University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Theoretische Informatik

Prof. Dr. Juho Mäkiö

21.11. 2018

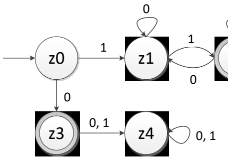
AUTOMATEN MIT Ɛ-ÜBERGÄNGEN

Nachweis: DEA < NEA

1. DEA A → es gibt äquivalenten NEA A‘
2. NEA A → es gibt äquivalenten DEA A

zu 1.

Da die Eindeutigkeit (Determinismus) eine spezielle Form der Mehrdeutigkeit ist (nur eine Wahl) ist diese Richtung einfach.



A = (Z, ∑, δ, z_0, E) mit

Z = {z_0, z_1, z_2, z_3, z_4}

∑ = {0, 1}

δ : s. Automatengraph

z_0 = z_0

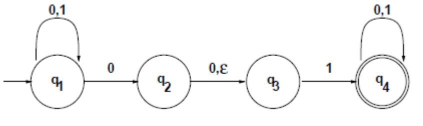
E = {z_2, z_3}

	0	1
z_0	{z_3}	{z_1}
z_1	{z_1}	{z_2}
$z_{2/E}$	{z_1}	{z_2}
$z_{3/E}$	{z_4}	{z_4}
z_4	{z_4}	{z_4}

~ immer nur eine Wahl

L(A) = L_{NEA} (A‘)

NEA

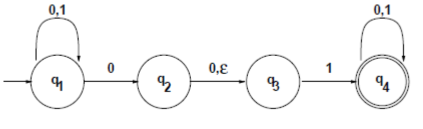


- Unterschiede zum DEA
  - Bei einem NEA ist es erlaubt, dass es
    - für ein gegebens Zeichen *a* bei einem Zustand mehrere Übergänge gibt
    - für ein gegebenes Zeichen *a* bei einem Zustand keinen Übergang für *a* gibt
    - dass einer oder mehrere Übergänge mit dem leeren Wort *ε* beschriftet ist.

Fragen?

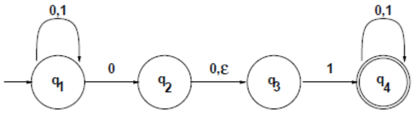
- Was passiert wenn mehrere Übergänge möglich sind?
- Was passiert, wenn kein Übergang vorhanden ist?
- Was bedeuten ε-Übergänge?

Funktionsweise eines NEA



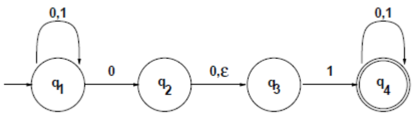
- Was passiert wenn für das aktuelle Zeichen mehrere Übergänge möglich sind?
  - Der Automat „teilt“ sich in mehreren Kopien
  - Jede Kopie folgt einer Möglichkeit
  - Die Kopien zusammen folgen allen Möglicheiten
  - Der Automat akzeptiert ein Wort ω, falls nach dem Lesen von ω eine oder mehrere Kopien einen akzeptierende Zustand erreichen

Funktionsweise eines NEA



- Was passiert wenn für das aktuelle Zeichen kein Übergang vorhanden ist?
  - Diese Kopie des Automaten „stirbt“

Funktionsweise eines NEA

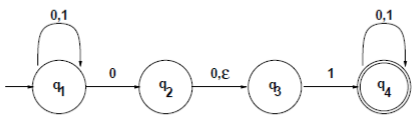


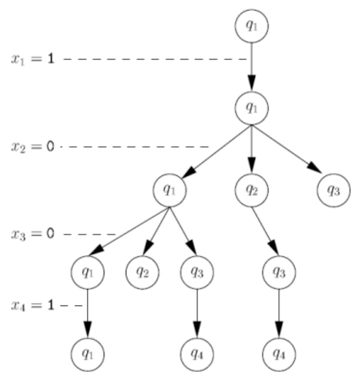
Was bedeuten ε-Übergänge?

- Der Automat „teilt“ sich in mehrere Kopien ohne ein Zeichen zu lesen
  - für jeden ε-Übergang eine, und
  - in eine Kopie, die im ursprünglichen Zustand bleibt

Funktionsweise eines NEA

Was passiert für den Input x = 1001?



