Algorithmen und Datenstrukturen

Theoretische Grundlagen der Informatik

Prof. Dr. rer. nat. Jörg Striegnitz

Fachhochschule Aachen
striegnitz@fh-aachen.de

19. April 2024

Algorithmen und Datenstrukturen

Deterministische endliche Automaten

Prof. Dr. rer. nat. Jörg Striegnitz

Fachhochschule Aachen
striegnitz@fh-aachen.de

19. April 2024

Standortbestimmung

Standort

- **Wir haben gelernt**
  - wie man auch komplexere Daten durch Wörter (Sprachen) beschreiben kann
  - dass Σ_{Bool} als Alphabet ausreichend ist (Homomorphismen)
  - wie man die Mächtigkeit (Größe) einer Sprache bestimmt
  - wie man mit Sprachen algorithmische Probleme beschreibt
  - dass sich auch komplexe Probleme auf **Entscheidungsproblem** zurückführen lassen

Neue Fragestellung

Wie kann man Algorithmen zur Lösung von **Entscheidungsproblemen** formal (systematisch) beschreiben?

Übersicht

- **Ziel**: Minimales und vollständiges Modell für Computer / Algorithmen
  - **minimal**: Konzentration auf das für eine Berechnung Wesentliche
    daher: Fokus auf Entscheidungsprobleme
  - **vollständig**: **jeder** Algorithmus kann simuliert werden

- Drei Modelle:
  - Endliche Automaten
    nur Entscheidungsprobleme
  - Kellerautomaten (hier nicht behandelt)
  - Turingmaschinen

Deterministische endliche Automaten (DEAs) - Übersicht

Komponenten:

- Eingabeband (endlich)

  Inhalt: Symbole eines **Eingabealphabets** Σ
- Zustände (endlich viele)

  Sozusagen der **Speicher** eines DEA. Zwei Arten:
  - akzeptierend und
  - verwerfend

  ein Startzustand
- Kontrolle

  Steuert den Ablauf

Arbeitsweise:

- schrittweise lesen der Eingabesymbole (von links nach rechts)
- in jedem Schritt: Wechsel des Zustandes gemäß einer totalen Übergangsfunktion

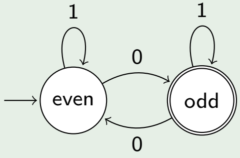
  Merken einer Eigenschaft der bisher gelesenen Eingabe
- Abarbeitung endet mit letztem Symbol

  Zustandsart entscheidet über Ergebnis

Beispiel: DEA

DEA für $L = \{w \in \Sigma^*_{Bool} \mid |w|_0 \text{ ist ungerade}\}$

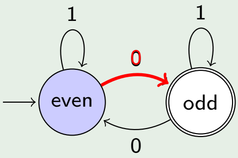
Darstellung als gerichteter Graph: Zustände entsprechen Knoten, Übergangsfunktion den gewichteten Kanten; Startzustand durch eingehenden Pfeil; akzeptierende Zustände mit doppeltem Kreis

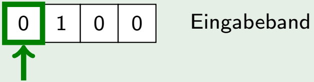


Beachte: Für jedes Symbol aus dem Eingabealphabet eine Kante!

Beispiel: Berechnung eines DEA 1

Berechnung eines DEA - Ausgangskonfiguration



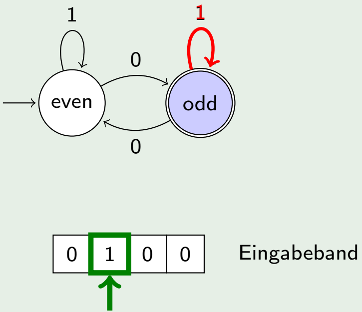


Der Automat befindet sich im (Start-)Zustand *even* und liest die **0**

⇒ Wechsel in Zustand *odd*, Lesekopf eine Stelle nach rechts ...

Beispiel: Berechnung eines DEA 2

Berechnung eines DEA - Ausgangskonfiguration

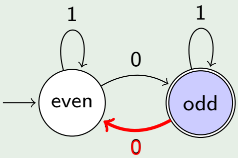


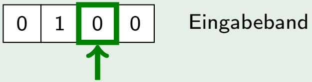
Der Automat befindet sich im Zustand *odd* und liest die **1**

$\Rightarrow$ Verbleib in Zustand *odd*, Lesekopf eine Stelle nach rechts ...

