leftarrow 0$  to 10 do //zählt von 0 bis 10  
... (Schleifeninhalt)
```

## Syntax der for-Schleife in Java

```
for (int  $i = 0$ ;  $i \leq 10$ ;  $i++$ ) //zählt von 0 bis 10  
... (Schleifeninhalt)
```



Als Schleifeninvariante werden Eigenschaften einer Schleife bezeichnet, die zu einem bestimmten Punkt bei jedem Durchlauf gültig sind, unabhängig von der Zahl ihrer derzeitigen Durchläufe. Typischerweise enthalten Schleifeninvarianten Wertebereiche von Variablen und Beziehungen der Variablen untereinander.

## Sinn und Zweck

### Schleifeninvarianten...

- sind Aussagen, die am Anfang und am Ende eines Schleifendurchlaufes gelten
- helfen, die Korrektheit eines Programmes zu beweisen
- beweist man meist durch Induktion

einfaches 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  $S$  //Ausgabe von  $S$  als Ergebnis

Funktion

Was macht dieses Programm?

## einfaches Beispiel

//Eingaben  $a, b \in N_0$

$S \leftarrow a$

$Y \leftarrow b$

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

$S \leftarrow S + 1$

$Y \leftarrow Y - 1$

od

Output  $S$  //Ausgabe von  $S$  als Ergebnis

## Funktion

Was macht dieses Programm?

Es berechnet die Summe von  $a$  und  $b$ .

einfaches Beispiel

$S \leftarrow a$

$Y \leftarrow b$

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

$S \leftarrow S + 1$

$Y \leftarrow Y - 1$

od

Wertetabelle für  $a = 6$  und  $b = 4$

	$S$	$Y$
	6	4
$i = 0$	7	3
$i = 1$	8	2
$i = 2$	9	1
$i = 3$	10	0

einfaches Beispiel

$S \leftarrow a$

$Y \leftarrow b$

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

$S \leftarrow S + 1$

$Y \leftarrow Y - 1$

od

Wertetabelle für  $a = 6$  und  $b = 4$

	$S$	$Y$	
	6	4	
$i = 0$	7	3	$S + Y = a + b$
$i = 1$	8	2	
$i = 2$	9	1	
$i = 3$	10	0	

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

Suchalgorithmen sind

Algorithmen, die etwas über das Vorkommen eines Zeichens  $x \in A$  in einem Wort  $w \in A^*$  aussagen.

## Aufgabe 1

Entwerfe einen Algorithmus, der berechnet, ob  $x$  in  $w$  vorkommt!



## Aufgabe 1

Entwerfe einen Algorithmus, der berechnet, ob  $x$  in  $w$  vorkommt!

## Lösung

```
 $p \leftarrow -1$   
for  $i \leftarrow 0$  to  $n - 1$  do  
   $p \leftarrow \begin{cases} 1 & \text{falls } w(i) = x \\ p & \text{sonst} \end{cases}$ 
```

## Aufgabe 2

Entwerfe einen Algorithmus, der die letzte Stelle im Wort, an der x in w vorkommt, berechnet!

## Aufgabe 2

Entwerfe einen Algorithmus, der die letzte Stelle im Wort, an der  $x$  in  $w$  vorkommt, berechnet!

## Lösung

```
 $p \leftarrow -1$   
for  $i \leftarrow 0$  to  $n - 1$  do  
   $p \leftarrow \begin{cases} i & \text{falls } w(i) = x \\ p & \text{sonst} \end{cases}$ 
```

## Aufgabe 3

Entwerfe einen Algorithmus, der die erste Stelle im Wort, an der x in w vorkommt, berechnet!

## Aufgabe 3

Entwerfe einen Algorithmus, der die erste Stelle im Wort, an der  $x$  in  $w$  vorkommt, berechnet!

## Lösung

```
 $p \leftarrow -1$   
for  $i \leftarrow 0$  to  $n - 1$  do  
   $p \leftarrow \begin{cases} i & \text{falls } w(i) = x \wedge p < 0 \\ p & \text{sonst} \end{cases}$ 
```

## Beweis der Korrektheit eines Programms (Aufgabe 2)

Durch Induktion wird gezeigt, dass nach den ersten  $k$  Schleifendurchläufen  $p$  die letzte Position von  $x$  in den ersten  $k$  Zeichen von  $w$  ist.

## Beweis der Korrektheit eines Programms (Aufgabe 2)

Durch Induktion wird gezeigt, dass nach den ersten  $k$  Schleifendurchläufen  $p$  die letzte Position von  $x$  in den ersten  $k$  Zeichen von  $w$  ist.

### Beweis

- Induktionsanfang:  $k=0$ ,  $p=-1$  ist wahr.
- Induktionsannahme: für ein festes  $k < |w|$  gilt: Nach den ersten  $k$  Schleifendurchläufen ist  $p$  die Position des letzten  $x$  in den ersten  $k$  Zeichen von  $w$ .
- Induktionsschritt:  $k \rightarrow k + 1$   
Wir betrachten den  $k+1$ ten Schleifendurchlauf, während dem das Zeichen  $w(k)$  betrachtet wird (2 Fälle!).

Fall 1:  $w(k) = x$

Die Position des letzten  $x$  ist unter den ersten  $k+1$  Zeichen jetzt die Position  $k+1$ . Nach Induktionsannahme gilt zu Beginn des Schleifendurchlaufs:  $p$  ist die letzte Position von  $x$  unter den ersten  $k$  Zeichen von  $w$ . Aufgrund des Programmes wird  $p$  nun  $k+1$ , so dass am Ende des  $k+1$ ten Schleifendurchlaufs gilt:  $p$  ist die Position des letzten  $x$  unter den ersten  $k+1$  Zeichen von  $w$ .



Fall 2:  $w(k) \neq x$ :

Die letzte Position von  $x$  ist unter den ersten  $k+1$  Zeichen gleich der letzten Position von  $x$  unter den ersten  $k$  Zeichen von  $w$ . Nach Induktionsannahme gilt zu Beginn des Schleifendurchlaufs:  $p$  ist die letzte Position von  $x$  unter den ersten  $k$  Zeichen von  $w$ . Aufgrund des Programmes bleibt  $p$  nun gleich, so dass am Ende des  $k+1$ ten Schleifendurchlaufs gilt:  $p$  ist die letzte Position von  $x$  unter den ersten  $k+1$  Zeichen von  $w$ .

Damit ist die Behauptung gezeigt.

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

Was ist ein Palindrom?

Palindrome sind Wörter, die von vorne gelesen das gleiche Wort ergeben, wie von hinten gelesen.

## Aufgabe 5

Schreibe ein Programm, das für Wörter  $w$  untersucht, ob  $w$  ein Palindrom ist!

## Aufgabe 5

Schreibe ein Programm, das für Wörter  $w$  untersucht, ob  $w$  ein Palindrom ist!

## Lösung

```
 $p \leftarrow 1$   
for  $i \leftarrow 0$  to  $n - 1$  do  
   $p \leftarrow \begin{cases} p & \text{falls } w(i) = w(n - 1 - i) \\ -1 & \text{sonst} \end{cases}$ 
```

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

## Die Theorie

Der Beweis erfolgt in folgenden Schritten:

1. Induktionsanfang: Die Aussage wird für  $n = n_0$  gezeigt
2. Induktionsvoraussetzung/-annahme: Die Aussage sei für **ein**  $n$  wahr.
3. Induktionsschluss/-schritt: Aus dem Schluss von  $n$  auf  $n + 1$  (in der Regel mit Hilfe der IV) folgt, dass die Aussage für alle natürlichen Zahlen  $n > n_0$  gilt.

## Die Theorie

Der Beweis erfolgt in folgenden Schritten:

1. Induktionsanfang: Die Aussage wird für  $n = n_0$  gezeigt
2. Induktionsvoraussetzung/-annahme: Die Aussage sei für **ein**  $n$  wahr.
3. Induktionsschluss/-schritt: Aus dem Schluss von  $n$  auf  $n + 1$  (in der Regel mit Hilfe der IV) folgt, dass die Aussage für alle natürlichen Zahlen  $n > n_0$  gilt.

## Ein Beispiel:

Beweise durch vollständig Induktion  $1 + 2 + 3 + \dots + n = \frac{n \cdot (n+1)}{2}$ :



# Beweisverfahren der vollständigen Induktion

Ein Beispiel:

Beweise durch vollständig Induktion  $1 + 2 + 3 + \dots + n = \frac{n*(n+1)}{2}$ :

IA  $n = 1$ :  $1 = \frac{1*(1+1)}{2} = 1$  ist erfüllt

IV  $1 + 2 + 3 + \dots + n = \frac{n*(n+1)}{2}$  gilt für ein  $n \in \mathbb{N}$

IS

$$\begin{aligned} & 1 + 2 + 3 + \dots + (n + 1) \\ &= 1 + 2 + 3 + \dots + n + (n + 1) \\ &\stackrel{IV}{=} \frac{n * (n + 1)}{2} + (n + 1) \\ &= \frac{(n + 1) * (n + 2) + 2(n + 1)}{2} = \frac{n^2 + 3n + 2}{2} \\ &= \frac{(n + 1) * (n + 2)}{2} \end{aligned}$$

# Und weils so schön ist...

Ihr schon wieder...

Es sei  $n \in \mathbf{N}$  und  $a, b \in \mathbf{R}$ . Beweist durch vollständige Induktion: Für  $f(x) = e^{ax+b}$  gilt  $f^{(n)} = a^n * e^{ax+b}$

Aufwachen!

Aufgabenblatt 3

Aufgabenblatt 4

Definitionen

Suchalgorithmen für Wörter

Palindromtest

Vollständige Induktion

Abschluss

# Zum Schluss...

Was ihr nun wissen solltet!

# Zum Schluss...

Was ihr nun wissen solltet!

- Was ist eine for-Schleife?

# Zum Schluss...

Was ihr nun wissen solltet!

- Was ist eine for-Schleife?
- Was ist eine Schleifeninvariante?

# Zum Schluss...

Was ihr nun wissen solltet!

- Was ist eine for-Schleife?
- Was ist eine Schleifeninvariante?
- Wie entwirft man einen Suchalgorithmus?

Was ihr nun wissen solltet!

- Was ist eine for-Schleife?
- Was ist eine Schleifeninvariante?
- Wie entwirft man einen Suchalgorithmus?
- Wie funktioniert ein Korrektheitsbeweis



Was ihr nun wissen solltet!

- Was ist eine for-Schleife?
- Was ist eine Schleifeninvariante?
- Wie entwirft man einen Suchalgorithmus?
- Wie funktioniert ein Korrektheitsbeweis
- Was sind Palindrome? Vorgehen beim Algorithmenentwurf.

Ihr wisst was nicht?

Stellt **jetzt** Fragen!

