Automatentheorie und Formale Sprachen

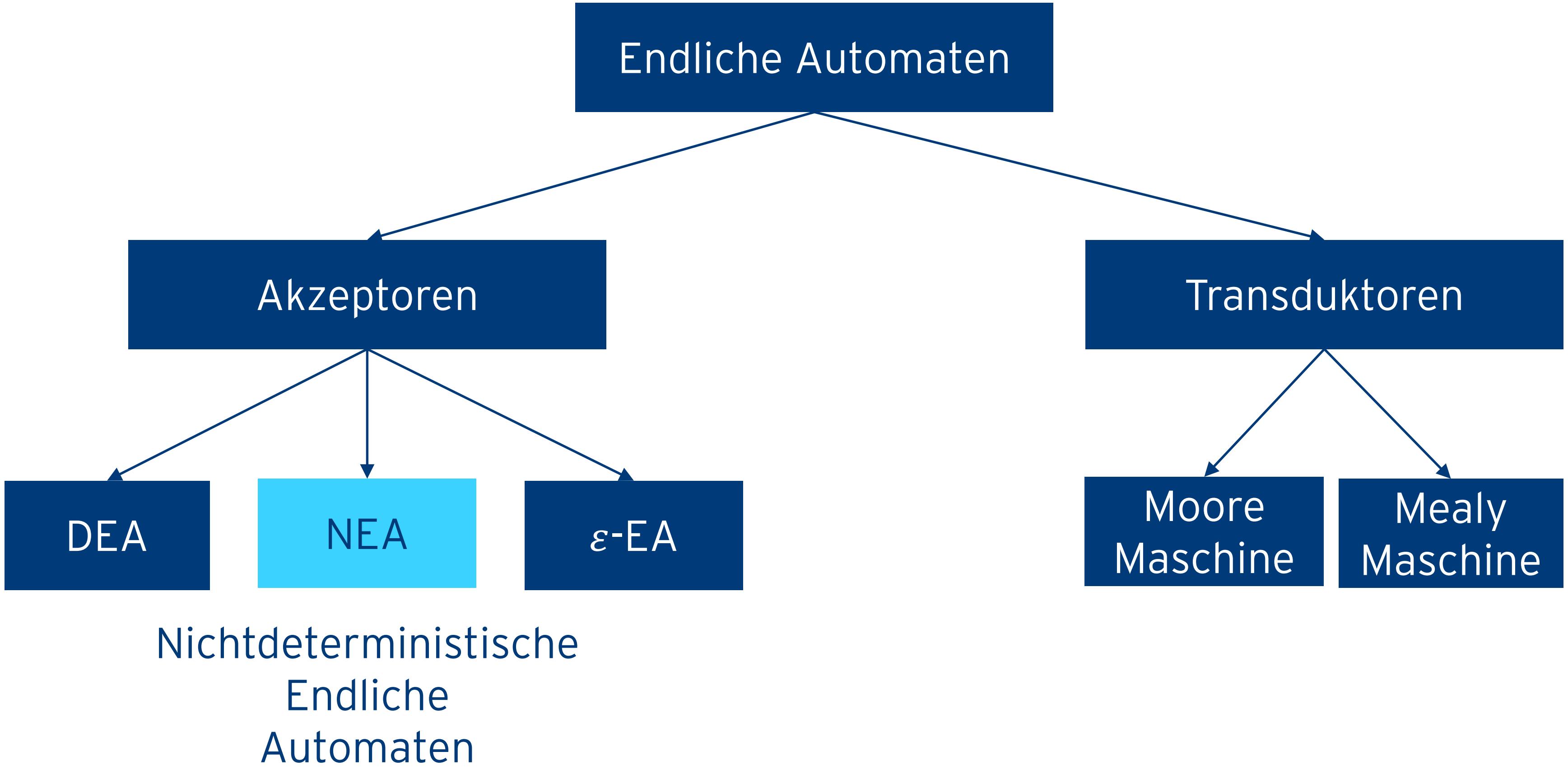
Teil 4 – Nichtdeterministische Endliche Automaten

Prof. Dr. Hans-Werner Sehring

NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

Endliche Automaten – EA

Deterministische Endliche Automaten können je nach Sprache $\boldsymbol{L}$ komplex und in ihrer Darstellung unübersichtlich werden.



Motivation – Erinnerung DEA

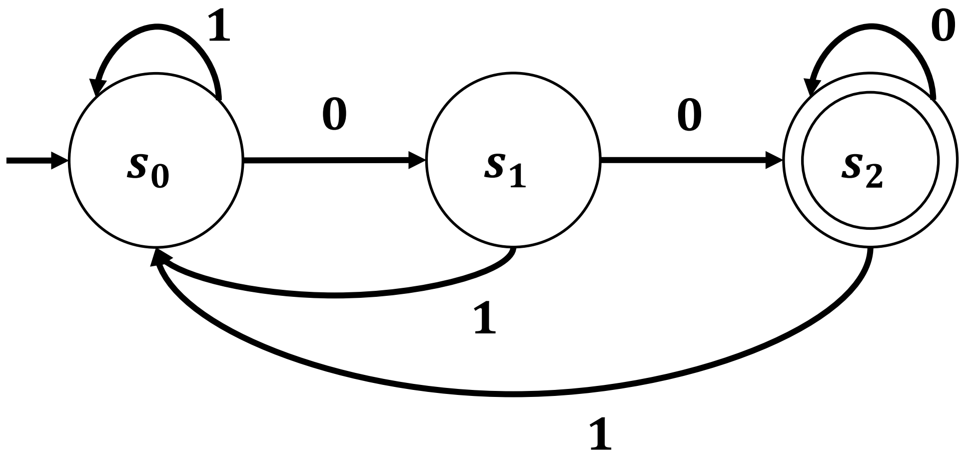
Wir wollen einen DEA für die Sprache der durch vier teilbaren Binärzahlen definieren.

$$\boldsymbol{L = \{w \in \{0,1\}^* | w = uv, u \in \{1,0\}^*, v = 00\}}$$

Extensional:

$$\boldsymbol{L = \{100, 1000, 1100, 10000, 10100, 11100, \dots\}}$$

Ein DEA $\boldsymbol{A_D}$ mit $\mathbf{L}(\boldsymbol{A_D}) = \boldsymbol{L}$:



Der Automat ist übersichtlich, aber die Sprache ist „schwer zu sehen“.

Verallgemeinerung der Zustandsübergänge

Funktionen sind ein Spezialfall von Relationen (rechtseindeutige Relationen).

Statt als Übergangsfunktion können wir für $\boldsymbol{\delta}$ auch eine Relation $\boldsymbol{R}$ zulassen:

$$\boldsymbol{\delta} = (\boldsymbol{s}, \boldsymbol{a})\boldsymbol{R}\boldsymbol{s}'$$

bzw. über das Tripel

$$\boldsymbol{\delta} = (\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \text{ mit } \boldsymbol{R} \subseteq \boldsymbol{S} \times \boldsymbol{\Sigma} \times \boldsymbol{S}$$

$\boldsymbol{\delta}$ wird eine mengenwertige Funktion

$$\boldsymbol{\delta}: \boldsymbol{S} \times \boldsymbol{\Sigma} \rightarrow \boldsymbol{P}(\boldsymbol{S})$$

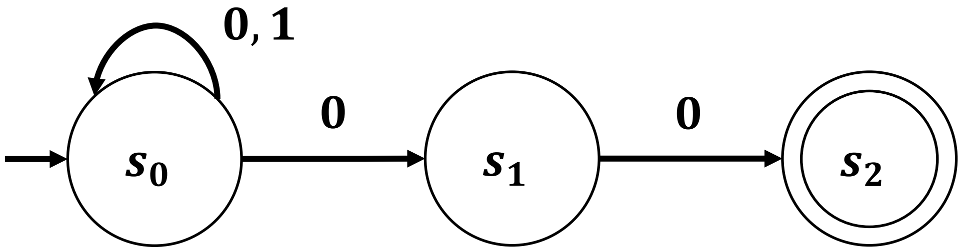
Was passiert, wenn wir für $\boldsymbol{\delta}$ eine allgemeine Relation statt einer Funktion erlauben?

Motivation – NEA für durch vier teilbare Binärzahlen

Wir wollen einen NEA für die Sprache der durch vier teilbaren Binärzahlen definieren.

$$\boldsymbol{L = \{w \in \{0,1\}^* | w = u00v, u \in \{1,0\}^*, v = 00\}}$$

Ein endlicher Automat $\boldsymbol{A_N}$ mit $\boldsymbol{\delta}$ als Relation und mit $\mathbf{L}(\boldsymbol{A_N}) = \boldsymbol{L}$:



Der Automat geht von Zustand $\boldsymbol{s_0}$ in Zustand $\boldsymbol{s_1}$, „wenn er die vorletzte Null liest“.

Veranschaulichung von Nichtdeterminismus

Insbesondere vor dem Hintergrund des Programmierens ist Nichtdeterminismus evtl. zunächst kontraintuitiv.

Mögliche Interpretationen unter deterministischer Denkweise:

- *Der Automat „rät“ mehreren möglichen Zustandsübergängen immer richtig, welche Variante zu einem akzeptierenden Endzustand führt.*
- *Für eine Konfiguration, für die es mehrere mögliche Zustandsübergänge gibt, wird der entsprechende DEA „kopiert“, und alle Kopien laufen in der Konfiguration weiter, jeder mit einem anderen möglichen Zustandswechsel.*

Mathematisch bleibt die Definition sehr einfach:

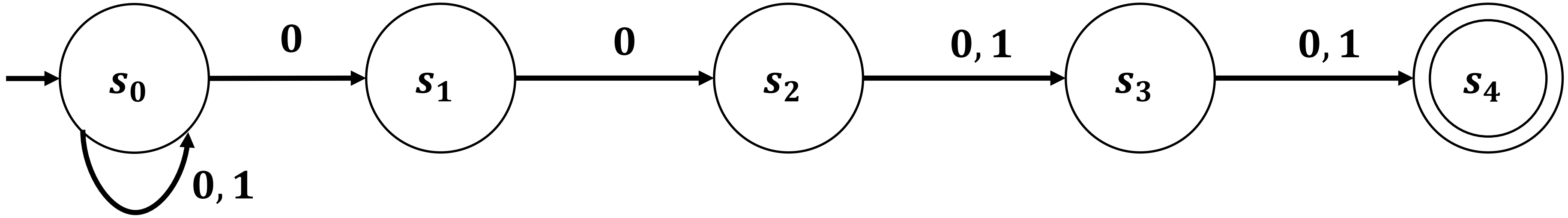
Wenn ein akzeptierender Endzustand für ein Wort durch eine Folge von Zustandstransitionen erreichbar ist, ist das Wort in der Sprache.

Motivation – NEA

Weiteres Beispiel:

$$L = \{w \in \Sigma^* | w = u00v, u \in \{1,0\}^*, v \in \{00,01,10,11\}\}$$

$$L = \{1110011, 1010000, 11111010011, \dots\}$$



Lösungsansatz: Liegt ein Wort vor, dessen Infix die Null enthält, ist eine eindeutige Zuordnung wie bei einem DEA nicht möglich. Beim Lesen der Null werden quasi mehrere Zustände eingenommen, das heißt, der Wertebereich der Zustandsübergangsfunktion wird auf Mengen erweitert.

$$\delta: S \times \Sigma \to P(S)$$

Beispiel:

$\delta(s_0, 0) = \{s_0, s_1\}$, da $\delta(s_0, 0) = s_0$ und $\delta(s_0, 0) = s_1$

Nichtdeterministische Endliche Automaten

Formal wird ein NEA beschrieben durch:

$$\boldsymbol{A = (\Sigma, S, \delta, S_0, F)}$$

Dabei entspricht $\boldsymbol{S_0}$ der Menge der Startzustände, die anderen Mengen sind analog zu einem DEA. Die Zustandsübergangsfunktion $\boldsymbol{\delta}$ führt zu einer Teilmengenbildung über $\boldsymbol{S}$.

Analog zu DEA gilt für Konfigurationen $\boldsymbol{k}$

$$\boldsymbol{(s, aw) \vdash (s', w)}$$

und akzeptierte Sprachen $\boldsymbol{L}$

$$\boldsymbol{L(A) = \{w \in \Sigma^* | (s_0, w) \vdash^* (s', \varepsilon), s_0 \in S_0, s' \in F\}},$$

die in Analogie zu $\boldsymbol{DFA_\Sigma}$ über

$$\boldsymbol{NFA_\Sigma = \bigcup_i L_i,\ \ L_i \subseteq \Sigma^*} \text{ und } \exists \text{ NEA } \boldsymbol{A} \text{ mit } \boldsymbol{L_i = L(A)}$$

in der Klasse $\boldsymbol{NFA_{\ \Sigma}}$ zusammengefasst werden (NFA – Nondeterministic Finite Automata).

Zustandsübergänge von NEA (1)

Die Zustandsübergangsfunktion eines NEA $\boldsymbol{A = (\Sigma, S, \delta, S_0, F)}$ wird mithilfe von Potenzmengen ausgedrückt:

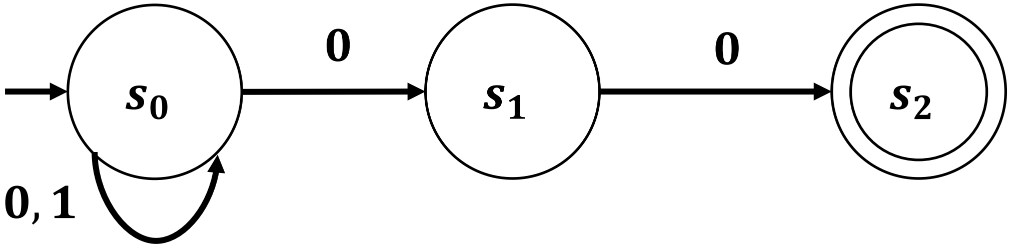
$$\boldsymbol{\delta: S \times \Sigma \rightarrow P(S)}$$

Algorithmisch sinnvoller ist die Erweiterung des Definitionsbereichs der Zustandsübergangsfunktion $\boldsymbol{\delta}$ ebenfalls auf Teilmengen von $\boldsymbol{S}$:

$$\boldsymbol{\delta': P(S) \times \Sigma \rightarrow P(S)}$$

Zustandsübergänge von NEA (2)

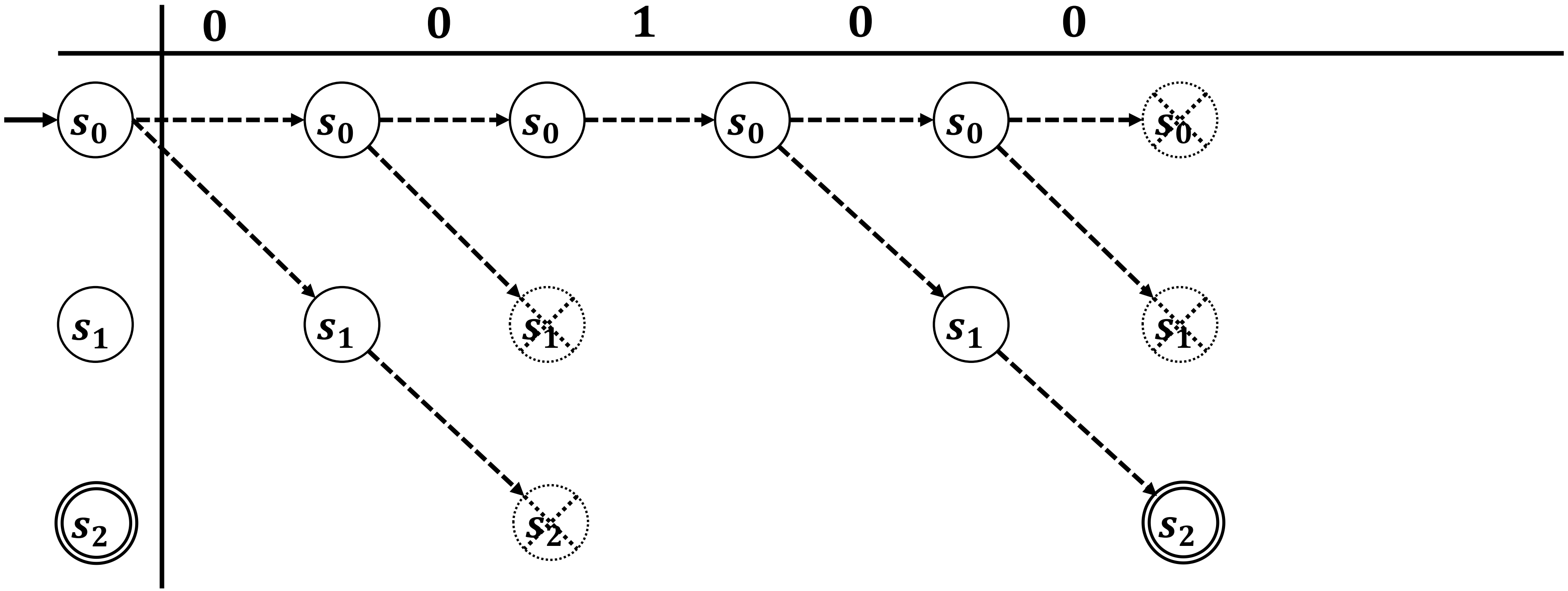
Beispiel: $\boldsymbol{A_N}$



$$w = 00100$$
$$\delta'(\{s_0\}, 0) = \{s_0, s_1\}$$
$$\delta'(\{s_0, s_1\}, 0) = \{s_0, s_1, s_2\}$$
$$\delta'(\{s_0, s_1, s_2\}, 1) = \{s_0\}$$
$$\delta'(\{s_0\}, 0) = \{s_0, s_1\}$$
$$\delta'(\{s_0, s_1\}, 0) = \{s_0, s_1, s_2\}$$
$$\delta'(\{s_0, s_1, s_2\}, \varepsilon) = \{s_0, s_1, s_2\}$$

Zustandsübergänge von NEA (3)

Visualisierung: **Trellis-Schema** für $\boldsymbol{w} = \mathbf{00100}$ und $\boldsymbol{A}_{\mathrm{N}}$



Zustandsübergänge von NEA (4)

Da die Prüfung einer Eingabesequenz $\boldsymbol{w}$ zu „Sackgassen“ führen kann, erfolgt die Prüfung eines Worts über die erweiterte Zustandsübergangsfunktion, die reflexiv-transitive Hülle von $\boldsymbol{\delta}'$:

$$\boldsymbol{\delta}^*: \boldsymbol{P}(\boldsymbol{S}) \times \boldsymbol{\Sigma}^* \rightarrow \boldsymbol{P}(\boldsymbol{S})$$

Die Berechnung von $\boldsymbol{\delta}^*$ unter Berücksichtigung der Zustandsmenge $\boldsymbol{S}' \subseteq \boldsymbol{S}$ erfolgt dabei durch

$$\boldsymbol{\delta}^*(\boldsymbol{S}', \boldsymbol{aw}) = \boldsymbol{\delta}^*\left(\bigcup_{\boldsymbol{s} \in \boldsymbol{S}'} \boldsymbol{\delta}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{w}\right), \boldsymbol{a} \in \boldsymbol{\Sigma}, \boldsymbol{w} \in \boldsymbol{\Sigma}^*$$

und

$$\forall \boldsymbol{S}' \subseteq \boldsymbol{S}\colon\ \boldsymbol{\delta}^*(\boldsymbol{S}', \boldsymbol{\varepsilon}) = \boldsymbol{S}'$$

Zustandsübergänge von NEA (5)

Beispiel:

Berechnung der Akzeptanz der Eingabesequenz $\boldsymbol{w} = \mathbf{110011}$ für den Automaten $\boldsymbol{A}$, der die Sprache $\boldsymbol{L} = \{\boldsymbol{w} \in \boldsymbol{\Sigma}^* | \boldsymbol{w} = \boldsymbol{u}\mathbf{00}\boldsymbol{v}, \boldsymbol{u} \in \{\mathbf{1},\mathbf{0}\}^*, \boldsymbol{v} \in \{\mathbf{00},\mathbf{01},\mathbf{10},\mathbf{11}\}\}$ akzeptiert.

$$\boldsymbol{\delta}^*(\{\boldsymbol{s}_0\}, \mathbf{110011}) = \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{1}), \mathbf{10011}\right) = \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{1}), \mathbf{0011}\right)$$

$$= \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{0}), \mathbf{011}\right) = \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0,s_1\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{0}), \mathbf{11}\right) = \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0,s_1,s_2\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{1}), \mathbf{1}\right)$$

$$= \boldsymbol{\delta}^*\left(\bigcup_{s\in\{s_0,s_3\}} \boldsymbol{\delta}(\boldsymbol{s},\mathbf{1}), \boldsymbol{\varepsilon}\right) = \{\boldsymbol{s}_0, \boldsymbol{s}_4\}$$

Das Wort $\boldsymbol{w}$ wird akzeptiert, da

$$\{\boldsymbol{s}_0, \boldsymbol{s}_4\} \cap \boldsymbol{F} \neq \emptyset$$

Äquivalenz NEA und DEA

Die Klassen der DEA und der NEA sind äquivalent, $\boldsymbol{DFA_\Sigma \equiv NFA_\Sigma}$

- Zu jedem DEA gibt es einen NEA, der die gleiche Sprache erkennt. Und umgekehrt.
  $\forall$ DEA $\boldsymbol{A_D}$: $\exists$ NEA $\boldsymbol{A_N}$: $\boldsymbol{L(A_D) = L(A_N)}$ und $\forall$ NEA $\boldsymbol{A_N}$: $\exists$ DEA $\boldsymbol{A_D}$: $\boldsymbol{L(A_D) = L(A_N)}$
- DEA und NEA erkennen die gleiche Sprachklasse, $\boldsymbol{DFA_\Sigma = NFA_\Sigma}$

Somit:

$$\boldsymbol{REG_\Sigma = DFA_\Sigma = NFA_\Sigma}$$

Beweis der Äquivalenz von NEA und DEA (1)

$$DFA_{\Sigma} = NFA_{\Sigma}$$

Beweis: durch Zeigen beider Mengeninklusionen

„⊆" Überführung DEA in NEA

DEA sind als Teilmenge bzw. als Sonderfall eines NEA zu betrachten, bei dem pro Eingabesymbol $\boldsymbol{a} \in \boldsymbol{\Sigma}$ nur ein Folgezustand $\boldsymbol{s}' \in \boldsymbol{S}$ statt einer Zustandsmenge resultiert.

Gegeben sei ein deterministischer Automat $\boldsymbol{A_{DEA}} = (\boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{\delta_{DEA}}, \boldsymbol{s_0}, \boldsymbol{F})$

der durch folgende Vorschrift $\boldsymbol{A_{NEA}} = (\boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{\delta_{NEA}}, \{\boldsymbol{s_0}\}, \boldsymbol{F})$

mit $$\boldsymbol{\delta_{NEA}}(\{\boldsymbol{s}\}, \boldsymbol{a}) = \begin{cases} \{\boldsymbol{s}'\}, \boldsymbol{gdw}\ \boldsymbol{\delta_{DEA}}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{s}' \\ \emptyset \end{cases}$$

in einen NEA $\boldsymbol{A_{NEA}}$ transformiert wird.

Beweis der Äquivalenz von NEA und DEA (2)

Es gilt:

$$\boldsymbol{\delta^*}_{NEA}(\{s_0\}, w) \cap F \neq \emptyset \; gdw \; \boldsymbol{\delta^*}_{DEA}(s_0, w) \in F$$

denn

$$\delta^*_{NEA}(\{s_0\}, aw) = \delta^*_{NEA}\left(\bigcup_{s \in \{s_0\}} \delta_{NEA}(s, a), w\right) = \delta^*_{NEA}(\delta_{NEA}(\{s_0\}, a), w) = \delta^*_{NEA}(\{\delta_{DEA}(s_0, a)\}, w)$$

$$\delta^*_{NEA}(\delta_{DEA}(s_0, a), w) \cap F \neq \emptyset \; gdw \; \delta^*_{DEA}(\delta_{DEA}(s_0, a), w) \in F$$

und

$$\delta^*_{NEA}(\{s_0\}, \varepsilon) = \delta^*_{DEA}(s, \varepsilon) = \{s\}, \{s\} \cap F \neq \emptyset \; gdw \; s \in F$$

wodurch die Behauptung

$$DFA_\Sigma \subseteq NFA_\Sigma$$

erfüllt ist bzw. DEA stellen als Sonderfall eine Teilmenge von NEA dar.

Beweis der Äquivalenz von NEA und DEA (3)

„⊇“ Überführung NEA in DEA

NEA sind als Teilmenge bzw. als Sonderfall eines DEA zu betrachten, bei dem die Zustandsmengen auf einzelne Zustände à la $\boldsymbol{P}\{\boldsymbol{S}\} \rightarrow \boldsymbol{S}$ abgebildet werden.

Vorgehensweise (Konstruktion des **Potenzautomaten**):

Die Vorschrift zur Identifikation neuer Zustände bzw. der Zustände des korrespondierenden DEA‘ lautet:

$$\boldsymbol{\delta_{DEA'}(NEA)\colon P(S) \times \Sigma \rightarrow P(S)}$$

wobei die Konvention

$$\boldsymbol{f\colon \{S_{DEA'} \in P(S)\} \rightarrow \{s_j^*\}}$$

für das Mapping relevant ist, d.h., die neuen Zustände $\boldsymbol{S}$ werden über eine Abbildung $\boldsymbol{f}$ auf eine festzulegende Menge abgebildet (Beispielsweise $\{\boldsymbol{s_0}, \boldsymbol{s_1}\} \rightarrow \boldsymbol{s_1}^*$ etc.).

Beweis der Äquivalenz von NEA und DEA (4)

Die Berechnung der abzubildenden Teilmengen $\boldsymbol{S'}$ lässt sich über die erweiterte Zustandsübergangsfunktion

$$\boldsymbol{\delta'}: \boldsymbol{P}(\boldsymbol{S}) \times \boldsymbol{\Sigma} \rightarrow \boldsymbol{P}(\boldsymbol{S})$$

bzw.

$$\boldsymbol{\delta_{DEA}}(\boldsymbol{S'}, \boldsymbol{a}) = \bigcup_{\boldsymbol{s} \in \boldsymbol{S'}} \boldsymbol{\delta}(\boldsymbol{s}, \boldsymbol{a})$$

durchführen, wobei folgende Bedingung gilt:

$$\mathbf{max}(|\boldsymbol{S_{DEA}}|) = |\boldsymbol{P}(\boldsymbol{S_{NEA}})|$$

Das heißt, die maximale Anzahl der Zustände des zu konstruierenden DEA* entspricht der Anzahl der Elemente der Potenzmenge der Zustandsmenge des NEA.

Der zu entwerfende Automat DEA* erhält damit folgende Gestalt:

$$\boldsymbol{DEA}^* = \left(\boldsymbol{\Sigma}, \{\boldsymbol{S_{DEA}} \subseteq \boldsymbol{P}(\boldsymbol{S_{NEA}})\}, \boldsymbol{\delta_{DEA}}, \boldsymbol{S_0}^{\boldsymbol{DEA}}, \boldsymbol{F_{DEA}}\right)$$

mit

$$\boldsymbol{S_0}^{\boldsymbol{DEA}} = \{\boldsymbol{s_0}\}$$

$$\boldsymbol{F_{DEA}} = \{\boldsymbol{S'} \subseteq \boldsymbol{S} | \boldsymbol{S'} \cap \boldsymbol{F_{NEA}} \neq \emptyset\}$$

Beweis der Äquivalenz von NEA und DEA (5)

Somit gilt

$$\boldsymbol{DFA_\Sigma \subseteq NFA_\Sigma}$$

und

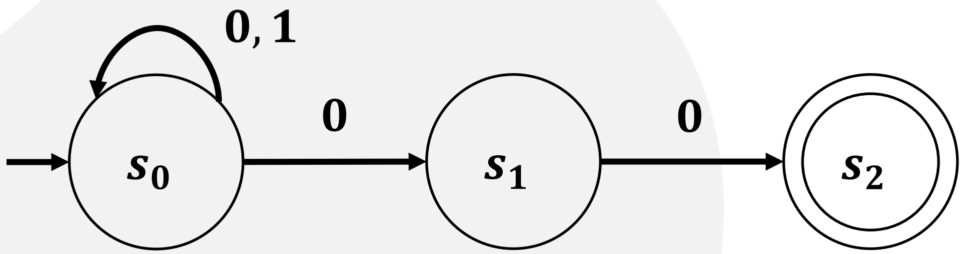
$$\boldsymbol{DFA_\Sigma \supseteq NFA_\Sigma}$$

und somit

$$\boldsymbol{REG_\Sigma = DFA_\Sigma = NFA_\Sigma}$$

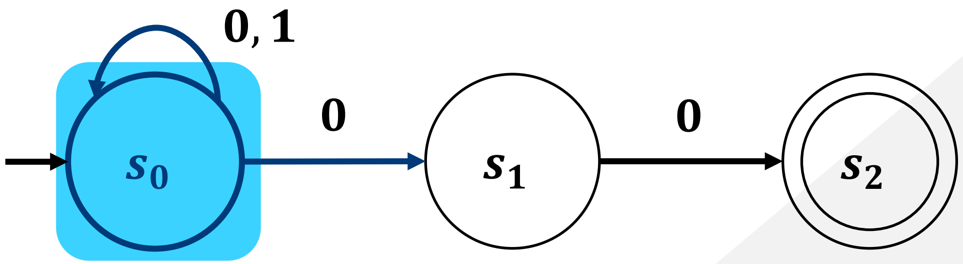
q.e.d.

Beispiel für Potenzautomaten – Zustände

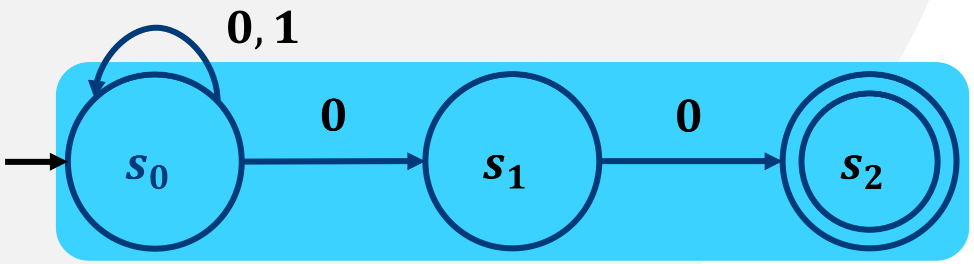


NEA $\boldsymbol{A_N}$ zum Akzeptieren von durch vier teilbaren Binärzahlen (s.o.):
Erstellen der Zustände des Potenzautomaten:

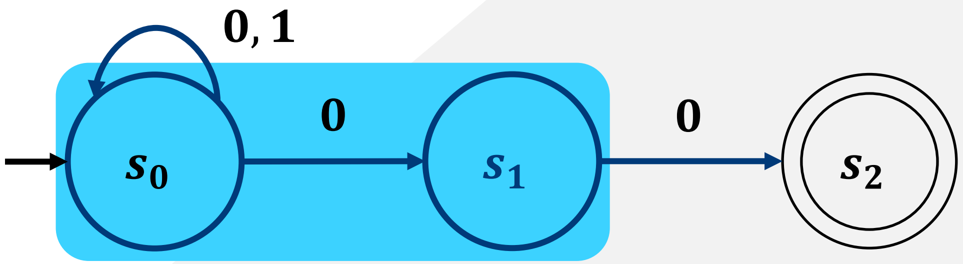
1. $\{\boldsymbol{s_0}\}$ (aus dem Startzustand $\boldsymbol{s_0}$ von $\boldsymbol{A_N}$)



2. $\{\boldsymbol{s_0}, \boldsymbol{s_1}\}$ (aus $\boldsymbol{\delta(s_0, 0)}$ von $\boldsymbol{A_N}$)

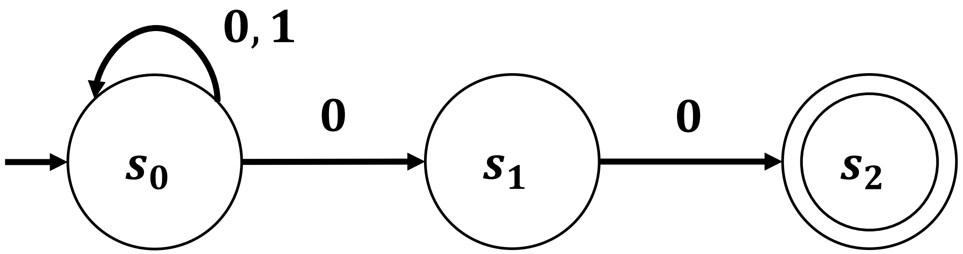


3. $\{\boldsymbol{s_0}, \boldsymbol{s_1}, \boldsymbol{s_2}\}$ (aus $\boldsymbol{\delta(s_0, 0)}$, $\boldsymbol{\delta(s_1, 0)}$ von $\boldsymbol{A_N}$)

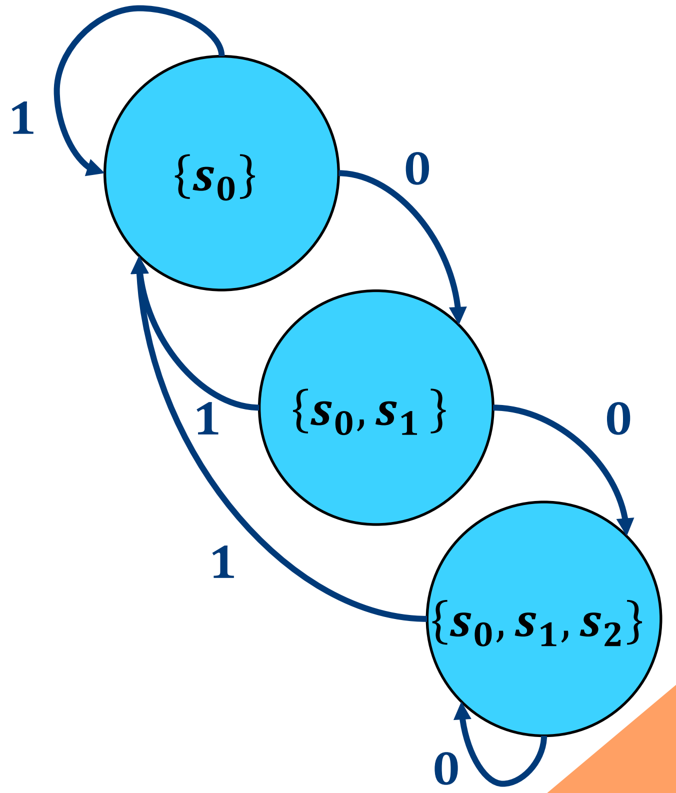


Beispiel für Potenzautomaten – Zustandsübergänge

NEA $\boldsymbol{A_N}$ zum Akzeptieren von durch vier teilbaren Binärzahlen (s.o.):
Erstellen der Zustandsübergänge des Potenzautomaten:



4. $\boldsymbol{\delta(\{s_0\}, 0) = \{s_0, s_1\}}$
   (aus $\boldsymbol{\delta(s_0, 0) = s_0}$ und $\boldsymbol{\delta(s_0, 0) = s_1}$ von $\boldsymbol{A_N}$)
5. $\boldsymbol{\delta(\{s_0\}, 1) = \{s_0\}}$ (aus $\boldsymbol{\delta(s_0, 1) = s_0}$ von $\boldsymbol{A_N}$)
6. $\boldsymbol{\delta(\{s_0, s_1\}, 0) = \{s_0, s_1, s_2\}}$
   (aus $\boldsymbol{\delta(s_0, 0) = s_0}$, $\boldsymbol{\delta(s_0, 0) = s_1}$ und $\boldsymbol{\delta(s_1, 0) = s_2}$ von $\boldsymbol{A_N}$)
7. $\boldsymbol{\delta(\{s_0, s_1\}, 1) = \{s_0\}}$ (aus $\boldsymbol{\delta(s_0, 1) = s_0}$ von $\boldsymbol{A_N}$)
8. $\boldsymbol{\delta(\{s_0, s_1, s_2\}, 0) = \{s_0, s_1, s_2\}}$
   (aus $\boldsymbol{\delta(s_0, 0) = s_0}$, $\boldsymbol{\delta(s_0, 0) = s_1}$ und $\boldsymbol{\delta(s_1, 0) = s_2}$ von $\boldsymbol{A_N}$)
9. $\boldsymbol{\delta(\{s_0, s_1, s_2\}, 1) = \{s_0\}}$ (aus $\boldsymbol{\delta(s_0, 1) = s_0}$ von $\boldsymbol{A_N}$)

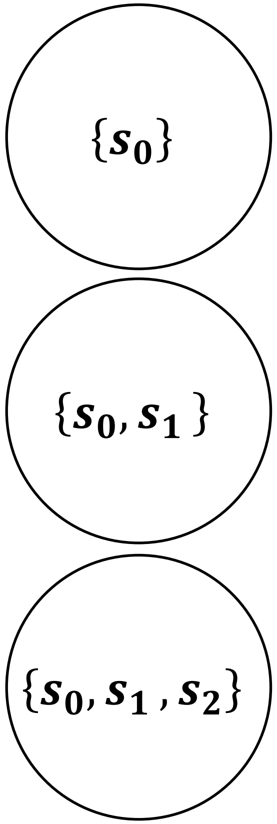


Beispiel für Potenzautomaten – Start- u. Endzustände

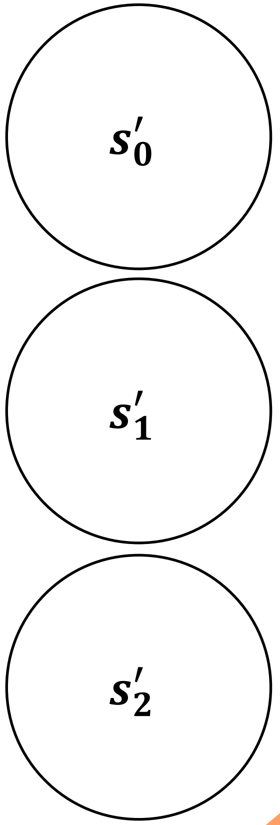
Startzustand des Potenzautomaten ist: $\{\boldsymbol{s_0}\}$

Akzeptierende Endzustände sind diejenigen, in die die akzeptierenden Endzustände des NEA, hier $\boldsymbol{s_2}$ von $\boldsymbol{A_N}$, eingangen sind: $\{\boldsymbol{s_0}, \boldsymbol{s_1}, \boldsymbol{s_2}\}$

Schließlich können die Zustände noch mit einer Korrespondenztabelle sinnvoll benannt werden.

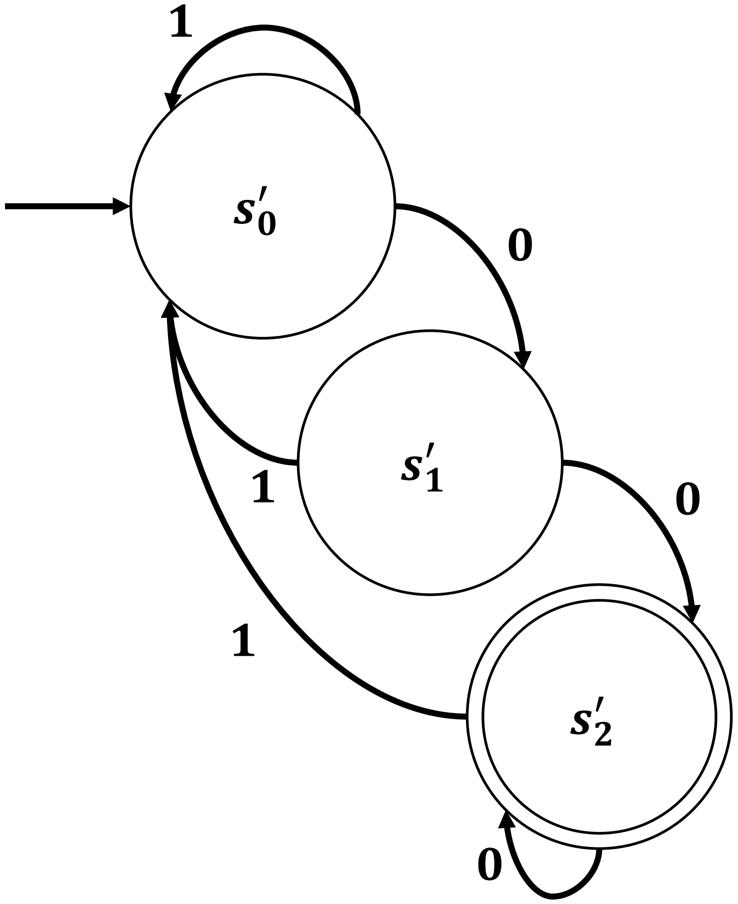


s_0	s'_0
s_0, s_1	s'_1
s_0, s_1, s_2	s'_2



Beispiel für Potenzautomaten – DEA

Der resultierende Automat ist deterministisch.

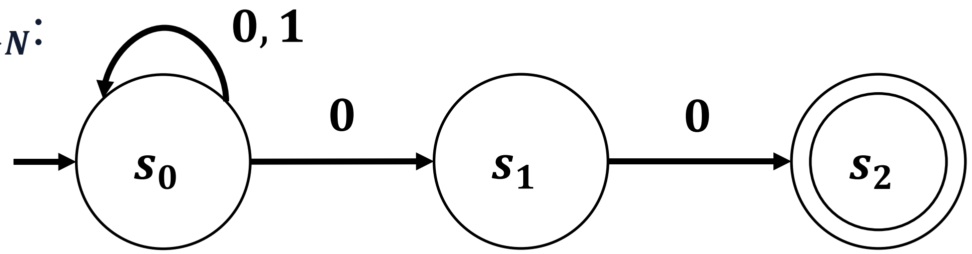


Konstruktion von Potenzautomaten (1)

Tabellarischer Ansatz zur Ermittlung der Zustandsübergangsfunktion:

1. Tabelle mit Spalte für Zustand(smenge) und Spalten für Zeichen aus Σ.
2. Erste Zeile mit Startzustand
3. Füllen der jeweils nächsten Zeile mit erreichbaren Zuständen
4. Ergänzen einer Zeile für jede dabei neu entstandene Zustandsmenge
5. Weiter mit 3 bis keine neuen Zustandsmengen
6. Identifikation der akzeptierenden Endzustände

Beispiel für $\boldsymbol{A_N}$:



	0	1
s_0	s_0, s_1	s_0
s_0, s_1	s_0, s_1, s_2	s_0
$\boldsymbol{s_0, s_1, s_2}$	s_0, s_1, s_2	s_0

Konstruktion von Potenzautomaten (2)

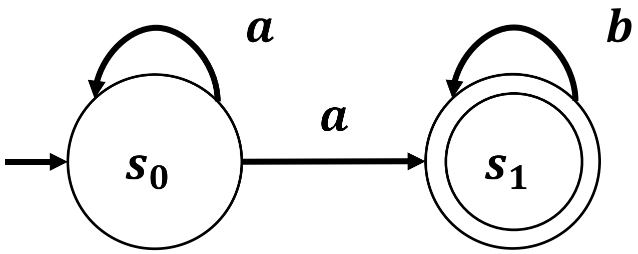
Bei partieller Zustandsübergangsrelation ist der Fall der leeren Menge erreichbarer Zustände zu beachten.

Im tabellarischen Ansatz:

3. Gibt es in einem/r Zustand/smenge für ein Zeichen aus Σ keinen Übergang, ist die erreichbare Zustandsmenge leer.

Die leere Menge entspricht im Potenzmengenautomaten dem Fangzustand.

Beispiel:



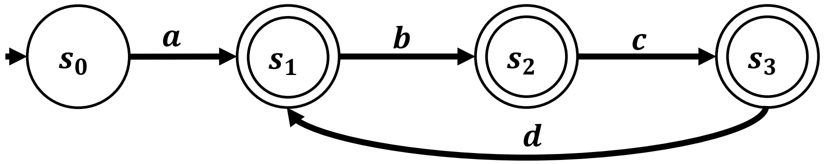
	a	b
s_0	s_0, s_1	$\emptyset$
s_0, s_1	s_0, s_1	s_1
s_1	$\emptyset$	s_1
$\emptyset$	$\emptyset$	$\emptyset$

Abgeschlossenheit von unter Vereinigung (erneut)

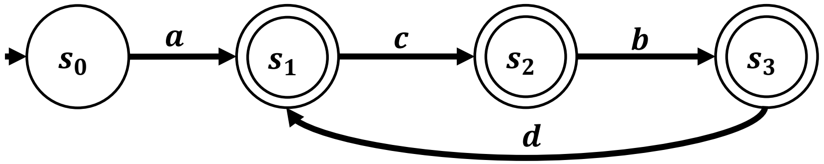
Für $\boldsymbol{L_1}$ und $\boldsymbol{L_2}$ reguläre Sprachen ist auch $\boldsymbol{L_1} \cup \boldsymbol{L_2}$ eine reguläre Sprache.

Beweisskizze: Zu $\boldsymbol{L_1}$ und $\boldsymbol{L_2}$ reguläre Sprachen muss es also NEAs $\boldsymbol{A_1}$ und $\boldsymbol{A_2}$ geben mit $\boldsymbol{L}(\boldsymbol{A_1}) = \boldsymbol{L_1}$und $\boldsymbol{L}(\boldsymbol{A_2}) = \boldsymbol{L_2}$. Für den Beweis ist ein NEA $\boldsymbol{A}_3$ zu konstruieren mit $\boldsymbol{L}(\boldsymbol{A_3}) = \boldsymbol{L_1} \cup \boldsymbol{L_2}$.

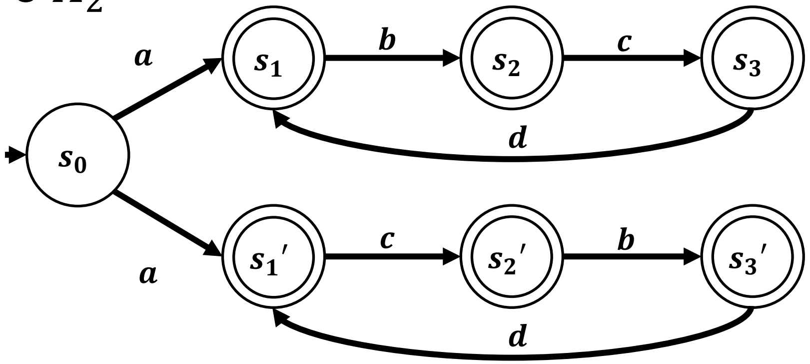
$\boldsymbol{A}_1$



$\boldsymbol{A}_2$



$\boldsymbol{A}_1 \cup \boldsymbol{A}_2$



Jetzt könnten wir den Beweis etwas leichter mit NEA führen (immer noch Problem mit Rückkanten zu $\boldsymbol{s_0}$)

Zusammenfassung

Nichtdeterministische Endliche Automaten (**NEA**) verallgemeinern Deterministische Endliche Automaten (DEA) durch eine Menge von Startzuständen und eine mengenwertige Zustandsübergangsfunktion.

DEA lassen sich als NEA darstellen und umgekehrt. Die Konzepte sind **gleichmächtig**.

Es gilt: $\boldsymbol{DFA_\Sigma \equiv NFA_\Sigma}$

NEA sind eine grafisch und formal kompaktere Darstellungsform als DEA.
DEA kommen allerdings der rechnergestützten Implementierung näher.

NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

NORDAKADEMIE gAG Hochschule der Wirtschaft

Köllner Chaussee 11 · 25337 Elmshorn · Tel.: +49 (0) 4121 4090-0 · E-Mail: info@nordakademie.de · Web: www.nordakademie.de