

Zentralübung 09.05.2019: DFAs & NFAs

Prof. Dr. David Sabel

LFE Theoretische Informatik



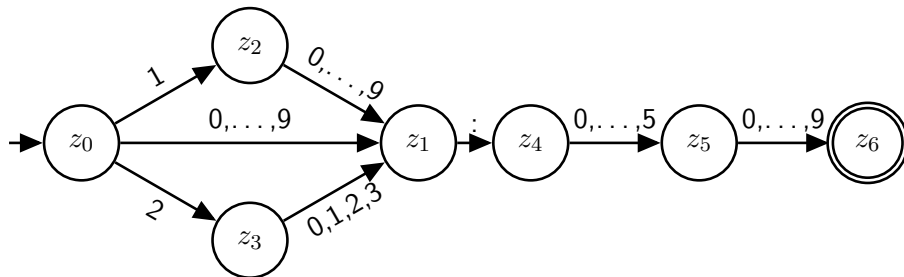
Letzte Änderung der Folien: 14. Mai 2019

Uhrzeiten erkennen

Aufgabe

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

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



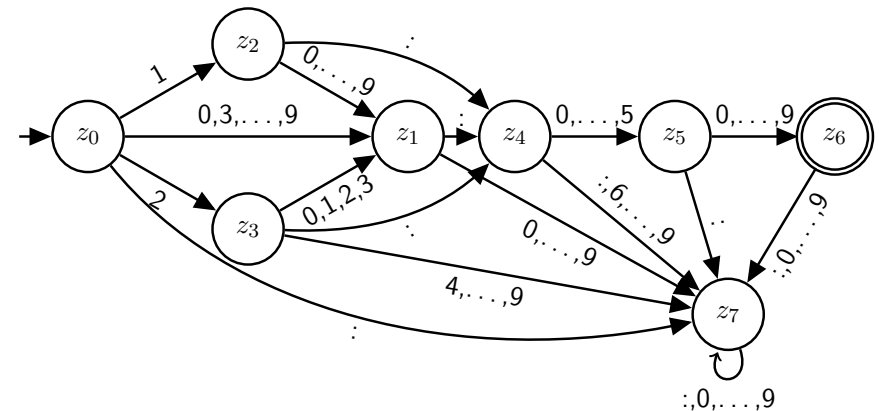
Endliche Automaten

	DFA	NFA
Zustandsmenge	Z	Z
Alphabet	Σ	Σ
Zustandsübergang	$\delta :: (Z \times \Sigma) \rightarrow Z$	$\delta :: (Z \times \Sigma) \rightarrow \mathcal{P}(Z)$
Startzust.	$z_0 \in Z$	$S \subseteq Z$
Endzustände	$E \subseteq Z$	$E \subseteq Z$
Übergang f. Worte $w \in L(M)$	$\hat{\delta} :: (Z \times \Sigma^*) \rightarrow Z$ $\hat{\delta}(z_0, w) \in E$	$\hat{\delta} :: (\mathcal{P}(Z) \times \Sigma^*) \rightarrow \mathcal{P}(Z)$ $\hat{\delta}(S, w) \cap E \neq \emptyset$

Uhrzeiten erkennen

Aufgabe

Geben Sie einen **DFA** an, der genau alle gültigen Uhrzeiten im 24-Stunden-Format der Form **Stunde:Minute** mit $\text{Stunde} \in \{0, \dots, 23\}$ und $\text{Minute} \in \{00, 01, \dots, 59\}$ (zweistellig) erkennt.



NFAs konstruieren – Nichtdeterminismus ausnutzen

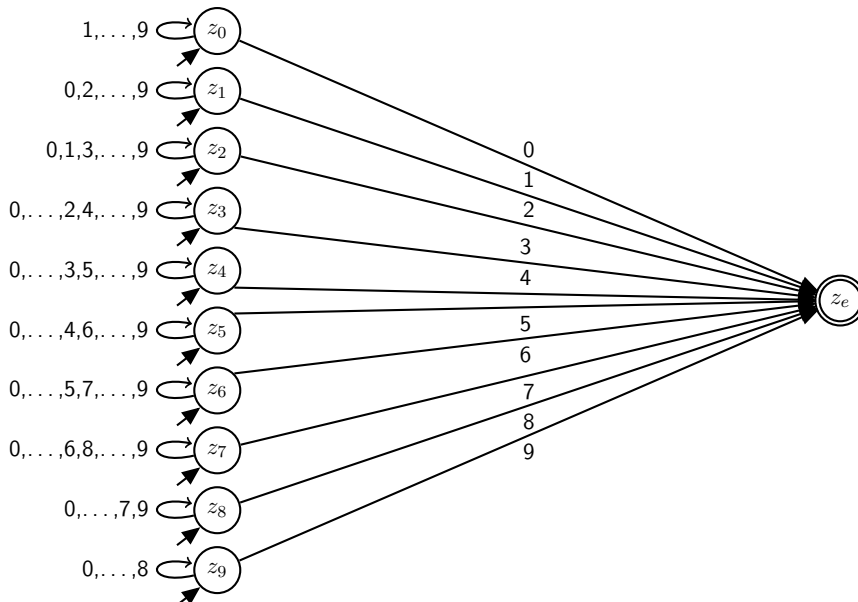
Aufgabe (vgl. Hopcroft, Motwani, Ullman)

Geben Sie einen NFA an, der Ziffernfolgen aus $\Sigma = \{0, \dots, 9\}$ verarbeitet und Worte genau dann akzeptiert, wenn

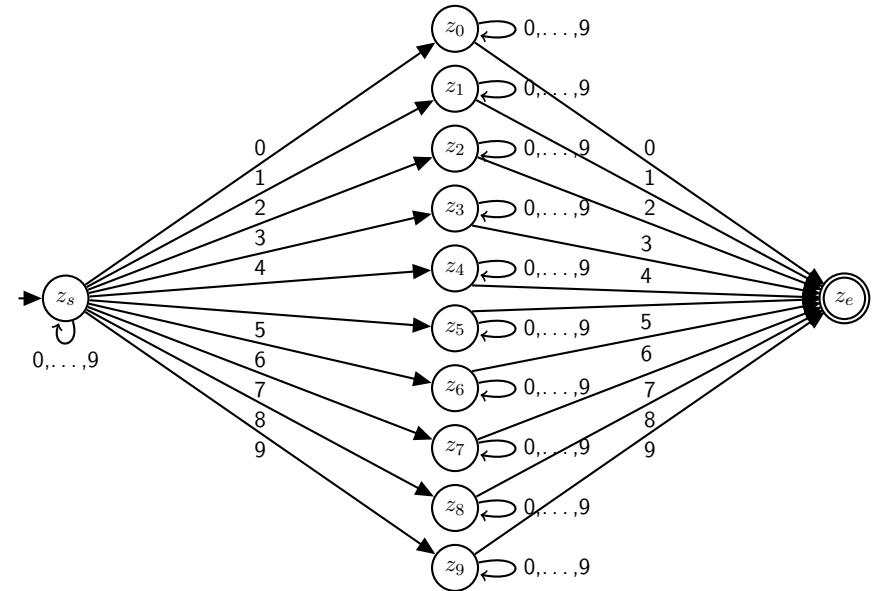
- a) die letzte Ziffer vorher schon mindestens einmal in der Folge vorkam
- b) die letzte Ziffer vorher nicht schon einmal vorkam

Nutzen Sie den Nichtdeterminismus möglichst gut aus.

- b) die letzte Ziffer vorher nicht schon einmal vorkam

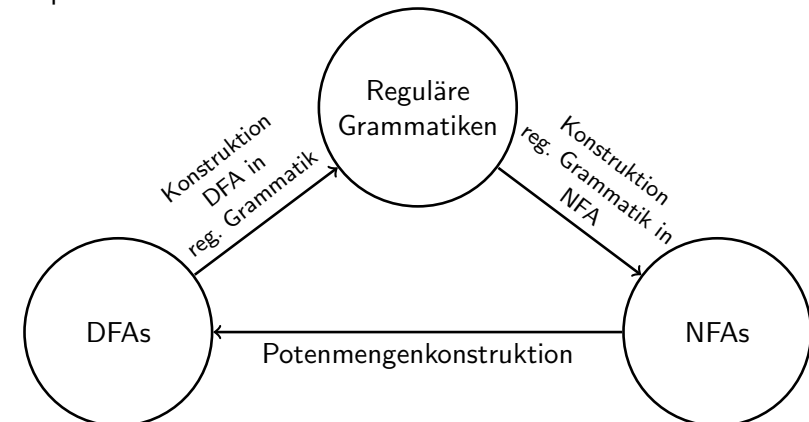


- a) die letzte Ziffer vorher schon mindestens einmal in der Folge vorkam



Reguläre Sprachen und Endliche Automaten

Äquivalenz der Formalismen:



Seien

- \mathcal{D} = Menge der durch DFAs akzeptierten Sprachen
- \mathcal{N} = Menge der durch NFAs akzeptierten Sprachen
- \mathcal{R} = Menge der regulären Sprachen

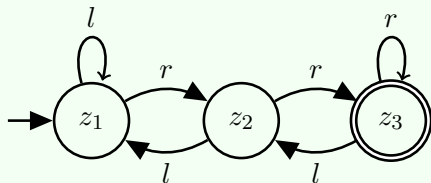
Wir haben gezeigt

- $\mathcal{D} \subseteq \mathcal{R}$
- $\mathcal{R} \subseteq \mathcal{N}$
- $\mathcal{N} \subseteq \mathcal{D}$