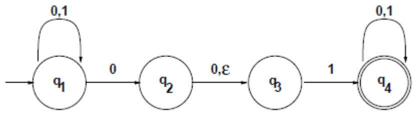


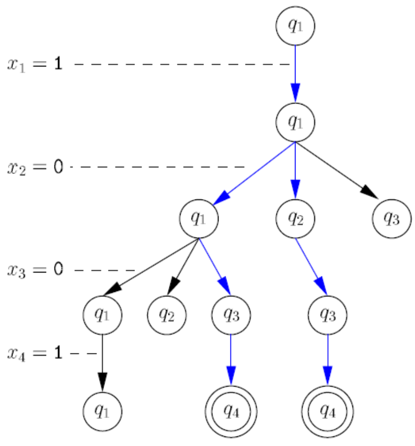
Simulation

- Initialisierung:
  - Startzustand markieren;
  - alle Zustände markieren, die mit ε-Übergängen erreichbar sind (→die ε-Hülle des Startzustandes).
- Für jedes gelesene Eingabezeichen:
  - Markiere alle Folgezustände unter dem Eingabezeichen;
  - markiere alle Zustände, die mit ε-Übergängen erreichbar sind (also die ε-Hülle dieser Folgezustände).
- Wenn ein Endzustand erreicht wird und das Wort ist vollständig abgearbeitet, hat der Automat das Eingabewort erkannt.

Funktionsweise eines NEA

Wird der String x = 1001 akzeptiert?





Nichtdeterminismus als Spiel

Intuitive Interpretation

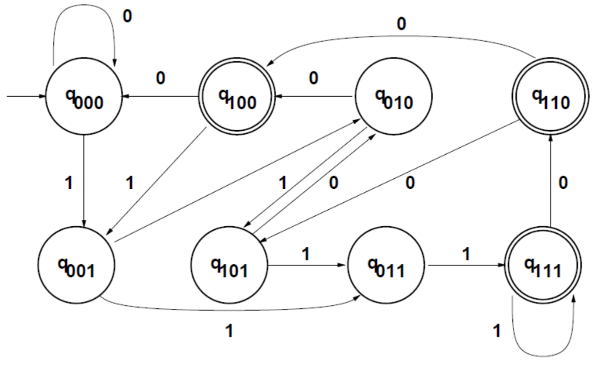
- Man kann die Übergangsfunktion eines NEA als Spielregeln für ein 1-Personen-Spiel auffassen.
- Ähnlich wie in einem Spiel kann der Spieler oft zwischen mehreren möglichen Zügen wählen
- Der NEA akzeptiert den Input ω, falls es in diesem Spiel eine Folge von Spielzügen gibt, bei der der Spieler nach dem Lesen von ω in einem Endzustand landet.

Wozu brauchen wir NEAs

- **Theorem:** Zu jedem NEA N gibt es einem DEA M, der die selbe Sprache akzeptiert, d.h. L(N) = L(M).
- Warum dann NEAs?
  - NEAs kann man als eine höhere Abstraktion von DEAs auffassen.

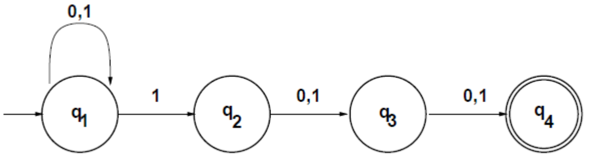
NEA als eine Abstraktion

- Beispiel: Konstruieren Sie einen endlichen Automaten DEA der alle Worte $\omega = \omega_1 \ldots \omega_n$ mit $\omega_{n-2} = 1$ akzeptiert (das drittletzte Zeichen ist eine 1)



NEA als eine Abstraktion

- Beispiel: Konstruieren Sie einen endlichen Automaten NEA der alle Worte $\omega = \omega_1 \ldots \omega_n$ mit $\omega_{n-2} = 1$ akzeptiert (das drittletzte Zeichen ist eine 1)

