

GBI Tutorium Nr. 2⁵

Tutorium 10

Dominik Muth - dominik.muth@student.kit.edu | 9. Januar 2013

INSTITUT FÜR INFORMATIK



- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten
- 4 Moore-Automaten
- 5 Endliche Akzeptoren
- 6 Fragen

- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten
- 4 Moore-Automaten
- 5 Endliche Akzeptoren
- 6 Fragen

- Aus $f \in \Omega(g) \wedge f \in \Theta(g) \Rightarrow f \in \mathcal{O}(g)$
- $n^5 \in \mathcal{O}(2^n)$
- $\frac{n^3+2n}{2n+1} \in \mathcal{O}(n)$
- Alle Algorithmen liegen in $\Omega(1)$

- Aus $f \in \Omega(g) \wedge f \in \Theta(g) \Rightarrow f \in \mathcal{O}(g)$ ✓
- $n^5 \in \mathcal{O}(2^n)$
- $\frac{n^3+2n}{2n+1} \in \mathcal{O}(n)$
- Alle Algorithmen liegen in $\Omega(1)$

- Aus $f \in \Omega(g) \wedge f \in \Theta(g) \Rightarrow f \in \mathcal{O}(g)$ ✓
- $n^5 \in \mathcal{O}(2^n)$ ✓
- $\frac{n^3+2n}{2n+1} \in \mathcal{O}(n)$
- Alle Algorithmen liegen in $\Omega(1)$

- Aus $f \in \Omega(g) \wedge f \in \Theta(g) \Rightarrow f \in \mathcal{O}(g)$ ✓
- $n^5 \in \mathcal{O}(2^n)$ ✓
- $\frac{n^3+2n}{2n+1} \in \mathcal{O}(n)$ ✗
- Alle Algorithmen liegen in $\Omega(1)$

- Aus $f \in \Omega(g) \wedge f \in \Theta(g) \Rightarrow f \in \mathcal{O}(g)$ ✓
- $n^5 \in \mathcal{O}(2^n)$ ✓
- $\frac{n^3+2n}{2n+1} \in \mathcal{O}(n)$ ✗
- Alle Algorithmen liegen in $\Omega(1)$ ✓

- 1 Wiederholung
- 2 Master Theorem**
- 3 Mealy-Automaten
- 4 Moore-Automaten
- 5 Endliche Akzeptoren
- 6 Fragen

Wozu?

Laufzeitabschätzung von rekursiv definierten Funktionen

Grundaufbau

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Erläuterung

- a = Anzahl der Unterprobleme in der Rekursion
- $\frac{1}{b}$ = Teil des Originalproblems, welches wiederum durch alle Unterprobleme repräsentiert wird
- $f(n)$ = Aufwand, welcher durch die Rekursion der Teilprobleme und Kombination der Teillösungen auftritt.

Wozu?

Laufzeitabschätzung von rekursiv definierten Funktionen

Grundaufbau

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Erläuterung

- a = Anzahl der Unterprobleme in der Rekursion
- $\frac{1}{b}$ = Teil des Originalproblems, welches wiederum durch alle Unterprobleme repräsentiert wird
- $f(n)$ = Aufwand, welcher durch die Rekursion der Teilprobleme und Kombination der Teillösungen auftritt.

Wozu?

Laufzeitabschätzung von rekursiv definierten Funktionen

Grundaufbau

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Erläuterung

- a = Anzahl der Unterprobleme in der Rekursion
- $\frac{1}{b}$ = Teil des Originalproblems, welches wiederum durch alle Unterprobleme repräsentiert wird
- $f(n)$ = Aufwand, welcher durch die Rekursion der Teilprobleme und Kombination der Teillösungen auftritt.

Wozu?

Laufzeitabschätzung von rekursiv definierten Funktionen

Grundaufbau

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Erläuterung

- a = Anzahl der Unterprobleme in der Rekursion
- $\frac{1}{b}$ = Teil des Originalproblems, welches wiederum durch alle Unterprobleme repräsentiert wird
- $f(n)$ = Aufwand, welcher durch die Rekursion der Teilprobleme und Kombination der Teillösungen auftritt.

Wozu?

Laufzeitabschätzung von rekursiv definierten Funktionen

Grundaufbau

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Erläuterung

- a = Anzahl der Unterprobleme in der Rekursion
- $\frac{1}{b}$ = Teil des Originalproblems, welches wiederum durch alle Unterprobleme repräsentiert wird
- $f(n)$ = Aufwand, welcher durch die Rekursion der Teilprobleme und Kombination der Teillösungen auftritt.

Wie funktioniert's?

Man unterscheidet zwischen 3 Fällen:

- 1 Wenn $f(n) \in \mathcal{O}(n^{\log_b a - \epsilon})$ mit $\epsilon > 0$, $\Rightarrow T(n) \in \Theta(n^{\log_b a})$
- 2 Wenn $f(n) \in \Theta(n^{\log_b a})$, $\Rightarrow T(n) \in \Theta(n^{\log_b a} \log n)$
- 3 Wenn $f(n) \in \Omega(n^{\log_b a + \epsilon})$ mit $\epsilon > 0$,
und wenn es ein c gibt, mit $0 < c < 1$,
sodass für alle hinreichend großen n gilt: $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$,
 $\Rightarrow T(n) \in \Theta(f(n))$

Wie funktioniert's?

Man unterscheidet zwischen 3 Fällen:

- 1 Wenn $f(n) \in \mathcal{O}(n^{\log_b a - \epsilon})$ mit $\epsilon > 0$, $\Rightarrow T(n) \in \Theta(n^{\log_b a})$
- 2 Wenn $f(n) \in \Theta(n^{\log_b a})$, $\Rightarrow T(n) \in \Theta(n^{\log_b a} \log n)$
- 3 Wenn $f(n) \in \Omega(n^{\log_b a + \epsilon})$ mit $\epsilon > 0$,
und wenn es ein c gibt, mit $0 < c < 1$,
sodass für alle hinreichend großen n gilt: $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$,
 $\Rightarrow T(n) \in \Theta(f(n))$

Wie funktioniert's?

Man unterscheidet zwischen 3 Fällen:

- 1 Wenn $f(n) \in \mathcal{O}(n^{\log_b a - \epsilon})$ mit $\epsilon > 0$, $\Rightarrow T(n) \in \Theta(n^{\log_b a})$
- 2 Wenn $f(n) \in \Theta(n^{\log_b a})$, $\Rightarrow T(n) \in \Theta(n^{\log_b a} \log n)$
- 3 Wenn $f(n) \in \Omega(n^{\log_b a + \epsilon})$ mit $\epsilon > 0$,
und wenn es ein c gibt, mit $0 < c < 1$,
sodass für alle hinreichend großen n gilt: $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$,
 $\Rightarrow T(n) \in \Theta(f(n))$

Wie funktioniert's?

Man unterscheidet zwischen 3 Fällen:

- 1 Wenn $f(n) \in \mathcal{O}(n^{\log_b a - \epsilon})$ mit $\epsilon > 0$, $\Rightarrow T(n) \in \Theta(n^{\log_b a})$
- 2 Wenn $f(n) \in \Theta(n^{\log_b a})$, $\Rightarrow T(n) \in \Theta(n^{\log_b a} \log n)$
- 3 Wenn $f(n) \in \Omega(n^{\log_b a + \epsilon})$ mit $\epsilon > 0$,
und wenn es ein c gibt, mit $0 < c < 1$,
sodass für alle hinreichend großen n gilt: $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$,
 $\Rightarrow T(n) \in \Theta(f(n))$

Beispiele

$$49 \cdot T\left(\frac{n}{7}\right) + 3n + 5$$

$$49 \cdot T\left(\frac{n}{7}\right) + 3n^3 + 5$$

- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten**
- 4 Moore-Automaten
- 5 Endliche Akzeptoren
- 6 Fragen

Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

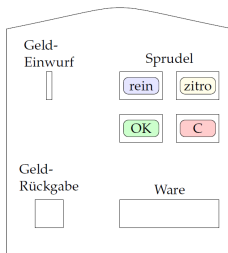
Definition: Mealy-Automat

Der Mealy-Automat $A = (Z, z_0, X, f, Y, g)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $g : Z \times X \rightarrow Y^*$.

Interaktive Aufgabe

Einen Getränke**automaten** modellieren:
saurer Sprudel = *rein* (Eingabe/Ausgabe: R)
süßer Sprudel = *zitro* (Eingabe/Ausgabe: Z)
Abbrechen = C (Eingabe: C)
Bestätigen = OK (Eingabe: O)
Ein Getränk kostet 1 Euro (Eingabe: 1)



f^* und f^{**}

f^* : $f^*(z, w)$ kann im Gegensatz zu f ein ganzes Wort w , als zweites Funktionsargument nehmen, und gibt somit an, in welchem Zustand sich man sich befindet, nachdem man das Wort w abgearbeitet hat.

f^{**} : $f^{**}(z, w)$ gibt die Durchlaufenen Zustände bei der Eingabe w an.

g^* und g^{**}

Simultan zu f^* und f^{**} geben die Funktionen $g^*(z, w)$ und $g^{**}(z, w)$ die Ausgabe nach dem eingegebenen Wort w an.

f^* und f^{**}

f^* : $f^*(z, w)$ kann im Gegensatz zu f ein ganzes Wort w , als zweites Funktionsargument nehmen, und gibt somit an, in welchem Zustand sich man sich befindet, nachdem man das Wort w abgearbeitet hat.

f^{**} : $f^{**}(z, w)$ gibt die Durchlaufenen Zustände bei der Eingabe w an.

g^* und g^{**}

Simultan zu f^* und f^{**} geben die Funktionen $g^*(z, w)$ und $g^{**}(z, w)$ die Ausgabe nach dem eingegebenen Wort w an.

- Berechnen Sie $f^{**}((0, -), RZR11C)$ für den an der Tafel stehenden Automaten
- Berechnen Sie $g^{**}((0, -), RZR11O)$ für den an der Tafel stehenden Automaten

- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten
- 4 Moore-Automaten**
- 5 Endliche Akzeptoren
- 6 Fragen

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

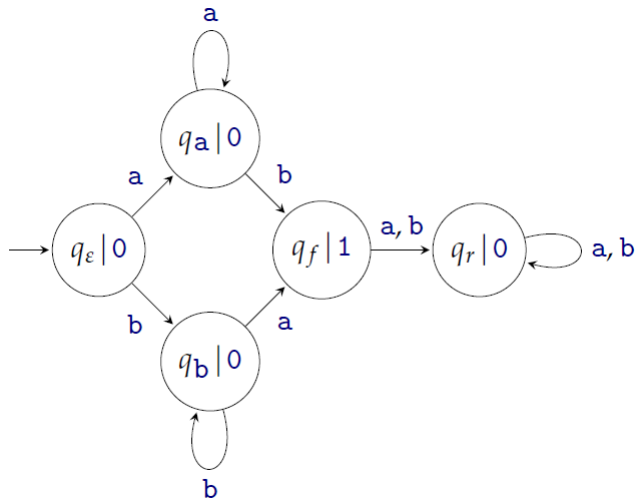
- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Definition: Moore-Automat

Der Moore-Automat $A = (Z, z_0, X, f, Y, h)$ besteht aus

- der endlichen Zustandsmenge Z ,
- dem Startzustand z_0 ,
- dem Eingabealphabet X ,
- der Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$,
- einem Ausgabealphabet Y und
- der Ausgabefunktion $h : Z \rightarrow Y^*$.

Beispiel



- Wandeln Sie den Automaten von Eben in einen Mealy-Automaten um
- Modellieren Sie einen Automaten, welcher bei Wörter mit gerader Länge, welche mindestens einmal aa und einmal bb enthalten eine 1 ausgibt, sonst 0.

- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten
- 4 Moore-Automaten
- 5 Endliche Akzeptoren**
- 6 Fragen

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

Definition

Ein endlicher Akzeptor $A = (Z, z_0, X, f, F)$ ist festgelegt durch:

- eine endliche Zustandsmenge Z ,
- einen Anfangszustand $z_0 \in Z$,
- ein Eingabealphabet X ,
- eine Zustandsübergangsfunktion $f : Z \times X \rightarrow Z$ und
- eine Menge $F \subseteq Z$ akzeptierender Zustände.

Akzeptierende Zustände werden in der Regel als doppelte Kringel dargestellt.

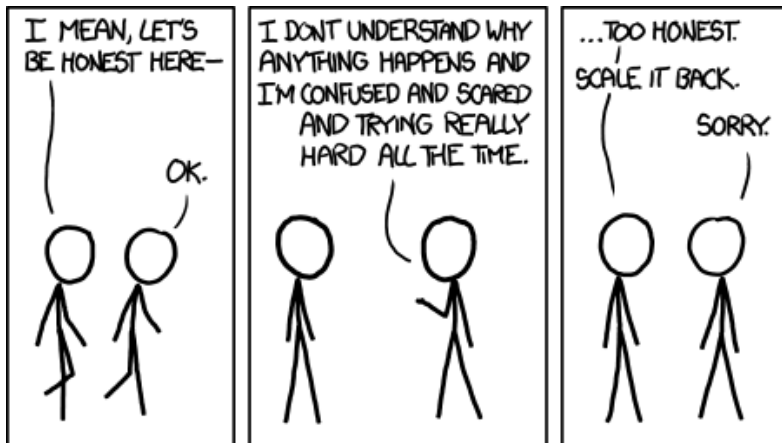
Beispiel

Ein endlicher Akzeptor, welcher alle Wörter aus $\{a, b\}^*$ akzeptiert, welche mit a anfangen und aufhören oder mit b anfangen und aufhören.

- Entwerfen Sie einen endlichen Akzeptor, mit $X = \{a, b\}$, der alle Wörter akzeptiert, bei denen die Anzahl der a durch 5 teilbar ist.
- Entwerfen Sie einen endlichen Akzeptor, mit $X = \{a, b\}$, der alle Wörter akzeptiert, in denen nirgends hintereinander zwei b vorkommen.

- 1 Wiederholung
- 2 Master Theorem
- 3 Mealy-Automaten
- 4 Moore-Automaten
- 5 Endliche Akzeptoren
- 6 Fragen**

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



source : <http://imgs.xkcd.com/comics/honest.png>