Beispiel: Berechnung eines DEA 3

Berechnung eines DEA - Ausgangskonfiguration



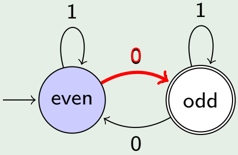


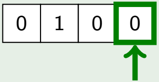
Der Automat befindet sich im Zustand *odd* und liest die **0**

⇒ Wechsel in Zustand *even*, Lesekopf eine Stelle nach rechts ...

Beispiel: Berechnung eines DEA 4

Berechnung eines DEA - Ausgangskonfiguration





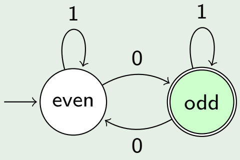
Eingabeband

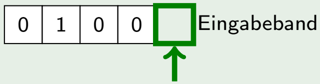
Der Automat befindet sich im Zustand *even* und liest die **0**

⇒ Wechsel in Zustand *odd*, Lesekopf eine Stelle nach rechts ...

Beispiel: Berechnung eines DEA 5

Berechnung eines DEA - Ausgangskonfiguration





Der Automat befindet sich im **akzeptierenden** Zustand *odd*
$\Rightarrow$ Keine weitere Eingabe: der Automat **akzeptiert das Wort** 0100

! Wie schon bei der RAM, wollen wir die Arbeitsweise eines DEA präzise beschreiben.

Aufbau eines DEA

Definition 1.37 (DEA)

Ein deterministischer endlicher Automat (DEA) M ist ein Quintupel $(Q, \Sigma, \delta, q_0, F)$ aus

- einer endlichen Menge von **Zuständen** Q,
- einem **Eingabealphabet** Σ,
- einer **Transitionsfunktion** $\delta : Q \times \Sigma \to Q$
- einem **Startzustand** $q_0 \in Q$ und
- einer Menge von **akzeptierenden Zuständen** $F \subseteq Q$.

Notation: Falls $\delta(p, a) = q$ schreiben wir $p \xrightarrow{a}_M q$ (für $p, q \in Q$ und $a \in \Sigma$), oder, falls M aus dem Kontext heraus bekannt, $p \xrightarrow{a} q$.

Beachte:

- Die Transitionsfunktion ist total! Für jeden Zustand muss für jedes Symbol festgelegt sein, welches der neue Zustand ist
- DEA ohne akzeptierende Zustände ist zulässig

Konfiguration eines DEA

Definition 1.38 (Konfiguration eines DEA)

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat. Eine **Konfiguration** von M ist ein Paar aus $Q \times \Sigma^*$.
(q_0, w) heißt **Startkonfiguration**, (q, ε) **Endkonfiguration** (für ein beliebiges $q \in Q$, $w \in \Sigma^*$).
Gilt für eine **Endkonfiguration** (q, ε)

- $q \in F$, sprechen wir von einer **akzeptierenden Endkonfiguration**, falls jedoch
- $q \in Q - F$, von einer **verwerfenden Endkonfiguration**.

Konfiguration (q, aw) bedeutet:

- Automat befindet sich im Zustand q
- Lesekopf über Symbol a
- noch zu lesende Eingabe ist aw

Notation: Sofern eindeutig schreiben wir qaw anstelle von (q, aw)

Konfigurationsübergang (Schritt)

Definition 1.39 (Schrittrelation)

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Ein **Konfigurationsübergang** (ein Schritt) von M ist eine Relation $\underset{M}{\vdash}: (Q \times \Sigma^*) \times (Q \times \Sigma^*)$, die definiert ist durch:

$$(p, aw) \underset{M}{\vdash} (q, w) \;\Leftrightarrow\; p \overset{a}{\rightarrow}_M q$$

wobei $a \in \Sigma$, $w \in \Sigma^*$ und $\{p, q\} \subseteq Q$. $\underset{M}{\vdash}$ nennen wir **Schrittrelation**.

