FSK
Zentralübung 2

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik

Stand: 3. Mai 2023
Folien ursprünglich von PD Dr. David Sabel

DFAs

Deterministische endliche Automaten: $M = (Z, \Sigma, \delta, z_0, E)$ mit

	DFA	Beispiel
Zustandsmenge	Z	$\{z_0, z_1, z_2\}$
Alphabet	Σ	$\{a, b\}$
Zustandsübergang	$\delta : (Z \times \Sigma) \to Z$	$\delta(z_0, a) = z_1$
		$\delta(z_0, b) = z_2$
		$\delta(z_1, a) = z_1$
		$\delta(z_1, b) = z_2$
		$\delta(z_2, a) = z_2$
		$\delta(z_2, b) = z_2$
Startzustand	$z_0 \in Z$	z_0
Endzustände	$E \subseteq Z$	$\{z_0, z_1\}$

Erkannte Sprache: $w \in L(M)$ g.d.w. $\widehat{\delta}(z_0, w) \in E$

(wobei $\widehat{\delta}(z, \varepsilon) = z$ und $\widehat{\delta}(z, aw) = \widehat{\delta}(\delta(z, a), w)$)

Darstellungsformen

- als 5-Tupel $(\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_0, z_1\})$ mit
$$\begin{aligned} \delta(z_0, a) &= z_1 & \delta(z_0, b) &= z_2 \\ \delta(z_1, a) &= z_1 & \delta(z_1, b) &= z_2 \\ \delta(z_2, a) &= z_2 & \delta(z_2, b) &= z_2 \end{aligned}$$
- mit Übergangstabelle für δ:

δ	a	b
z_0	z_1	z_2
z_1	z_1	z_2
z_2	z_2	z_2

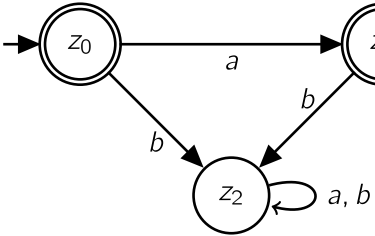
- als Zustandsgraph

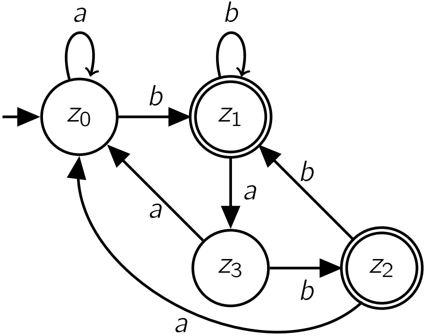
Darstellungsformen

- als 5-Tupel $(\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_0, z_1\})$ mit $\delta(z_0, a) = z_1 \quad \delta(z_0, b) = z_2$
  $\delta(z_1, a) = z_1 \quad \delta(z_1, b) = z_2$
  $\delta(z_2, a) = z_2 \quad \delta(z_2, b) = z_2$
- mit Übergangstabelle für δ:

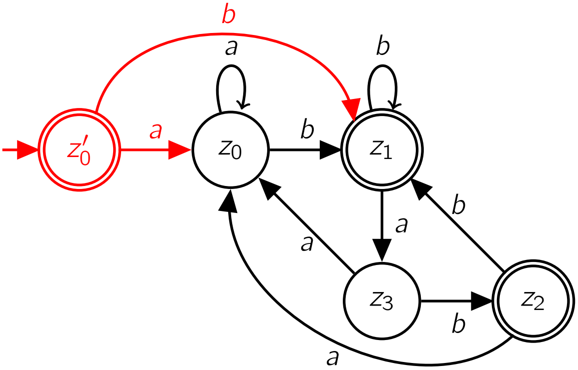
δ	a	b
z_0	z_1	z_2
z_1	z_1	z_2
z_2	z_2	z_2

- als Zustandsgraph





- Welcher Zustand ist der Startzustand?
- Was sind die Endzustände?
- Wo entlang läuft der Automat auf dem Wort ababab?
- Erkennt der Automat ε? Wenn nein, was müsste man ändern, damit der Automat ε erkennt? Wenn ja, was müsste man ändern, damit der Automat ε nicht erkennt?



- Welcher Zustand ist der Startzustand?
- Was sind die Endzustände?
- Wo entlang läuft der Automat auf dem Wort ababab?
- Erkennt der Automat ε? Wenn nein, was müsste man ändern, damit der Automat ε erkennt? Wenn ja, was müsste man ändern, damit der Automat ε nicht erkennt?

Uhrzeiten erkennen

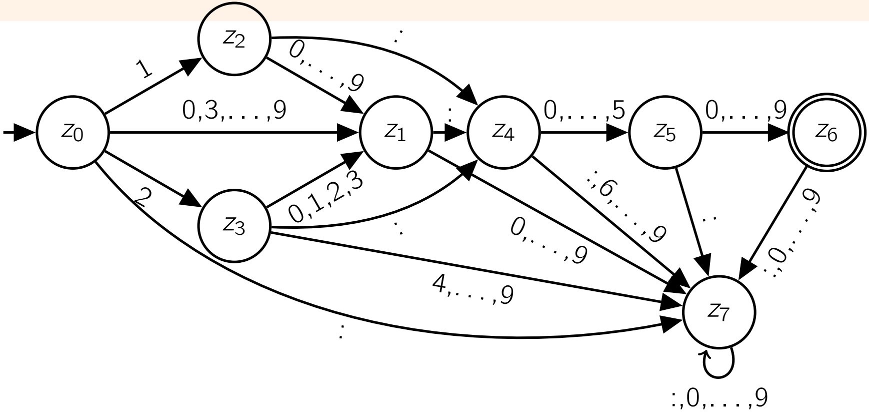
Aufgabe

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

Uhrzeiten erkennen

Aufgabe

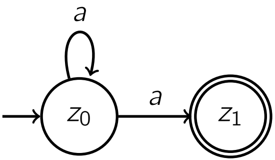
Geben Sie einen **DFA** an, der genau alle gülten Uhrzeiten im 24-Stunden-Format der Form Stunde:Minute mit Stunde $\in \{0, \ldots, 23\}$ und Minute $\in \{00, 01, \ldots, 59\}$ erkennt.



Endliche Automaten: DFA & NFA

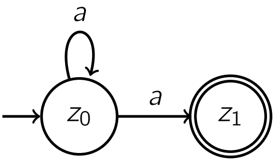
	DFA	NFA
Zustände	Z	Z
Alphabet	Σ	Σ
Übergang	$\delta :: (Z \times \Sigma) \to Z$	$\delta : (Z \times \Sigma) \to \mathcal{P}(Z)$
Startzust.	$z_0 \in Z$	$S \subseteq Z$
Endzust.	$E \subseteq Z$	$E \subseteq Z$
Übergang	$\widehat{\delta} : (Z \times \Sigma^*) \to Z$	$\widehat{\delta} : (\mathcal{P}(Z) \times \Sigma^*) \to \mathcal{P}(Z)$
f. Wörter	$\widehat{\delta}(z, \varepsilon) = z$	$\widehat{\delta}(X, \varepsilon) = X$
	$\widehat{\delta}(z, aw) = \widehat{\delta}(\delta(z, a), w)$	$\widehat{\delta}(X, aw) = \widehat{\delta}(\bigcup_{z \in X} \delta(z, a), w)$
$w \in L(M)$	$\widehat{\delta}(z_0, w) \in E$	$\widehat{\delta}(S, w) \cap E \neq \emptyset$

DFA oder NFA oder nix? (1)



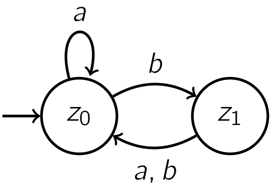
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (1)



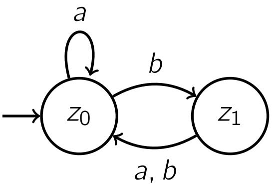
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (2)



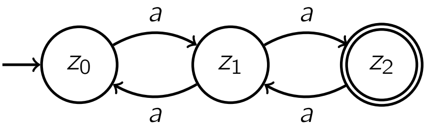
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (2)



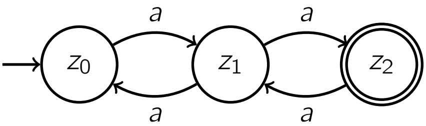
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (3)



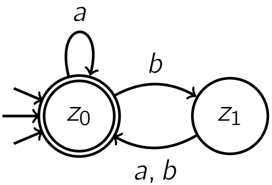
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (3)



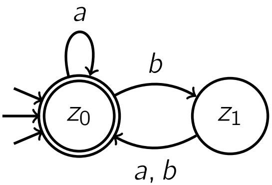
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (4)



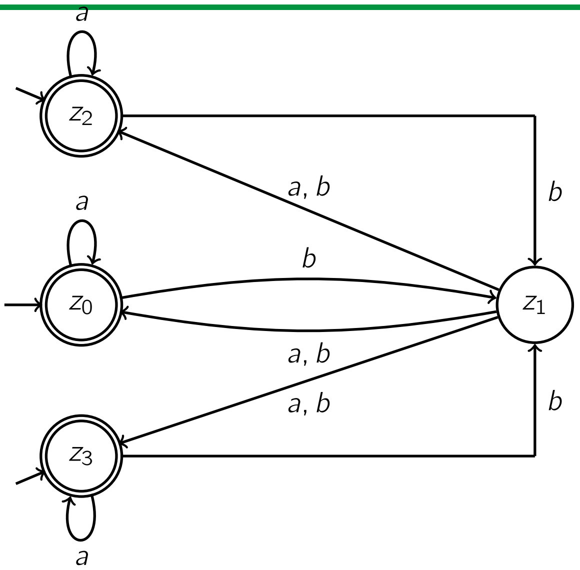
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (4)



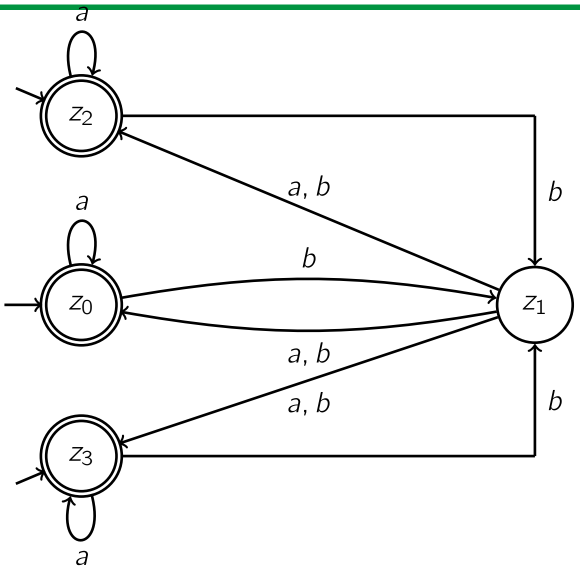
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (5)



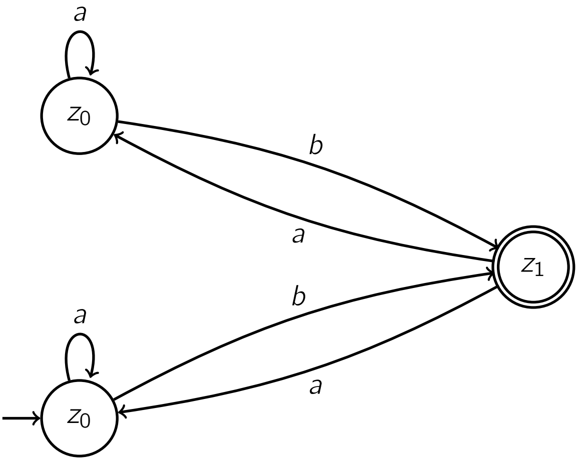
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (5)



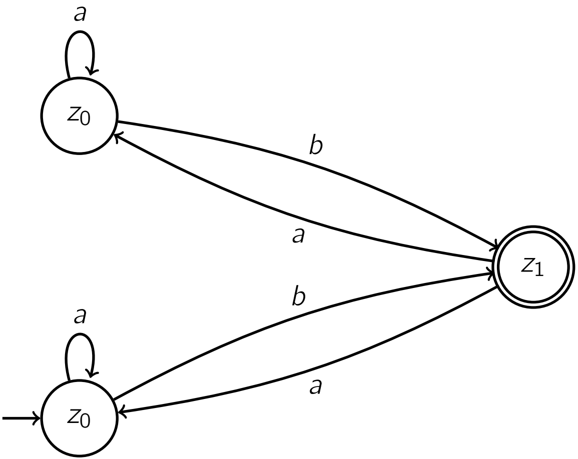
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (6)



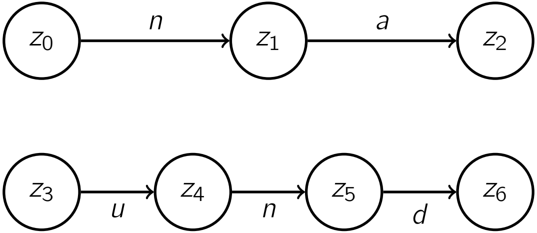
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (6)



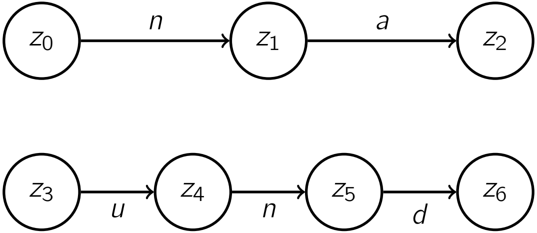
1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (7)



1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

DFA oder NFA oder nix? (7)



1. Ist ein DFA
2. Ist ein NFA, aber kein DFA
3. Ist weder DFA noch NFA

NFAs konstruieren – Nichtdeterminismus ausnutzen

Aufgabe (vgl. Hopcroft, Motwani, Ullman)

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

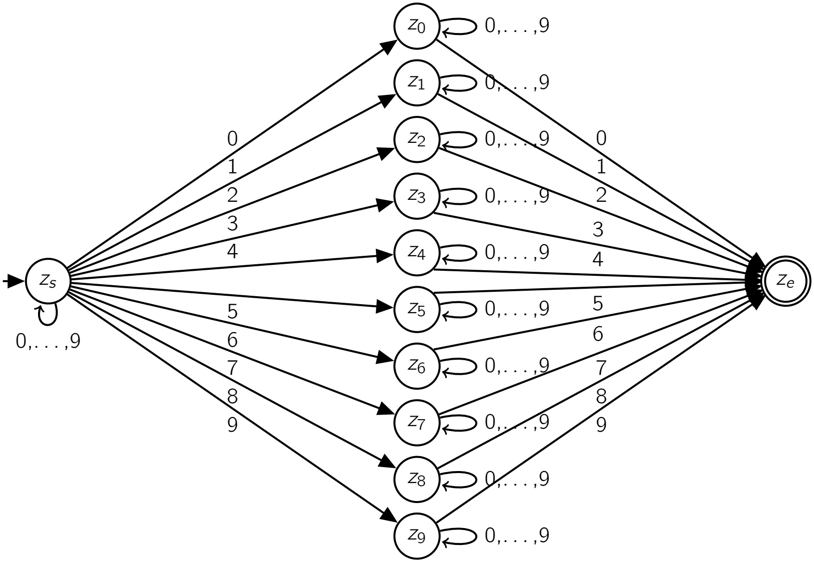
a) die letzte Ziffer vorher schon mindestens einmal in der Folge vorkam

b) die letzte Ziffer vorher nicht schon einmal vorkam

Nutzen Sie den Nichtdeterminismus möglichst gut aus.

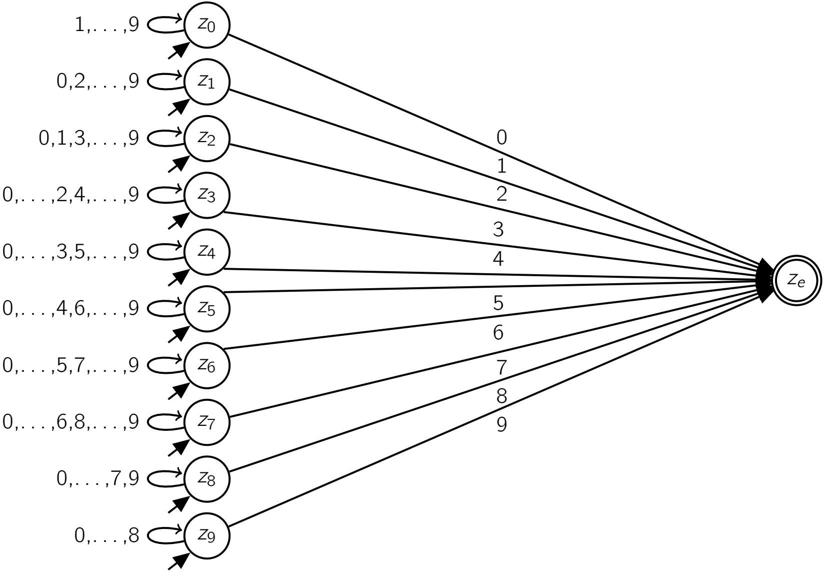
a) die letzte Ziffer vorher schon mindestens einmal in der Folge vorkam

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



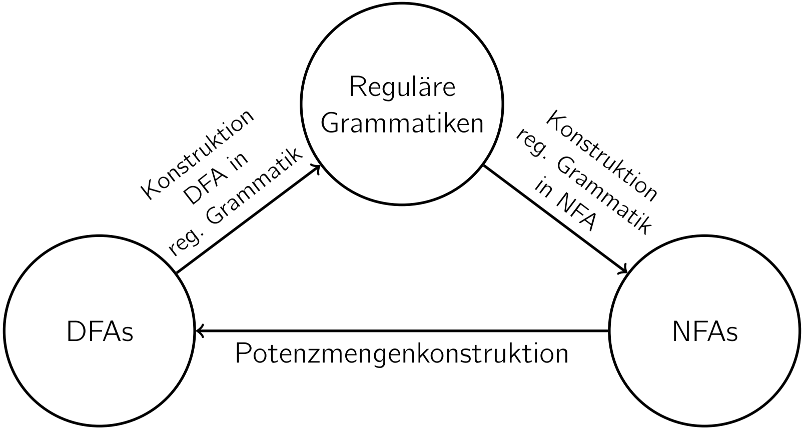
b) die letzte Ziffer vorher nicht schon einmal vorkam

b) die letzte Ziffer vorher nicht schon einmal vorkam



Reguläre Sprachen und Endliche Automaten

Äquivalenz der Formalismen:



Nochmal: Beweisstruktur

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

Ein Spiel

Spielfeld:

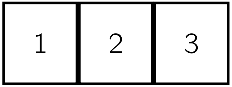
1	2	3

Aktionen:

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

Ein Spiel

Spielfeld:



Aktionen:

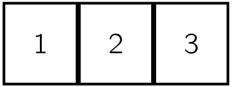
- ℓ und r: Wechseln um ein Feld nach ℓinks bzw. $r$echts.
- ℓ bei Feld 1 und r bei Feld 3 bleiben wirkungslos.

Aufgabe

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

Ein Spiel

Spielfeld:

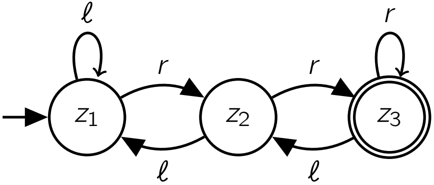


Aktionen:

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

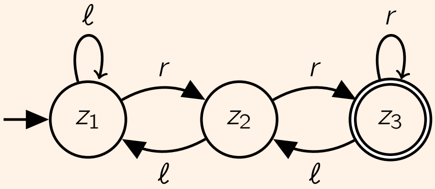
Aufgabe

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



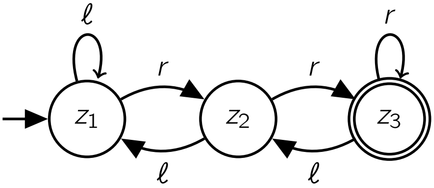
DFA $\to$ reg. Grammatik

Aufgabe

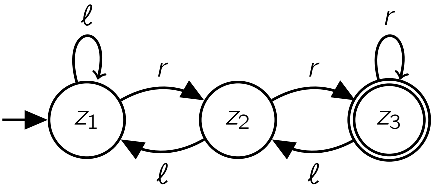


Geben Sie zum gezeigten DFA eine reguläre Grammatik an, welche die vom DFA akzeptierte Sprache erzeugt.

DFA $\rightarrow$ reguläre Grammatik (2)



DFA → reguläre Grammatik (2)

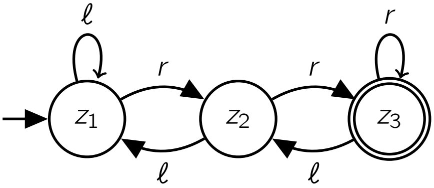


Erinnerung: DFA → reguläre Grammatik

Für DFA $M = (Z, \Sigma, \delta, z_0, E)$ konstruiere $G = (V, \Sigma, P, S)$ mit

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

DFA $\rightarrow$ reguläre Grammatik (2)



Erinnerung: DFA $\rightarrow$ reguläre Grammatik

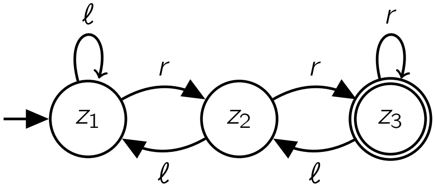
Für DFA $M = (Z, \Sigma, \delta, z_0, E)$ konstruiere
$G = (V, \Sigma, P, S)$ mit
$V = Z$,
$S = z_0$,

$$\begin{aligned} P = \quad & \{z_i \rightarrow az_j \mid \delta(z_i, a) = z_j\} \\ \cup \quad & \{z_i \rightarrow a \mid \delta(z_i, a) = z_j \wedge z_j \in E\} \\ \cup \quad & \{z_0 \rightarrow \varepsilon \mid \text{falls } z_0 \in E\} \end{aligned}$$

Grammatik dazu:

$$\begin{aligned} G &= (V, \Sigma, P, S) \text{ mit} \\ V &= \{z_1, z_2, z_3\} \\ S &= z_1 \\ P &= \end{aligned}$$

DFA → reguläre Grammatik (2)



Erinnerung: DFA → reguläre Grammatik

Für DFA $M = (Z, \Sigma, \delta, z_0, E)$ konstruiere
$G = (V, \Sigma, P, S)$ mit

$V = Z$,

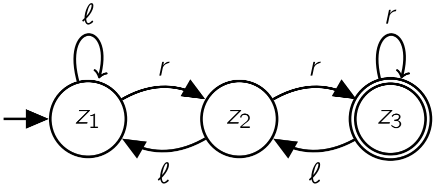
$S = z_0$,

$$\begin{aligned} P = \quad & \{z_i \to a z_j \mid \delta(z_i, a) = z_j\} \\ \cup \quad & \{z_i \to a \mid \delta(z_i, a) = z_j \land z_j \in E\} \\ \cup \quad & \{z_0 \to \varepsilon \mid \text{falls } z_0 \in E\} \end{aligned}$$

Grammatik dazu:

$$\begin{aligned} G &= (V, \Sigma, P, S) \text{ mit} \\ V &= \{z_1, z_2, z_3\} \\ S &= z_1 \\ P &= \{z_1 \to \ell z_1, z_1 \to r z_2, \\ &\quad z_2 \to \ell z_1, z_2 \to r z_3, \\ &\quad z_3 \to \ell z_2, z_3 \to r z_3, \end{aligned}$$

DFA $\to$ reguläre Grammatik (2)



Erinnerung: DFA $\to$ reguläre Grammatik

Für DFA $M = (Z, \Sigma, \delta, z_0, E)$ konstruiere $G = (V, \Sigma, P, S)$ mit

$V = Z$,

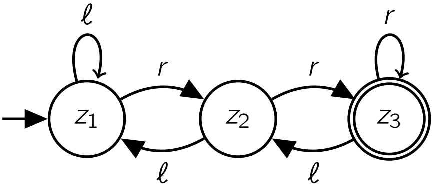
$S = z_0$,

$$
\begin{aligned}
P = \quad & \{z_i \to az_j \mid \delta(z_i, a) = z_j\} \\
\cup \quad & \{z_i \to a \mid \delta(z_i, a) = z_j \wedge z_j \in E\} \\
\cup \quad & \{z_0 \to \varepsilon \mid \text{falls } z_0 \in E\}
\end{aligned}
$$

Grammatik dazu:

$$
\begin{aligned}
G &= (V, \Sigma, P, S) \text{ mit} \\
V &= \{z_1, z_2, z_3\} \\
S &= z_1 \\
P &= \{z_1 \to \ell z_1, z_1 \to r z_2, \\
&\quad z_2 \to \ell z_1, z_2 \to r z_3, \\
&\quad z_3 \to \ell z_2, z_3 \to r z_3, \\
&\quad z_2 \to r, z_3 \to r\}
\end{aligned}
$$

DFA $\rightarrow$ reguläre Grammatik (2)



Erinnerung: DFA $\rightarrow$ reguläre Grammatik

Für DFA $M = (Z, \Sigma, \delta, z_0, E)$ konstruiere $G = (V, \Sigma, P, S)$ mit

$V = Z$,

$S = z_0$,

$$\begin{aligned} P = \; & \{z_i \rightarrow a z_j \mid \delta(z_i, a) = z_j\} \\ \cup \; & \{z_i \rightarrow a \mid \delta(z_i, a) = z_j \wedge z_j \in E\} \\ \cup \; & \{z_0 \rightarrow \varepsilon \mid \text{falls } z_0 \in E\} \end{aligned}$$

Grammatik dazu:

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

Reguläre Grammatik → NFA (1)

Aufgabe

$G = (\{S, A, D, E\}, \{\ell, r\}, P, S)$ beschreibt Spielzugfolgen, mit
$P = \{S \to \ell A \mid rS,\ A \to \ell D \mid \ell E \mid \ell \mid rS,\ D \to \ell \mid \ell D \mid rA \mid \ell E,\ E \to \ell \mid \ell D\}$.
Konstruieren Sie einen NFA M mit $L(M) = L(G)$.