Daraus folgt: $\mathcal{D} = \mathcal{R} = \mathcal{N}$

DFA → reg. Grammatik

Aufgabe



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
 $V = Z, S = z_0, P = \{z_i \rightarrow az_j \mid \delta(z_i, a) = z_j\}$
 $\cup \{z_i \rightarrow a \mid \delta(z_i, a) = z_j \wedge z_j \in E\}$
 $\cup \{z_0 \rightarrow \varepsilon \mid \text{falls } z_0 \in E\}$

Ein Spiel

Spielfeld:

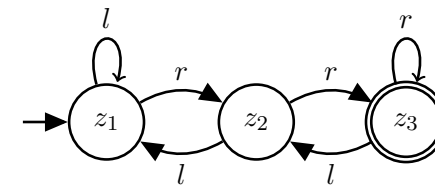


Aktionen:

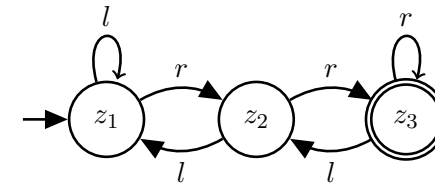
- l und r : Wechseln um ein Feld nach *links* bzw. *rechts*.
- l bei Feld 1 und r bei Feld 3 bleiben wirkungslos.

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 → reguläre Grammatik (2)



Reguläre Grammatik dazu:

$$\begin{aligned}
 G &= (V, \Sigma, P, S) \text{ mit} \\
 V &= \{z_1, z_2, z_3\} \\
 P &= \{z_1 \rightarrow lz_1, z_1 \rightarrow rz_2, \\
 &\quad z_2 \rightarrow lz_1, z_2 \rightarrow rz_3, \\
 &\quad z_3 \rightarrow lz_2, z_3 \rightarrow rz_3, \\
 &\quad z_2 \rightarrow r, z_3 \rightarrow r\} \\
 S &= z_1
 \end{aligned}$$

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

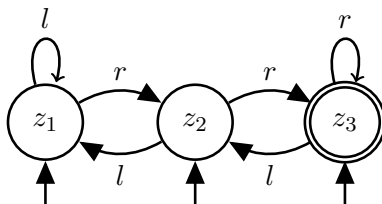
Für (V, Σ, P, S) erzeuge $(Z, \Sigma, \delta, S', E)$ mit

$$\begin{aligned} Z &= V \cup \{z_E\} \\ S' &= \{S\} \\ E &= \{z_E\} \cup \{S \mid \text{falls } S \rightarrow \varepsilon \in P\} \\ \delta(A, a) &= \{N \mid A \rightarrow aN \in P\} \cup \{z_E \mid A \rightarrow a \in P\} \\ \delta(z_E, a) &= \emptyset \end{aligned}$$

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.



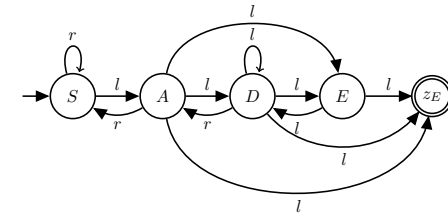
Reguläre Grammatik → NFA (2)

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

$$\begin{aligned} \delta(S, l) &= \{A\} \\ \delta(S, r) &= \{S\} \\ \delta(A, l) &= \{D, E, z_E\} \\ \delta(A, r) &= \{S\} \\ \delta(D, l) &= \{D, E, z_E\} \\ \delta(D, r) &= \{A\} \\ \delta(E, l) &= \{D, z_E\} \\ \delta(E, r) &= \emptyset \\ \delta(z_E, l) &= \emptyset \\ \delta(z_E, r) &= \emptyset \end{aligned}$$

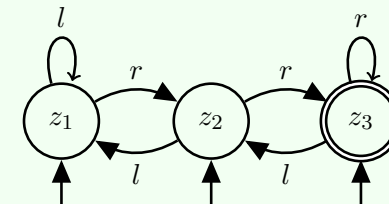


Alle Folgen um von Feld 3
zu Feld 1 zu kommen

NFA → DFA

Aufgabe

Überführen Sie den NFA



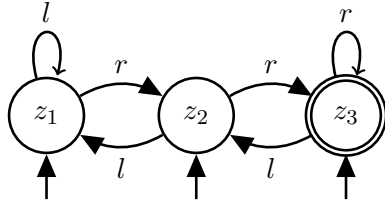
in einen DFA durch Verwendung der Potenzmengenkonstruktion.

Potenzmengenkonstruktion

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

NFA \rightarrow DFA (2)

NFA:



DFA:

