LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Formale Sprachen und Komplexität
Theoretische Informatik für Medieninformatiker
Sommersemester 2022

DFAs akzeptieren reguläre Sprachen, Nichtdeterministische Endliche Automaten

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 10. Mai 2022

Wiederholung: Deterministische endliche Automaten

Definition (Deterministischer Endlicher Automat, DFA)

Ein deterministischer endlicher Automat (deterministic finite automaton, DFA) ist ein 5-Tupel $M = (Z, \Sigma, \delta, z_0, E)$ wobei

- Z ist eine endliche Menge von Zuständen,
- Σ ist das (endliche) Eingabealphabet mit $(Z \cap \Sigma) = \emptyset$,
- $\delta : Z \times \Sigma \to Z$ ist die Zustandsüberführungsfunktion (oder nur Überführungsfunktion),
- $z_0 \in Z$ ist der Startzustand und
- $E \subseteq Z$ ist die Menge der Endzustände (oder auch akzeptierende Zustände).

DFAs akzeptieren reguläre Sprachen

Theorem 4.2.1

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis: Konstruiere für DFA M eine reguläre Grammatik G, sodass $L(M) = L(G)$:
Sei $G = (V, \Sigma, P, S)$ die reguläre Grammatik mit $V = Z$, $S = z_0$ und

$$P = \{z_i \to a z_j \mid \delta(z_i, a) = z_j\} \cup \{z_i \to a \mid \delta(z_i, a) = z_j \wedge z_j \in E\} \cup \{z_0 \to \varepsilon \mid \text{falls } z_0 \in E\}$$

Leeres Wort: Offensichtlich gilt $\varepsilon \in L(M) \iff \varepsilon \in L(G)$.
Worte w mit $|w| = m \geq 1$:

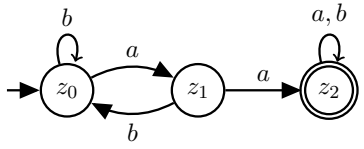
$w = a_1 \cdots a_m \in L(M)$
g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ und $z_m \in E$.
g.d.w. $z_0 \Rightarrow_G a_1 z_1$, für $1 \leq i < m$: $a_1 \cdots a_{i-1} z_{i-1} \Rightarrow_G a_1 \cdots a_i z_i$ und
$a_1 \cdots a_{m-1} z_{m-1} \Rightarrow_G a_1 \cdots a_m$, d.h. $z_0 \Rightarrow_G^* a_1 \cdots a_m$
g.d.w. $w = a_1 \cdots a_m \in L(G)$

Daher gilt $L(M) = L(G)$ und somit ist $L(M)$ regulär. □

Beispiel: Konstruktion Typ 3-Grammatik aus DFA

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{ll} \delta(z_0, a) = z_1 & \delta(z_1, b) = z_0 \\ \delta(z_0, b) = z_0 & \delta(z_2, a) = z_2 \\ \delta(z_1, a) = z_2 & \delta(z_2, b) = z_2 \end{array}$$



Die erzeugte reguläre Grammatik dazu ist:

$$\begin{array}{l} G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0) \text{ mit} \\ P = \{z_0 \to a z_1 \mid b z_0, \\ \qquad z_1 \to a z_2 \mid a \mid b z_0, \\ \qquad z_2 \to a z_2 \mid a \mid b z_2 \mid b\} \end{array} .$$

Wird jede reguläre Sprache durch einen DFA akzeptiert?

- Der vorherige Beweis konstruiert:
  „für jeden DFA gibt es eine äquivalente reguläre Grammatik“
- Für die andere Richtung wäre notwendig
  „für jede reguläre Grammatik gibt es einen äquivalenten DFA“

Problem:

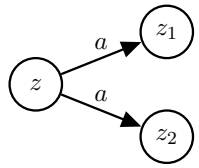
- Produktionen: $A \to aA_1$ und $A \to aA_2$ können in Grammatiken vorkommen
- Konstruktion des determinstischen Automaten zunächst unklar

