Zentralübung 09.05.2019: DFAs & NFAs

Prof. Dr. David Sabel
LFE Theoretische Informatik

Endliche Automaten

<table>
<thead>
<tr>
<th>Zustandsmenge</th>
<th>DFA</th>
<th>NFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>Alphabet</td>
<td>Σ</td>
<td>Σ</td>
</tr>
<tr>
<td>Zustandsübergang</td>
<td>δ : (Z × Σ) → Z</td>
<td>δ : (Z × Σ) → P(Z)</td>
</tr>
<tr>
<td>Startzust.</td>
<td>z₀ ∈ Z</td>
<td>S ⊆ Z</td>
</tr>
<tr>
<td>Endzustände</td>
<td>E ⊆ Z</td>
<td>E ⊆ Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Übergang f. Worte</th>
<th>DFA</th>
<th>NFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>w ∈ L(M)</td>
<td>δ̂ : (Z × Σ*) → Z</td>
<td>δ̂ : (P(Z) × Σ*) → P(Z)</td>
</tr>
<tr>
<td></td>
<td>δ̂(z₀, w) ∈ E</td>
<td>δ̂(S, w) ∩ E ≠ ∅</td>
</tr>
</tbody>
</table>

Uhrzeiten erkennen

**Aufgabe**

Geben Sie einen NFA an, der genau alle gültigen Uhrzeiten im 24-Stunden-Format der Form Stunde:Minute mit Stunde ∈ {0, ..., 23} und Minute ∈ {00, 01, ..., 59} (zweistellig) erkennt.

Beachte: Σ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :}

Uhrzeiten erkennen

**Aufgabe**

Geben Sie einen DFA an, der genau alle gültigen Uhrzeiten im 24-Stunden-Format der Form Stunde:Minute mit Stunde ∈ {0, ..., 23} und Minute ∈ {00, 01, ..., 59} (zweistellig) erkennt.
NFAs konstruieren – Nichtdeterminismus ausnutzen

Aufgabe (vgl. Hopcroft, Motwani, Ullman)
Geben Sie einen NFA an, der Ziffernfolgen aus \( \Sigma = \{0, \ldots, 9\} \) verarbeitet und Worte genau dann akzeptiert, wenn

a) die letzte Ziffer vorher schon mindestens einmal in der Folge vorkam
b) die letzte Ziffer vorher nicht schon einmal vorkam

Nutzen Sie den Nichtdeterminismus möglichst gut aus.

b) die letzte Ziffer vorher nicht schon einmal vorkam

Reguläre Sprachen und Endliche Automaten

Äquivalenz der Formalismen:
Nochmal: Beweisstruktur

Seien
- \( D \) = Menge der durch DFAs akzeptierten Sprachen
- \( N \) = Menge der durch NFAs akzeptierten Sprachen
- \( R \) = Menge der regulären Sprachen
Wir haben gezeigt
- \( D \subseteq R \)
- \( R \subseteq N \)
- \( N \subseteq D \)
Daraus folgt: \( D = R = N \)

Ein Spiel

Spielfeld:

![Spielfelddiagramm](image1)

Aktionen:
- \( l \) und \( r \): Wechseln um ein Feld nach links bzw. rechts.
- \( l \) bei Feld 1 und \( r \) bei Feld 3 bleiben wirkungslos.

Aufgabe

Geben Sie einen DFA über dem Alphabet \( \Sigma = \{l, r\} \) an, der alle Folgen von Spielzügen akzeptiert, sodass der Spieler auf Feld 1 beginnt und auf Feld 3 das Spiel beendet.

![DFA-Abbildung](image2)

DFA → reguläre Grammatik

Aufgabe

![Abbildung zu Aufgabenstellung](image3)

Geben Sie zum gezeigten DFA eine reguläre Grammatik an, welche die vom DFA akzeptierte Sprache erzeugt.

Erinnerung: DFA → reguläre Grammatik

Für DFA \( M = (Z, \Sigma, \delta, z_0, E) \) konstruiere \( G = (V, \Sigma, P, S) \) mit
\[
V = Z, \quad S = z_0, \quad P = \{ z_i \rightarrow az_j | \delta(z_i, a) = z_j \} \\
\cup \{ z_i \rightarrow a | \delta(z_i, a) = z_j \wedge z_j \in E \} \\
\cup \{ z_0 \rightarrow \varepsilon | \text{falls } z_0 \in E \}
\]
Reguläre Grammatik → NFA

**Aufgabe**

\[ G = (\{S, A, D, E\}, \{l, r\}, P, A) \]
beschreibt Spielzugfolgen, mit
\[ P = \{ S \rightarrow lA \mid rS, A \rightarrow lD \mid lE \mid l \mid rS, D \rightarrow l \mid lD \mid rA \mid lE, E \rightarrow l \mid lD \} \].

Konstruieren Sie einen NFA \( M \) mit \( L(M) = L(G) \).

**Zur Erinnerung: Transformation reg. Grammatik in NFA**

Füre \((V, \Sigma, P, S)\) erzeuge \((Z, \Sigma, \delta, S', E)\) mit
\[
Z = V \cup \{z_E\} \\
S' = \{S\} \\
E = \{z_E\} \cup \{S \mid \text{ falls } S \rightarrow \varepsilon \in P\} \\
\delta(A, a) = \{N \mid A \rightarrow aN \in P\} \cup \{z_E \mid A \rightarrow a \in P\} \\
\delta(z_E, a) = \emptyset
\]

NFA angeben

**Aufgabe**

Geben Sie einen NFA über dem Alphabet \( \Sigma = \{l, r\} \) an, der alle Folgen von Spielzügen akzeptiert, sodass der Spieler auf irgendeinem Feld beginnt und auf Feld 3 das Spiel beendet.

Reguläre Grammatik → NFA (2)

**Aufgabe**

\[ G = (\{S, A, D, E\}, \{l, r\}, P, A) \]
beschreibt Spielzugfolgen, mit
\[ P = \{ S \rightarrow lA \mid rS, A \rightarrow lD \mid lE \mid l \mid rS, D \rightarrow l \mid lD \mid rA \mid lE, E \rightarrow l \mid lD \} \].

Konstruieren Sie einen NFA \( M \) mit \( L(M) = L(G) \).

**Konstruktion des NFA:**

\( M = (\{S, A, D, E, z_E\}, \{l, r\}, \delta, \{S\}, \{z_E\}) \) mit
\[
\delta(S, l) = \{A\} \\
\delta(S, r) = \{S\} \\
\delta(A, l) = \{D, E, z_E\} \\
\delta(A, r) = \{S\} \\
\delta(D, l) = \{D, E, z_E\} \\
\delta(D, r) = \{A\} \\
\delta(E, l) = \{D, z_E\} \\
\delta(E, r) = \emptyset \\
\delta(z_E, l) = \emptyset \\
\delta(z_E, r) = \emptyset
\]

NFA → DFA

**Aufgabe**

Überführen Sie den NFA

in einen DFA durch Verwendung der Potenzmengenkonstruktion.

**Potenzmengenkonstruktion**

Für NFA \( (Z, \Sigma, \delta, S, E) \) konstruiere DFA \( (P(Z), \Sigma, \delta', S, E') \) mit
\[ E' = \{X \in P(Z) \mid E \cap X \neq \emptyset\} \]
und \[ \delta'(X, a) = \bigcup_{z \in X} \delta(z, a) \]
NFA → DFA (2)

NFA:

DFA: