

Grundbegriffe der Informatik - Tutorium

– Wintersemester 2011/12 –

Christian Jülg

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

11. Januar 2012



Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Quellennachweis & Dank an:
Martin Schadow, Susanne Dinkler, Tobias Dencker, Sebastian Heßlinger,
Joachim Wilke

Übersicht



- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten
- 6 Abschluss

- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten
- 6 Abschluss

Zum Warmwerden...



Algorithmen-Effizienz...

- ① ... wird häufig in Abhängigkeit der Eingabelänge angegeben.
- ② ... ist unabhängig von der Struktur der eingegebenen Daten.
- ③ ... muss für jede Rechenmaschine einzeln ermittelt werden.

Das O-Kalkül ...

- ① ... eignet sich gut um einen Mindestaufwand anzugeben.
- ② ... ist unabhängig von einfachen Faktoren.
- ③ ... beschreibt eine Menge von Funktionen.

Das Θ -Kalkül...

- ① ... gibt einen „Korridor“ an, den der Algorithmus nie verlässt.
- ② ... $\Theta(f(n))$ enthält alle Funktionen, die auch in $O(f(n))$ enthalten sind.
- ③ ... ist reflexiv (Es gilt: $f(n) \in \Theta(f(n))$).



Zum Warmwerden...



Algorithmen-Effizienz...

- ① ... wird häufig in Abhängigkeit der Eingabelänge angegeben.
- ② ... ist unabhängig von der Struktur der eingegebenen Daten.
- ③ ... muss für jede Rechenmaschine einzeln ermittelt werden.

Das O-Kalkül ...

- ① ... eignet sich gut um einen Mindestaufwand anzugeben.
- ② ... ist unabhängig von einfachen Faktoren.
- ③ ... beschreibt eine Menge von Funktionen.

Das Θ -Kalkül...

- ① ... gibt einen „Korridor“ an, den der Algorithmus nie verlässt.
- ② ... $\Theta(f(n))$ enthält alle Funktionen, die auch in $O(f(n))$ enthalten sind.
- ③ ... ist reflexiv (Es gilt: $f(n) \in \Theta(f(n))$).

Zum Warmwerden...



Algorithmen-Effizienz...

- ① ... wird häufig in Abhängigkeit der Eingabelänge angegeben.
- ② ... ist unabhängig von der Struktur der eingegebenen Daten.
- ③ ... muss für jede Rechenmaschine einzeln ermittelt werden.

Das O-Kalkül ...

- ① ... eignet sich gut um einen Mindestaufwand anzugeben.
- ② ... ist unabhängig von einfachen Faktoren.
- ③ ... beschreibt eine Menge von Funktionen.

Das Θ -Kalkül...

- ① ... gibt einen „Korridor“ an, den der Algorithmus nie verlässt.
- ② ... $\Theta(f(n))$ enthält alle Funktionen, die auch in $O(f(n))$ enthalten sind.
- ③ ... ist reflexiv (Es gilt: $f(n) \in \Theta(f(n))$).

Zum Warmwerden...



Algorithmen-Effizienz...

- ① ... wird häufig in Abhängigkeit der Eingabelänge angegeben.
- ② ... ist unabhängig von der Struktur der eingegebenen Daten.
- ③ ... muss für jede Rechenmaschine einzeln ermittelt werden.

Das O-Kalkül ...

- ① ... eignet sich gut um einen Mindestaufwand anzugeben.
- ② ... ist unabhängig von einfachen Faktoren.
- ③ ... beschreibt eine Menge von Funktionen.

Das Θ -Kalkül...

- ① ... gibt einen „Korridor“ an, den der Algorithmus nie verlässt.
- ② ... $\Theta(f(n))$ enthält alle Funktionen, die auch in $O(f(n))$ enthalten sind.
- ③ ... ist reflexiv (Es gilt: $f(n) \in \Theta(f(n))$).

- 1 Aufwachen
- 2 Aufgabenblatt 9**
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten
- 6 Abschluss

Aufgabenblatt 9



Blatt 9

- Abgaben: 14 / 24
- Punkte: Durchschnitt 12,1 von 18

Probleme

- 9.1: $f(n) \in \theta(g(n))$ bedeutet, dass es **zwei** geeignete Konstanten c_1 und c_2 gibt
- 9.3: n^2 und $O(n^2)$ sind nicht das gleiche

- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10**
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten
- 6 Abschluss

Aufgabenblatt 10



Blatt 10

- Abgabe: 13.01.2012 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 23

Themen

- Rekursion
- Master-Theorem
- Endliche Automaten
- Mealy, Moore, endl. Akzeptoren

- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz**
- 5 Endliche Automaten
- 6 Abschluss

Aufwandsklassen



Fallunterscheidung: Aufwandsklassen

- O -Kalkül Obere Schranke, die der Algorithmus erreichen, aber nicht überschreiten kann
- Ω -Kalkül Untere Schranke und ein “Mindestaufwand“, den der Algorithmus hat
- Θ -Kalkül Schnittmenge der Betrachtung aus $\Omega(n)$ und $O(n)$. Es entsteht eine Art „Korridor“, den der Algorithmus nie verlässt.



O-Kalkül



Definition

$$O(g(n)) = \{f(n) \mid \exists c > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$

Umgangssprachlich

$O(g(n))$ enthält alle nicht-negativen Funktionen, die höchstens so schnell wie $g(n)$ wachsen.

Dabei kümmern wir uns nicht

- darum, was am Anfang passiert ($\exists n_0 \in \mathbb{N} \dots \forall n \geq n_0$).
- um einfache Faktoren ($\exists c \in \mathbb{R} \dots c \cdot g(n)$).

Aufwandsklassen



Obere asymptotische Schranke

$$O(g(n)) = \{f(n) \mid$$

$$\exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \forall n > n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$

Untere asymptotische Schranke

$$\Omega(g(n)) = \{f(n) \mid$$

$$\exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \forall n > n_0 : 0 \leq c \cdot g(n) \leq f(n)\}$$

Asymptotisch scharfe Schranke

$$\Theta(g(n)) = \{f(n) \mid$$

$$\exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N} \forall n > n_0 : 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)\}$$

Beachte:

Alle Kalküle geben eine **Menge** von Funktionen an. $f(n) = O(n^2)$ bedeutet also eigentlich $f(n) \in O(n^2)$!

Rechenregeln



Reflexivität

- $f(n) \in O(f(n))$
- $g(n) \in \Omega(g(n))$
- $h(n) \in \Theta(h(n))$

Symmetrie

Hier gilt nur: $f(n) \in \Theta(g(n)) \Leftrightarrow g(n) \in \Theta(f(n))$

asymptotisches Wachstum

- $O(n^2 + n + \log(n)) = O(n^2)$
- $\Omega(n^2 + n + \log(n)) = \Omega(n^2) \subset \Omega(\log(n))$

Beispiel



Es gilt nicht:

$$f(n) \notin \Theta(g(n)) \Rightarrow g(n) \in O(f(n)) \vee f(n) \in O(g(n))$$

Sucht Gegenbeispiele:

Beispiel



Es gilt nicht:

$$f(n) \notin \Theta(g(n)) \Rightarrow g(n) \in O(f(n)) \vee f(n) \in O(g(n))$$

Sucht Gegenbeispiele:

- $|\cos(n)| * n$ und

Beispiel



Es gilt nicht:

$$f(n) \notin \Theta(g(n)) \Rightarrow g(n) \in O(f(n)) \vee f(n) \in O(g(n))$$

Sucht Gegenbeispiele:

- $|\cos(n)| * n$ und n
- n und

Beispiel



Es gilt nicht:

$$f(n) \notin \Theta(g(n)) \Rightarrow g(n) \in O(f(n)) \vee f(n) \in O(g(n))$$

Sucht Gegenbeispiele:

- $|\cos(n)| * n$ und n
- n und $f(n) = n$ für gerade, 0 für ungerade Werte von n

- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten**
- 6 Abschluss

Endlich ein Automat!



Wozu?

Ein endlicher Automat ist gerade mächtig genug, um einen regulären Ausdruck zu erkennen. Der Vorteil von endlichen Automaten ist, dass sie sehr einfach zu implementieren sind.

Endlich ein Automat!



Wozu?

Ein endlicher Automat ist gerade mächtig genug, um einen regulären Ausdruck zu erkennen. Der Vorteil von endlichen Automaten ist, dass sie sehr einfach zu implementieren sind.

Was braucht man?

- endliche Menge Z von Zuständen
- ein Anfangszustand $z_0 \in Z$
- ein Eingabealphabet X
- Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$
- ein Ausgabealphabet Y
- eine Ausgabefunktion (abhängig vom Typ des Automaten)

Endlich ein Automat!



Wie arbeitet er?

Das Lesen eines Zeichens $x \in X$ führt zu einem Zustandsübergang vom aktuellen Zustand $z \in Z$ in einen neuen Zustand $z' \in Z$

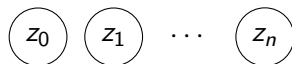
- Notation: $f(z, x) = z'$
- Der Zustand läßt sich als ein **Gedächtnis** über die Vorgeschichte, also die bisher eingegebenen Zeichen, auffassen.
Dieses ist leider nur **endlich** (endliche Menge an Zuständen!)

Darstellung von endlichen Automaten als Graphen



Zustandsmenge

$Z = \{z_0, z_1, \dots, z_n\}$ des
endlichen Automaten lassen sich
als Ecken eines Graphen auffassen

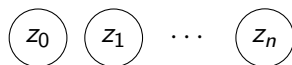


Darstellung von endlichen Automaten als Graphen



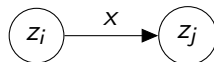
Zustandsmenge

$Z = \{z_0, z_1, \dots, z_n\}$ des
endlichen Automaten lassen sich
als Ecken eines Graphen auffassen



Zustandsübergänge

$f(z_i, x) = z_j$ mit $x \in X$
entsprechen markierten
gerichteten Kanten

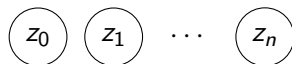


Darstellung von endlichen Automaten als Graphen



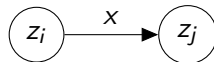
Zustandsmenge

$Z = \{z_0, z_1, \dots, z_n\}$ des
endlichen Automaten lassen sich
als Ecken eines Graphen auffassen



Zustandsübergänge

$f(z_i, x) = z_j$ mit $x \in X$
entsprechen markierten
gerichteten Kanten



Ein im endlichen Automaten erreichter Zustand z_k ist durch den Anfangszustand z_0 und die bisher eingegebene Zeichenreihe $w \in X^*$ mit $w = x_1 \dots x_i$ bestimmt

f^* und f^{**}  f^*

Nach Eingabe des ganzen Wortes $w \in X^*$ erreichen wir den Zustand $f^* : Z \times X^* \rightarrow Z$ mit

$$f^*(z, \epsilon) = z$$

$$\forall w \in X^* : \forall x \in X : f^*(z, wx) = f(f^*(z, w), x)$$

f^* und f^{**}



f^*

Nach Eingabe des ganzen Wortes $w \in X^*$ erreichen wir den Zustand $f^* : Z \times X^* \rightarrow Z$ mit

$$f^*(z, \epsilon) = z$$

$$\forall w \in X^* : \forall x \in X : f^*(z, wx) = f(f^*(z, w), x)$$

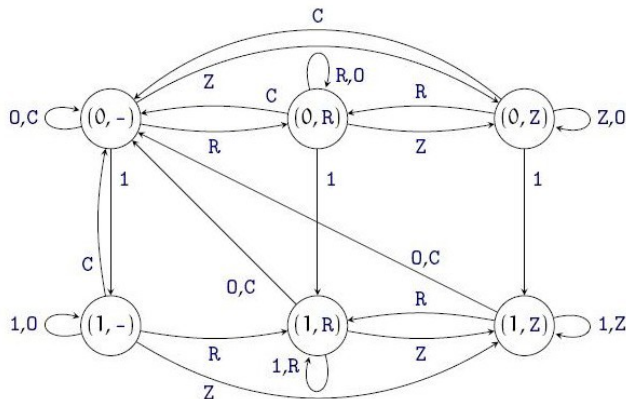
f^{**}

Nach Eingabe des ganzen Wortes $w \in X^*$ haben wir die Zustände $f^{**} : Z \times X^* \rightarrow Z^*$ durchlaufen, mit

$$f^{**}(z, \epsilon) = z$$

$$\forall w \in X^* : x \in X : f^{**}(z, wx) = f^{**}(z, w)f(f^*(z, w), x)$$

Ein Beispielautomat...



Was ist $f^*((0, -), R10)$? Was ist $f^{**}((0, -), R10)$?

Arten von Automaten



Es gibt zwei Arten, wie ein Automat eine Ausgabe tätigen kann.
Wir unterscheiden dabei:

Arten von Automaten



Es gibt zwei Arten, wie ein Automat eine Ausgabe tätigen kann.
Wir unterscheiden dabei:

Mealy-Automat

- Erzeugung einer Ausgabe bei jedem Zustandsübergang
- Ausgabefunktion $g : Z \times X \rightarrow Y^*$
- Markieren der Kanten mit $x_i|y_i$

Arten von Automaten



Es gibt zwei Arten, wie ein Automat eine Ausgabe tätigen kann.
Wir unterscheiden dabei:

Mealy-Automat

- Erzeugung einer Ausgabe bei jedem Zustandsübergang
- Ausgabefunktion $g : Z \times X \rightarrow Y^*$
- Markieren der Kanten mit $x_i|y_i$

Moore-Automat

- Erzeugung einer Ausgabe bei Erreichen eines Zustands
- Ausgabefunktion $h : Z \rightarrow Y^*$

In beiden Fällen ist die Ausgabe ein Wort $y = y_0 \dots y_{n-1}$ über einem Ausgabealphabet Y .

Mealy-Automat



Für die Ausgabefunktion $g : Z \times X \rightarrow Y^*$ lassen sich analog zur Zustandsübergangsfunktion $g^* : Z \times X^* \rightarrow Y^*$ und $g^{**} : Z \times X^* \rightarrow Y^*$ definieren:

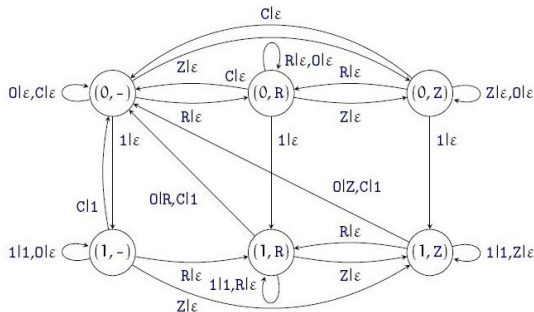
$$g^*(z, \epsilon) = \epsilon$$

$$g^*(z, wx) = g(f^*(z, w), x)$$

$$g^{**}(z, \epsilon) = \epsilon$$

$$g^{**}(z, xw) = g(z, x) \cdot g^{**}(f(z, x), w)$$

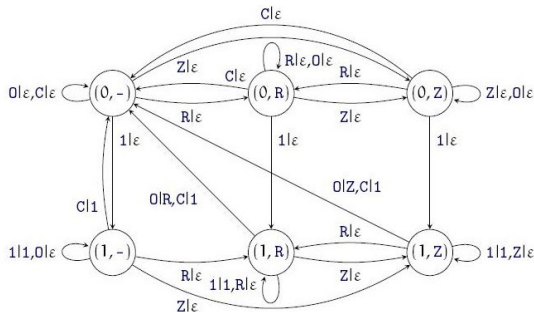
Noch ein Beispielautomat...



Was ist...

- $g^*((0, -), R1O)?$
- $g^{**}((0, -), R1O)?$
- $g^{**}((0, -), R11O)?$

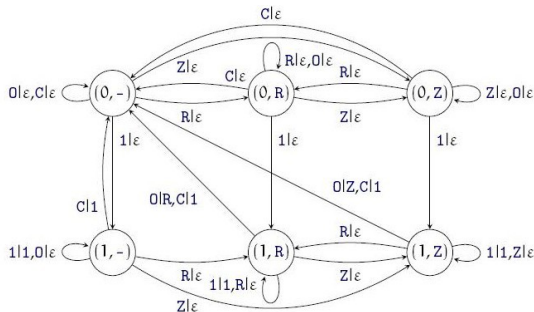
Noch ein Beispielautomat...



Was ist...

- $g^*((0, -), R1O) = R$
- $g^{**}((0, -), R1O)$
- $g^{**}((0, -), R11O)$

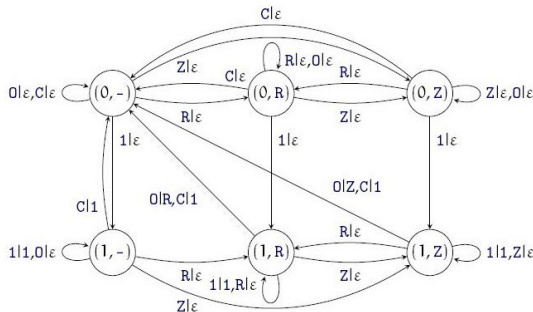
Noch ein Beispielautomat...



Was ist...

- $g^*((0, -), R1O) = R$
- $g^{**}((0, -), R1O) = R$
- $g^{**}((0, -), R11O)$

Noch ein Beispielautomat...



Was ist...

- $g^*((0, -), R1O) = R$
- $g^{**}((0, -), R1O) = R$
- $g^{**}((0, -), R11O) = 1R$

Ihr seid dran...



Entwickelt einen Mealy-Automaten, der...

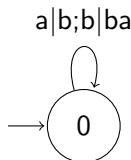
- nur ein Zustand z hat und $X = Y = \{a, b\}$, $g(z, a) = b$ und $g(z, b) = ba$ erfüllt
 - wie sieht $w_1 = g^{**}(z, a)$ aus?
 - $w_2 = g^{**}(z, w_1)$, $\dots w_{i+1} = g^{**}(z, w_i)$?
 - könnte man den Automaten mit weniger Zuständen darstellen?
- und einen weiteren Automaten mit $Z = \mathbb{G}_5$, $X = \{a, b\}$, $Y = \{0, 1\}$, bei b gleicher Zustand und Ausgabe 0, bei a einen Zustand weiter und bei jedem 5. a Ausgabe 1, sonst Ausgabe 0. Was tut der Automat?

Ihr seid dran...



Entwickelt einen Mealy-Automaten, der...

- nur ein Zustand z hat und $X = Y = \{a, b\}$, $g(z, a) = b$ und $g(z, b) = ba$ erfüllt
 - wie sieht $w_1 = g^{**}(z, a)$ aus?
 - $w_2 = g^{**}(z, w_1)$, $\dots w_{i+1} = g^{**}(z, w_i)$?
 - könnte man den Automaten mit weniger Zuständen darstellen?



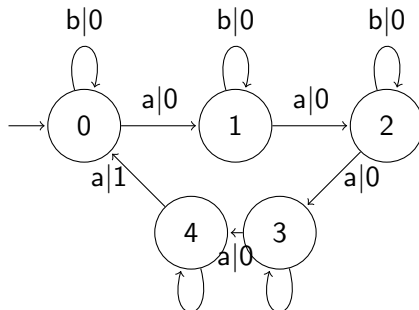
- und einen weiteren Automaten mit $Z = \mathbb{G}_5$, $X = \{a, b\}$, $Y = \{0, 1\}$, bei b gleicher Zustand und Ausgabe 0, bei a einen Zustand weiter und bei jedem 5. a Ausgabe 1, sonst Ausgabe 0. Was tut der Automat?

Ihr seid dran...



Entwickelt einen Mealy-Automaten, der...

- nur ein Zustand z hat und $X = Y = \{a, b\}$, $g(z, a) = b$ und $g(z, b) = ba$ erfüllt
- und einen weiteren Automaten mit $Z = \mathbb{G}_5$, $X = \{a, b\}$, $Y = \{0, 1\}$, bei b gleicher Zustand und Ausgabe 0, bei a einen Zustand weiter und bei jedem 5. a Ausgabe 1, sonst Ausgabe 0. Was tut der Automat?



Der Akzeptor



- Ist der häufigste **Spezialfall** eines Moore-Automaten
- Eine Ausgabe findet nicht bei allen Zuständen statt

Der Akzeptor



- Ist der häufigste **Spezialfall** eines Moore-Automaten
- Eine Ausgabe findet nicht bei allen Zuständen statt
- Die Zustände $F \subseteq Z$, bei denen eine Ausgabe (immer ein Bit lang) erfolgt, heißen akzeptierende Zustände
Es gilt $F = \{z | h(z) = 1\}$

Der Akzeptor



- Ist der häufigste **Spezialfall** eines Moore-Automaten
- Eine Ausgabe findet nicht bei allen Zuständen statt
- Die Zustände $F \subseteq Z$, bei denen eine Ausgabe (immer ein Bit lang) erfolgt, heißen akzeptierende Zustände
Es gilt $F = \{z \mid h(z) = 1\}$
- graphisch werden diese durch Doppelkreise angegeben



Der Akzeptor



- Ist der häufigste **Spezialfall** eines Moore-Automaten
- Eine Ausgabe findet nicht bei allen Zuständen statt
- Die Zustände $F \subseteq Z$, bei denen eine Ausgabe (immer ein Bit lang) erfolgt, heißen akzeptierende Zustände
Es gilt $F = \{z \mid h(z) = 1\}$
- graphisch werden diese durch Doppelkreise angegeben



- Ein Wort $w \in X^*$ wird akzeptiert, wenn gilt $f^*(z_0, w) \in F$
- Die von einem Akzeptor A akzeptierte formale Sprache ist $L(A) = \{w \in X^* \mid f^*(z_0, w) \in F\}$

Ihr seid dran...



Entwickelt einen Akzeptor mit

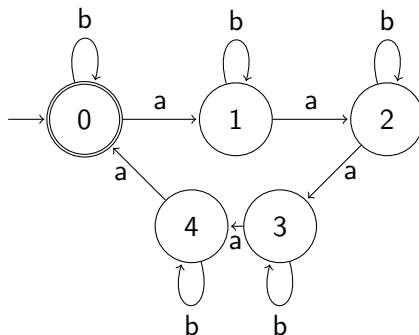
- $X = \{a, b\}$, der alle Wörter akzeptiert, bei denen die Anzahl der a durch 5 teilbar ist. (Anzahl der b ist egal).
- $X = \{a, b\}$, der alle Wörter akzeptiert, in denen nirgends hintereinander zwei b vorkommen.

Ihr seid dran...



Entwickelt einen Akzeptor mit

- $X = \{a, b\}$, der alle Wörter akzeptiert, bei denen die Anzahl der a durch 5 teilbar ist. (Anzahl der b ist egal).



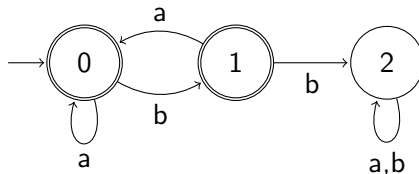
- $X = \{a, b\}$, der alle Wörter akzeptiert, in denen nirgends hintereinander zwei b vorkommen.

Ihr seid dran...



Entwickelt einen Akzeptor mit

- $X = \{a, b\}$, der alle Wörter akzeptiert, bei denen die Anzahl der a durch 5 teilbar ist. (Anzahl der b ist egal).
- $X = \{a, b\}$, der alle Wörter akzeptiert, in denen nirgends hintereinander zwei b vorkommen.



Ihr seid dran...



Entwickelt einen Akzeptor...

- der alle hexadezimalen IP-Adressen der Form *1A.BF.43.0F* akzeptiert
- was ändert sich, wenn man auch Adressen ohne führende 0 akzeptieren möchte?
- bei Langeweile: versucht alle IP-Adressen bei denen die Blöcke aus dezimalen Zahlen zwischen 000 und 255 bestehen zu akzeptieren

- 1 Aufwachen
- 2 Aufgabenblatt 9
- 3 Aufgabenblatt 10
- 4 Algorithmen-Effizienz
- 5 Endliche Automaten
- 6 Abschluss**

Zum Schluss...



Was ihr nun wissen solltet!

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?
- Worin unterscheiden sich Mealy-, Moore-Automaten und endl. Akzeptoren?

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?
- Worin unterscheiden sich Mealy-, Moore-Automaten und endl. Akzeptoren?
- Wie sind f^* , f^{**} , g^* , g^{**} , h^* , h^{**} definiert?

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?
- Worin unterscheiden sich Mealy-, Moore-Automaten und endl. Akzeptoren?
- Wie sind f^* , f^{**} , g^* , g^{**} , h^* , h^{**} definiert?
- Wie könnte man sie auch noch anders definieren?

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?
- Worin unterscheiden sich Mealy-, Moore-Automaten und endl. Akzeptoren?
- Wie sind f^* , f^{**} , g^* , g^{**} , h^* , h^{**} definiert?
- Wie könnte man sie auch noch anders definieren?
- Was haben Automaten mit Sprachen zu tun? Warum sind Automaten relevant?

Zum Schluss...



Was ihr nun wissen solltet!

- Was ist ein (endlicher) Automat?
Aus welchen Teilen besteht er?
- Worin unterscheiden sich Mealy-, Moore-Automaten und endl. Akzeptoren?
- Wie sind f^* , f^{**} , g^* , g^{**} , h^* , h^{**} definiert?
- Wie könnte man sie auch noch anders definieren?
- Was haben Automaten mit Sprachen zu tun? Warum sind Automaten relevant?

Ihr wisst was nicht?

Stellt **jetzt** Fragen!