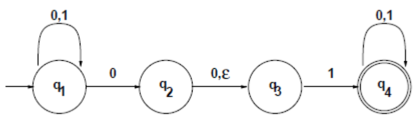


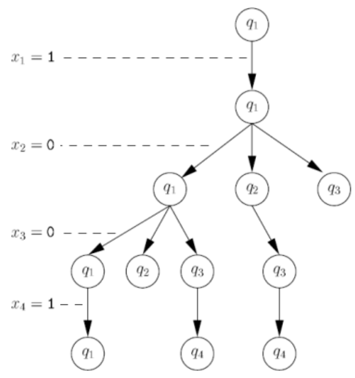
Äquivalenz von NEAs und DEAs

- Theorem: Zu jedem *NEA N* gibt es einen *DEA M*, der die selbe Sprache akzeptiert, d.h. $L(N) = L(M)$
- Beweisidee: Sei de *NEA* $N = (Z, \Sigma, \delta, z_0, E)$ gegeben. Wie konstruieren einen *DEA* $M = (Z', \Sigma, \delta', z'_0, E')$ der den *NEA N* simuliert, d.h. $L(M) = L(N)$
- Betrachten zuerst ohne ε-Übergänge.

Funktionsweise eines NEA

Was passiert für den Input x = 1001?

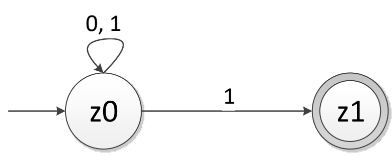




- Der Automat *M* muß sich „merken“, welche Kopie des *NEA N* sich gerade in welchem zustand *q ϵ Z* befindet.

Ein einfaches Beispiel

L(A) = ?{ω1 |ω ∈ {0, 1}*}



A ist ein NEA:

δ	0	1
z_0	$\{z_0\}$	$\{z_0, z_1\}$
$z_{1/E}$	$\varnothing$	$\varnothing$

nicht deterministisch!

Konstruktion eines DEA's A‘
mit L(A‘) = LNEA (A) – WIE?

Beispiel

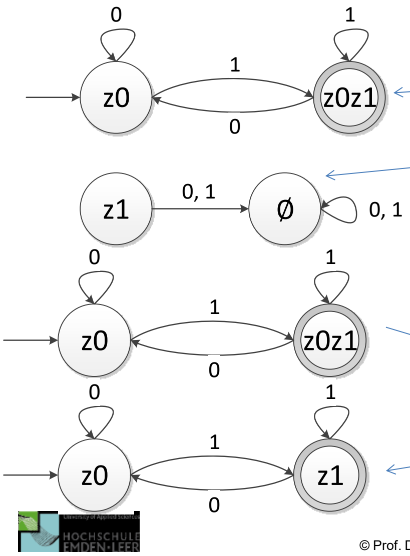
$A = (Z, \sum, \delta, z_0, E)$ mit
$Z = \{z_0, z_1\}$,
$\sum = \{0, 1\}$, $E = \{z_1\}$

δ	0	1
z_0	$\{z_0\}$	$\{z_0, z_1\}$
$z_{1/E}$	$\varnothing$	$\varnothing$

$A' = (Z', \sum, \delta', z_0', E')$
$Z' = P(Z) = \{\emptyset, \{z_0\}, \{z_1\}, \{z_0, z_1\}\}$
$z_0' = \{z_0\}$
$E' = \{\{z_1\}, \{z_0, z_1\}\}$

δ'	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\{z_0\}$	$\{z_0\}$	$\{z_0, z_1\}$
$\{z_{1/E}\}$	$\varnothing$	$\varnothing$
$\{z_0, z_{1/E}\}$	$\{z_0\}$	$\{z_0, z_1\}$

Beispiel



einfache Notation: $\{z_0, z_1\} \triangleq z_0z_1$

Überflüssig

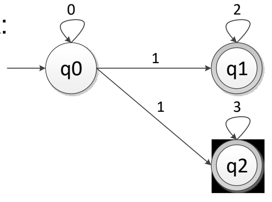
oder (isomorph)

Die erkannte Sprache?