Berechnung eines DEA

Definition 1.40 (Berechnung eines DEA)

$M = (Q, \Sigma, \delta, q_0, F)$ ein DEA und $w \in \Sigma^*$. Eine **Berechnung** C von M ist eine endliche Folge von Konfigurationen

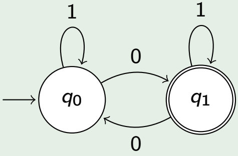
$$C = C_0, \cdots, C_n \;\; \text{mit} \;\; C_i \vdash C_{i+1} \;\; (1 \leq i \leq n-1)$$

C ist die Berechnung von M für eine Eingabe w, falls $C_0 = (q_0, w)$ eine Startkonfiguration und $C_n = (q, \varepsilon)$ eine Endkonfiguration ist. Ist C_n eine akzeptierende Endkonfiguration, so heißt C **akzeptierenden Berechnung** von M auf w. Ist C_n eine verwerfende Konfiguration, so heißt C eine **verwerfende Berechnung** von M auf w (wir sagen, dass M das Wort w verwirft).

Beispiel: Berechnung eines DEA

Berechnung

Gegeben sei der folgende DEA



Akzeptierende Berechnung:

$$q_0 0100 \vdash q_1 100 \vdash q_1 00 \vdash q_0 0 \vdash q_1$$

Verwerfende Berechnung:

$$q_0 0110 \vdash q_1 110 \vdash q_1 10 \vdash q_1 0 \vdash q_0$$

Abschluss der Schrittrelation

Definition 1.41 (Abschluss von Schrittrelation / Transitionsfunktion)

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Wir definieren $\overset{*}{\vdash}: (Q \times \Sigma^*) \times (Q \times \Sigma^*)$ als reflexiven und transitiven Abschluss der Schrittrelation $\vdash$:

- $(q, w) \overset{*}{\vdash} (q, w)$ für alle $q \in Q$ und $w \in \Sigma^*$;
- $(p, uv) \overset{*}{\vdash} (q, v)$, falls $\exists a_1, \cdots, a_n \in \Sigma$ mit $u = a_1 \cdots a_n$ und $\exists q_1, \cdots, q_{n-1} \in Q$, so dass $(p, a_1 a_2 \cdots a_n v) \vdash (q_1, a_2 \cdots a_n v) \vdash \cdots \vdash (q_{n-1}, a_n v) \vdash (q, v)$ (mit $u, v \in \Sigma^*$, $p, q \in Q$).

Die Fortsetzung $\hat{\delta} : Q \times \Sigma^* \to Q$ der Transitionsfunktion δ auf Wörter definieren wir induktiv durch:

- $\hat{\delta}(q, \varepsilon) = q$
- $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$

für $w \in \Sigma^*$, $a \in \Sigma$ und $q \in Q$.

- **Notation**: Falls $\hat{\delta}(p, w) = q$, so schreiben wir auch: $p \overset{w}{\rightarrow}_M q$ und sagen: „Es existiert ein **Pfad** von p nach q in M mit Beschriftung w".

Von DEA erkannte Sprache

Definition 1.42 (Von DEA erkannte Sprache)

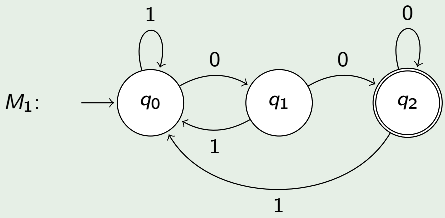
Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Die von M akzeptierte Sprache $L(M)$ ergibt sich aus

$$
\begin{aligned}
L(M) &= \left\{ w \in \Sigma^* \mid (q_0, w) \overset{*}{\vdash} (q_f, \varepsilon) \text{ mit } q_f \in F \right\} \\
&= \left\{ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F \right\} \\
&= \left\{ w \in \Sigma^* \mid q_0 \overset{w}{\to} q_f \text{ mit } q_f \in F \right\}
\end{aligned}
$$

Zwei DEAs M_1 und M_2 heißen äquivalent, falls sie dieselbe Sprache erkennen; also: $L(M_1) = L(M_2)$.

DEAs: Weitere Beispiele - 1

Beispiel 1.32 ($L_1 = \{w \in \Sigma^*_{Bool} \mid w \text{ endet auf } 00\}$)



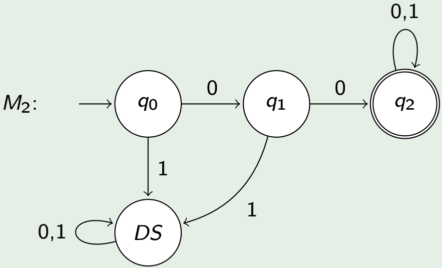
Da $\varepsilon \notin L_1$, ist Startzustand q_0 **nicht akzeptierend**.

Bedeutung der Zustände:

- q_0: bisher gelesene Eingabe ist leeres Wort oder endet auf 1
- q_1: bisher gelesene Eingabe ist 0 oder endet auf 10
- q_2: bisher gelesene Eingabe endet auf 00

DEAs: Weitere Beispiele - 2

Beispiel 1.34 ($L_2 = \{w \in \Sigma_{Bool}^* \mid w \text{ beginnt mit } 00\}$)



Bedeutung der Zustände:

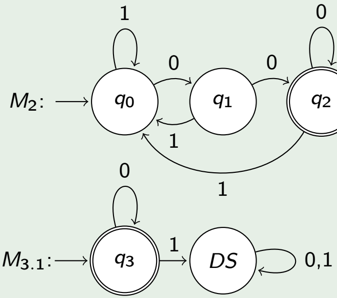
- q_0: bisher gelesene Eingabe ist leeres Wort
- q_1: bisher gelesene Eingabe ist 0
- q_2: bisher gelesene Eingabe beginnt mit 00
- DS: bisher gelesene Eingabe beginnt mit 1 oder 01

Beispiel 1.35 ($L_3 = \{w \in \Sigma^*_{Bool} \mid w \text{ ist teilbar durch } 4\}$)

Vorüberlegung: Binärstring durch 4 teilbar gdw.

- die letzten beiden Ziffern sind 0 (M_2) oder
- Binärstring ist beliebig lange Folge von 0en ($M_{3.1}$):

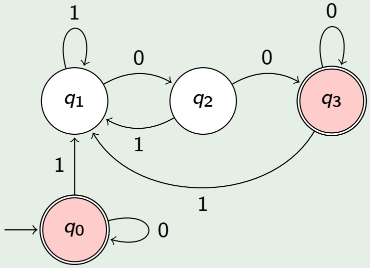
$$L_3 = L_2 \cup \{w \in \Sigma^*_{Bool} \mid w = 0^n,\ n \in \mathbb{N}_0\}$$



Kombiniere beide Automaten zu einem ...

Beispiel 1.35 ($L_3 = \{w \in \Sigma^*_{Bool} \mid w \text{ ist teilbar durch } 4\}$)

$M_{3.2}$:



Beachte: q_0 und q_3 sind in folgendem Sinne äquivalent

$$\forall w \in \Sigma^* : \hat{\delta}(q_0, w) \in F \Leftrightarrow \hat{\delta}(q_3, w) \in F$$

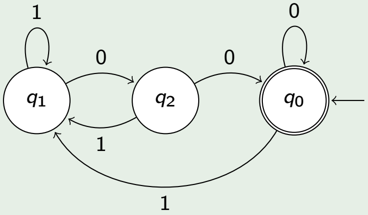
Anschaulich: Betrachte "Wegbeschreibung" w - dann ist es egal, ob man von q_0 oder q_3 losläuft, die Antwort wird dieselbe sein.

DEAs: Weitere Beispiele - 3.3

Beispiel 1.35 ($L_3 = \{w \in \Sigma_{Bool}^* \mid w \text{ ist teilbar durch } 4\}$)

Zusammenfassen der Zustände führt zu:

M_3:



Anmerkung:

- Erstellen von Automaten ist ein konstruktiver Prozess
- **funktionale Dekomposition** (Zerlegung in evtl. kleinere Teilprobleme) als Design-Paradigma ... werden wir noch vertiefen

Beispiel 1.37 ($L_4 = \{w \in \Sigma_{Bool}^* \mid w \text{ ist teilbar durch } 3\}$)

Problem: kein Muster für durch drei teilbare Zahlen

Lösung:

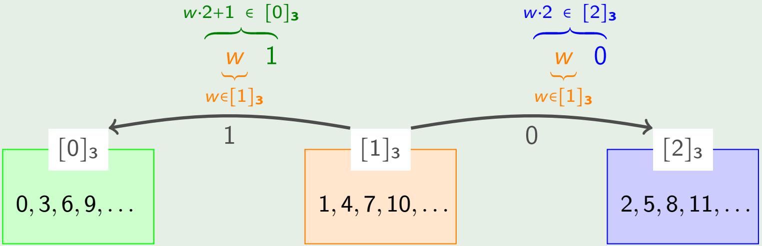
- Konvention: Initial ist Rest 0

  ε wird wie 0 behandelt

- Merke “Divisionsrest” in Zustand (Zustand $\equiv$ Kongruenzklasse)

  M in Zustand $[q]_3$: **bisher gelesene** Eingabe in Kongruenzklasse $[q]_3$

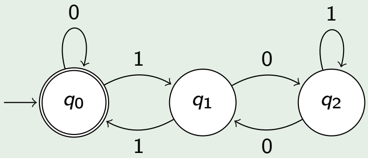
- Angenommen Automat in Zustand $[q]_3$, wohin bei
  - Lesen der 0 bzw. bei
  - Lesen der 1?



DEAs: Weitere Beispiele 4.2

Beispiel 1.37 ($L_4 = \{w \in \Sigma^*_{Bool} \mid w \text{ ist teilbar durch } 3\}$)

M_4:



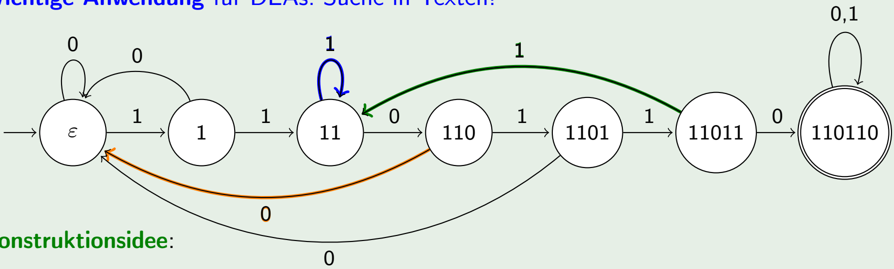
M_4 zerteilt Σ^*_{Bool} in folgende Äquivalenzklassen:

$$
\begin{aligned}
[q_0] &:= \{w \in \Sigma^*_{Bool} \mid q_0 \overset{w}{\to} q_0\} \,\mathbf{=}\, \{w \mid w \bmod 3 = 0\} \\
[q_1] &:= \{w \in \Sigma^*_{Bool} \mid q_0 \overset{w}{\to} q_1\} \,\mathbf{=}\, \{w \mid w \bmod 3 = 1\} \\
[q_2] &:= \{w \in \Sigma^*_{Bool} \mid q_0 \overset{w}{\to} q_2\} \,\mathbf{=}\, \{w \mid w \bmod 3 = 2\}
\end{aligned}
$$

Diese können z.B. für **Korrektheitsbeweis** verwendet werden.

Beispiel 1.38 ($L_5 = \{w \in \Sigma^*_{Bool} \mid w \text{ enthält } 110110\}$)

Wichtige Anwendung für DEAs: Suche in Texten!



Konstruktionsidee:

- Zustände entsprechen Präfixen des Suchmusters p hier: 110110
- Bedeutung: Automat in Zustand q: bisher gelesene Eingabe endet auf q
  Genauer: q ist längstes Suffix der bisher gelesenen Eingabe, welches Präfix von p ist
  Ausnahme: akzeptierender Zustand
- Transitions-Beispiele:
  - $11 \xrightarrow{1} 11$, da 11 längstes Suffix von 111 das Präfix von 110110 ist
  - $11011 \xrightarrow{1} 11$, da 11 längstes Suffix von 110111 das Präfix von 110110 ist
  - $110 \xrightarrow{0} \varepsilon$, da ε längstes Suffix von 1100 das Präfix von 110110 ist

Praktischer Einsatz von DEAs

Beispiel 1.39 ($L_5 = \{w \in \Sigma_{Bool}^* \mid w \text{ enthält } 110110\}$)

Automaten zur Beschreibung von **Prozessen**

- **Hier**: Suche alle Vorkommen von 110110
  mit Überdeckung: 110110110 enthält 110110 zwei mal
- roter Zustand löst Aktion aus (Muster gefunden)
- **Beachte**: Kein "Verharren" in 110110 mehr!