Daher: Beweis, dass DFAs alle regulären Sprachen akzeptieren, erfolgt auf Umwegen und verwendet nichtdeterministische endliche Automaten

Nichtdeterministische Endliche Automaten

Ideen:

- Zustandswechsel nicht eindeutig, sondern nichtdeterministisch in einen von mehreren möglichen
- D.h. der Automat darf sozusagen „raten“, welchen Nachfolgezustand er wählt
- Im Zustandsgraph erlaubt:



- Technisch:
  - DFA $\delta : Z \times \Sigma \to Z$ und ein Startzustand
  - NFA $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ und Menge von Startzuständen

Definition NFA

Definition

Ein **nichtdeterministischer endlicher Automat**
(nondeterministic finite automaton, NFA) ist ein 5-Tupel $(Z, \Sigma, \delta, S, E)$ wobei

- Z ist eine endliche Menge von Zuständen,
- Σ ist das (endliche) Eingabealphabet mit $(Z \cap \Sigma) = \emptyset$,
- $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ ist die Zustandsüberführungsfunktion,
- $S \subseteq Z$ ist die Menge der Startzustände und
- $E \subseteq Z$ ist die Menge der Endzustände.

Akzeptanz beim NFA

„Ein Wort w wird vom NFA akzeptiert, wenn es einen Pfad von einem Startzustand zum Endzustand entlang w **gibt** *“*

Definition (Akzeptierte Sprache eines NFA)

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA.
Wir definieren $\widehat{\delta} : (\mathcal{P}(Z) \times \Sigma^*) \to \mathcal{P}(Z)$ induktiv durch:

$$\begin{aligned} \widehat{\delta}(X, \varepsilon) &:= X \text{ für alle } X \subseteq Z \\ \widehat{\delta}(X, aw) &:= \bigcup_{z \in X} \widehat{\delta}(\delta(z, a), w) \text{ für alle } X \subseteq Z \end{aligned}$$

Die von M akzeptierte Sprache ist

$$L(M) = \{w \in \Sigma^* \mid \widehat{\delta}(S, w) \cap E \neq \emptyset\}$$

Beispiel: Leere Menge von Startzuständen

Sei $M = (Z, \Sigma, \delta, \emptyset, E)$ ein NFA.
Dann ist $L(M) = \emptyset$.

Lauf beim NFA

Definition

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA und $w \in \Sigma^*$ mit $|w| = n$.
Eine Folge von Zuständen $q_0, \ldots, q_n$ mit $q_0 \in S$ und $q_i \in \delta(q_{i-1}, w[i])$ bezeichnet man als **Lauf** von M für Wort w.

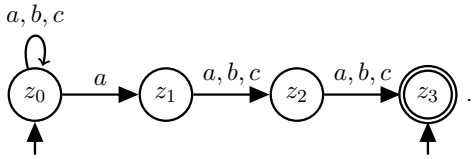
Ein Lauf der mit einem Endzustand endet, nennen wir auch **akzeptierender Lauf**.

Beachte: Während es bei DFAs genau einen Lauf pro Wort gibt, kann es bei NFAs mehrere geben.

Beispiel

Sei $M = (\{z_0, z_1, z_2, z_3\}, \{a, b, c\}, \delta, \{z_0, z_3\}, \{z_3\})$ ein NFA mit

$$\begin{array}{llll} \delta(z_0, a) = \{z_0, z_1\} & \delta(z_1, a) = \{z_2\} & \delta(z_2, a) = \{z_3\} & \delta(z_3, a) = \emptyset \\ \delta(z_0, b) = \{z_0\} & \delta(z_1, b) = \{z_2\} & \delta(z_2, b) = \{z_3\} & \delta(z_3, b) = \emptyset \\ \delta(z_0, c) = \{z_0\} & \delta(z_1, c) = \{z_2\} & \delta(z_2, c) = \{z_3\} & \delta(z_3, c) = \emptyset \end{array}$$

Der Zustandsgraph zu M ist 

.

$$L(M) = \{\varepsilon\} \cup \{uaw \mid u \in \{a, b, c\}^*, w \in \{a, b, c\}^2\}$$