$L(A') = L_{NEA}(A) = \{\omega 1 \mid \omega \in \{0, 1\}^*\}$

Beispiel

A:



A ist NEA (Warum?)

L_{NEA} (A) = ? $\{0^n\ 1\ 2^m\} \cup \{0^n\ 1\ 3^m\}$

δ	0	1	2	3
q_0	$\{q_0\}$	$\{q_1, q_2\}$	$\emptyset$	$\emptyset$
$q_{1/E}$	$\emptyset$	$\emptyset$	$\{q_1\}$	$\emptyset$
$q_{2/E}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_2\}$

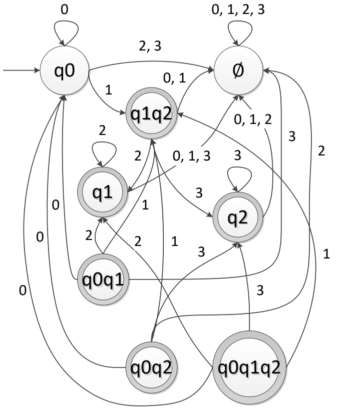
Rest?

Beispiel

	δ'	0	1	2	3
	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
	$\{q_0\}$	$\{q_0\}$	$\{q_1, q_2\}$	$\emptyset$	$\emptyset$
	$\{q_{1/E}\}$	$\emptyset$	$\emptyset$	$\{q_1\}$	$\emptyset$
	$\{q_{2/E}\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_2\}$
x	$\{q_0, q_{1/E}\}$	$\{q_0\}$	$\{q_1, q_2\}$	$\{q_1\}$	$\emptyset$
x	$\{q_0, q_{2/E}\}$	$\{q_0\}$	$\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
	$\{q_{1/E}, q_{2/E}\}$	$\emptyset$	$\emptyset$	$\{q_1\}$	$\{q_2\}$
x	$\{q_0, q_{1/E}, q_{2/E}\}$	$\{q_0\}$	$\{q_1, q_2\}$	$\{q_1\}$	$\{q_2\}$

Tipp: Nicht erreichbare Zustände X gleich weglassen!

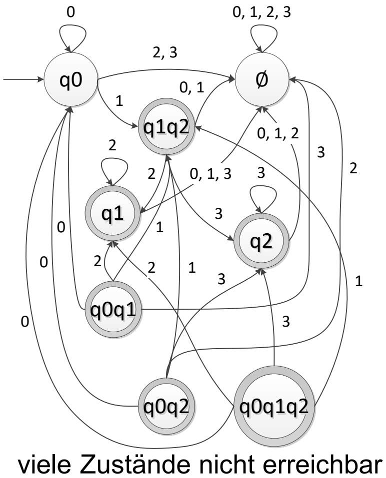
A‘



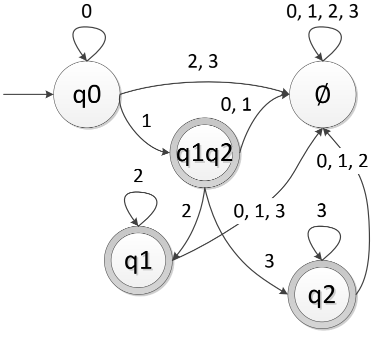
L(A‘) = L_{NEA} (A) ??

Schwer erkennbar

Beispiel



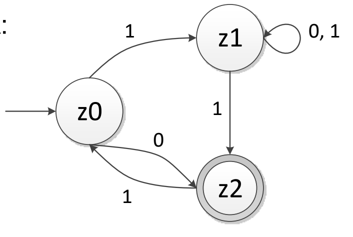
viele Zustände nicht erreichbar
(gleich bei der Konstruktion weglassen)



L(A‘) = L_{NEA} (A)
Hier zu erkennen – vor allem dann, wenn man auch noch den Fehlerzustand ∅ weglieβe

Übungsaufgabe

Gegeben ist der folgende NEA A:



- Geben Sie δ in Tabellenform an.
- Welche Sprache/Wörter akzeptiert/erkennt der NEA A?
- Konstruieren Sie einen DEA A‘ mit L(A‘) = L_{NEA} (A).
- Entfernen Sie aus A‘ nicht erreichbare Zustände
- Geben Sie einen isomorphen DEA mit vereinfachten Zustandsnamen an.