Reguläre Grammatik $\to$ NFA (1)

Aufgabe

$G = (\{S, A, D, E\}, \{\ell, r\}, P, S)$ beschreibt Spielzugfolgen, mit
$P = \{S \to \ell A \mid rS,\ A \to \ell D \mid \ell E \mid \ell \mid rS,\ D \to \ell \mid \ell D \mid rA \mid \ell E,\ E \to \ell \mid \ell D\}$.
Konstruieren Sie einen NFA M mit $L(M) = L(G)$.

Zur Erinnerung: reg. Grammatik $\to$ NFA

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

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

Reguläre Grammatik → NFA (1)

Aufgabe

$G = (\{S, A, D, E\}, \{\ell, r\}, P, S)$ beschreibt Spielzugfolgen, mit
$P = \{S \to \ell A \mid rS,\ A \to \ell D \mid \ell E \mid \ell \mid rS,\ D \to \ell \mid \ell D \mid rA \mid \ell E,\ E \to \ell \mid \ell D\}$.
Konstruieren Sie einen NFA M mit $L(M) = L(G)$.

Zur Erinnerung: reg. Grammatik → NFA

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

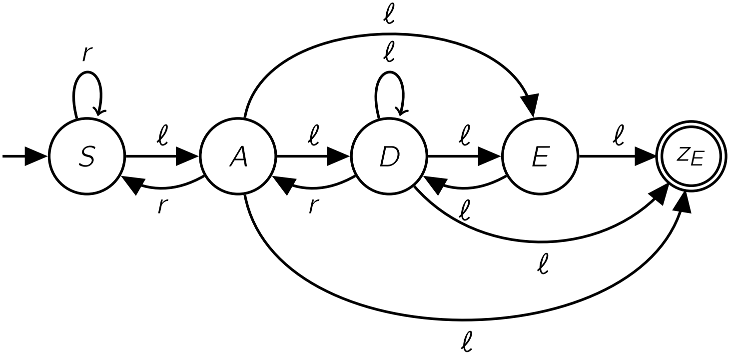
$$\begin{aligned}
Z &= V \dot\cup \{z_E\} \\
S' &= \{S\} \\
E &= \{z_E\} \cup \{S \mid \text{falls } S \to \varepsilon \in P\} \\
\delta(A, a) &= \{N \mid A \to aN \in P\} \\
&\quad \cup \{z_E \mid A \to a \in P\} \\
\delta(z_E, a) &= \emptyset
\end{aligned}$$

$M = (\{S, A, D, E, z_E\}, \{\ell, r\}, \delta, \{S\}, \{z_E\})$ mit

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

Reguläre Grammatik → NFA (2)

Zustandsgraph:



NFA angeben

Aufgabe

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

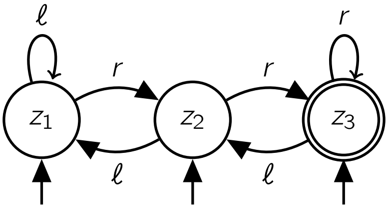
1	2	3

NFA angeben

Aufgabe

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

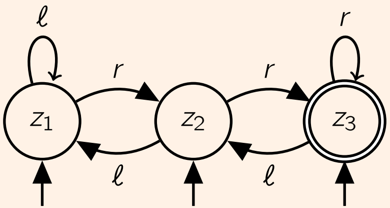
1	2	3



NFA $\to$ DFA

Aufgabe

Überführen Sie den NFA



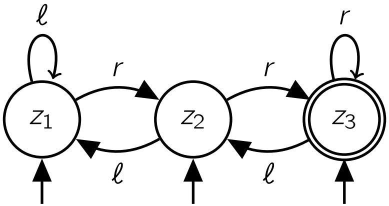
in einen DFA durch Verwendung der Potenzmengenkonstruktion.

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)

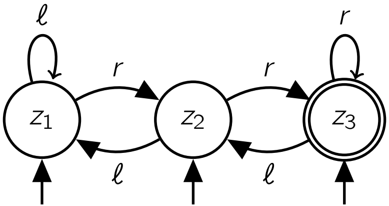
NFA:



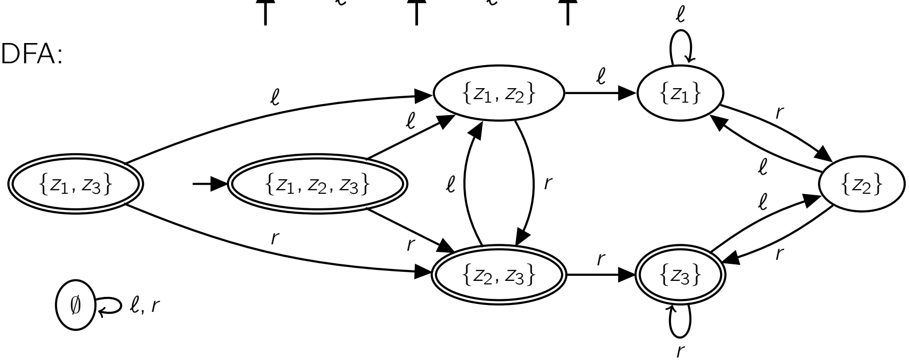
DFA:

NFA → DFA (2)

NFA:

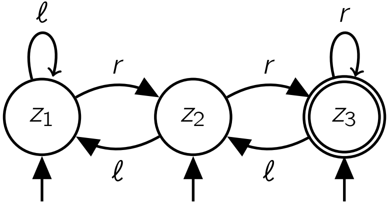


DFA:

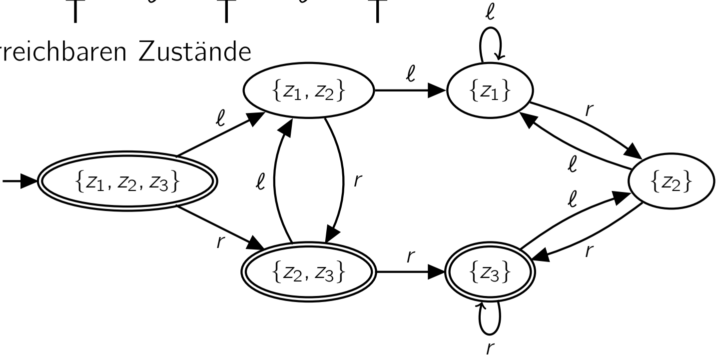


NFA $\to$ DFA (2)

NFA:



DFA: nur die erreichbaren Zustände



Endliche Automaten: DFA & NFA & NFA $+ \varepsilon$

	DFA	NFA	NFA $+ \varepsilon$
Zustände	Z	Z	Z
Alphabet	Σ	Σ	Σ
Übergang	$\delta :: (Z \times \Sigma) \to Z$	$\delta : (Z \times \Sigma) \to \mathcal{P}(Z)$	$\delta : (Z \times \Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Z)$
Startzust.	$z_0 \in Z$	$S \subseteq Z$	$S \subseteq Z$
Endzust.	$E \subseteq Z$	$E \subseteq Z$	$E \subseteq Z$
Übergang	$\widehat{\delta} : (Z \times \Sigma^*) \to Z$	$\widehat{\delta} : (\mathcal{P}(Z) \times \Sigma^*) \to \mathcal{P}(Z)$	$\widetilde{\delta} : (\mathcal{P}(Z) \times \Sigma^*) \to \mathcal{P}(Z)$
f. Wörter	$\widehat{\delta}(z, \varepsilon) = z$	$\widehat{\delta}(X, \varepsilon) = X$	$\widetilde{\delta}(X, \varepsilon) = X$
	$\widehat{\delta}(z, aw) = \widehat{\delta}(\delta(z, a), w)$	$\widehat{\delta}(X, aw) = \widehat{\delta}(\bigcup_{z \in X} \delta(z, a), w)$	$\widetilde{\delta}(X, aw) = \widetilde{\delta}(\bigcup_{z \in X} clos_\varepsilon(\delta(z, a)), w)$
$w \in L(M)$	$\widehat{\delta}(z_0, w) \in E$	$\widehat{\delta}(S, w) \cap E \neq \emptyset$	$\widetilde{\delta}(clos_\varepsilon(S), w) \cap E \neq \emptyset$