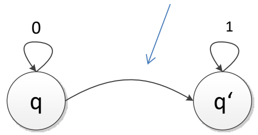
Automaten mit ε-Übergängen

Angenommen, man wollte für folgende Sprache einen Automaten (DEA, NEA) beschreiben:

$\{0^l\ 1^m\ 2^n \mid l, m, n \in \mathbb{N}\}$

~ 3 Zyklen: -> 0
-> 1
-> 2

Problem: Wechsel



Bisher: Zustandswechsel nur bei Verarbeitung eines Zeichens.

$\delta(z, a) = \ldots$

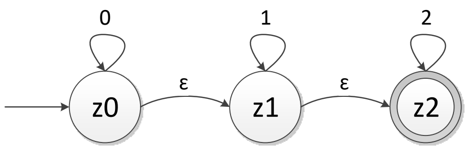
Alternative: Zustandswechsel ohne Verarbeitung eines Zeichens:
ε- Übergang (ε ≙ leeres Wort)

Automaten mit ε-Übergängen

Bisher:

DEA
NEA } ohne ε-Übergänge

Jetzt:



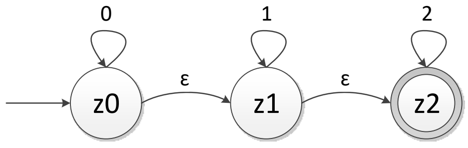
Damit akzeptiert A folgende Wörter:

ε
012
…

Der Einfachheit halber werden nur NEA's um ε-Übergänge erweitert…

Automaten mit ε-Übergängen

- <u>NEA/ε</u> : A = (Z, $\sum$, δ, z_0, E)
  - NEA mit ε-Übergängen
  - Entsprechend: -> , $->_a$, > , ->*
  - (Konfiguration) + $->_\varepsilon$, $\overset{\varepsilon}{>}$

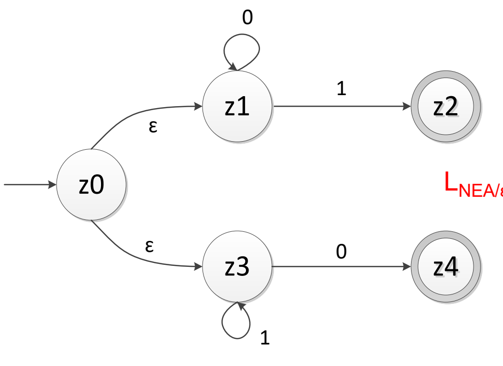


(z_0, 00122) $->_0$ (z_0, 0122)
$->_0$ (z_0, 122)
$->_\varepsilon$ (z_1, 122)
$->_1$ (z_1, 22)
$->_\varepsilon$ (z_2, 22)
$->_2$ (z_2, 2)
$->_2$ (z_2, ε)

Damit wird 00122 von diesem NEA/ε erkannt.

Aufgabe

- Welche Sprache erkennt der folgende NEA/ε?



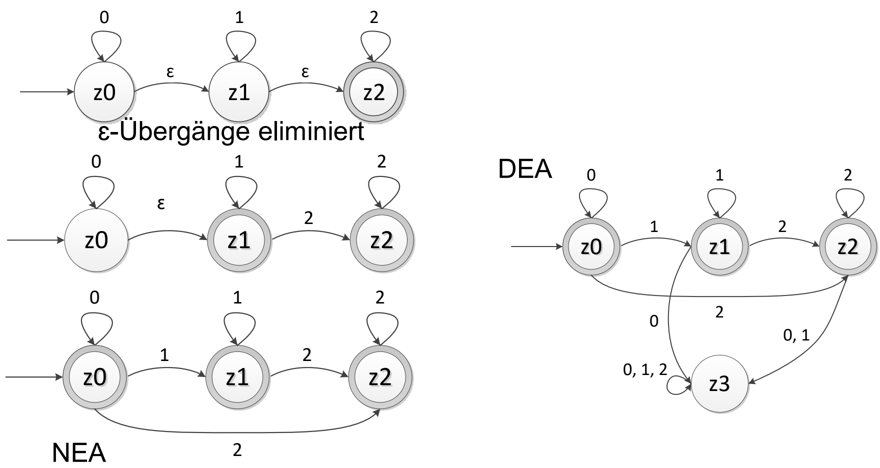
$L_{NEA/\varepsilon}$ = {0^n1 | n ≥ 0} ∪ {1^m0 | m ≥ 0}
= {v^n w | n ≥ 0; v, w ∈ {0, 1}, v ≠ w}

Automaten mit ε-Übergängen

- Gegeben ist ein NEA/ε $A = (Z, \sum, \delta, z_0, E)$.
  Bemerke: $A = (\sum, Z, \delta, z_0, E)$ auch möglich
  - $L_{NEA/\varepsilon}(A) = \{\omega \mid \omega \in \sum^* \text{ und } (z_0, \omega) \rightarrow^* (z, \varepsilon) \text{ und } z \in E\}$
  - ->* transitive Hülle von
  - $\rightarrow_a$ bzw. $\rightarrow_\varepsilon$

Aufgabe

- Erkennt die Klasse der NEA/ε‘s „mehr“ Sprachen als die Klasse der DEA/ NEAs?



akzeptierte Sprache: $\{0^l\, 1^m\, 2^n \mid l, m, n \in \mathbb{N}\}$

Zusammenfassung

Sprache L (regulär) ↔ DEA A : L(A) = L

-> minimaler Automat $\bar{A}$ mit $L(A) = L(\bar{A})$

NEA A‘ : L_{NEA} (A‘) = L

NEA/ε A“ : $L_{NEA/\varepsilon}$ (A“) = L

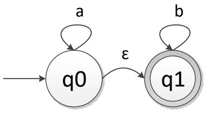
Von oben nach unten wird die Beschreibung des Automatens „komfortabler“.

Die Menge der akzeptierten Sprachen bleibt gleich (nur reguläre Sprachen).

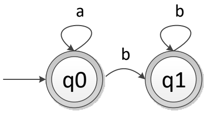
Beispiel

$L = \{a^n b^m \mid n, m \geq 0\}$

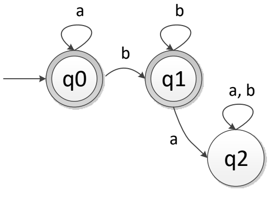
NEA/ε



NEA

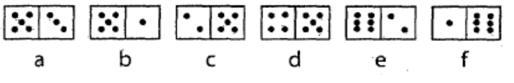


DEA

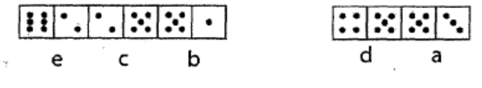


Aufgabe

Gegeben sind die folgenden Dominosteine:



Das Dominospiel wird nach den bekannten Regeln gespielt, das heißt es können beispielsweise folgende Ketten gebildet werden:



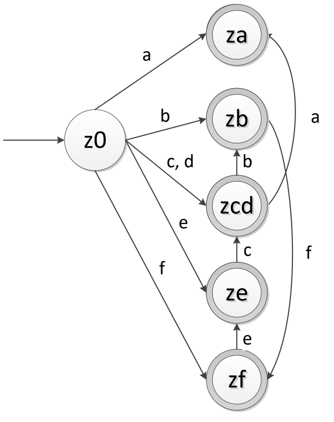
Hierbei gilt:

- Die Steine dürfen dabei nicht gedreht werden.
- Von jeder Sorte $(a, b, \ldots)$ sind unendlich viele Steine vorhanden.

Konstruieren Sie einen NEA (Automatentabelle oder Automatengraph) über dem Alphabet $\Sigma = \{a, b, c, d, e, f\}$, der diejenigen Wörter erkennt, die "gültigen" Dominoketten entsprechen. Beispiel: da oder ecb.

Lösung

- Jedem Stein a, …, f wird ein Zustand zugeordnet: z_a, z_b, …, z_f
- Startzustand z_0.



Yes we can…