TU WIEN Informatics

Endliche Automaten I

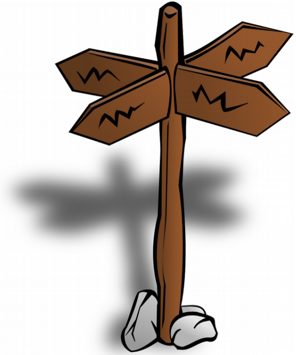
Grundzüge digitaler Systeme (192.134)

Vortrag von: Wolfgang Dvořák

Fakultät für Informatik
TU Wien
E-Mail: gds@list.tuwien.ac.at

Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung



2. Weitere Typen von Automaten

3. Umsetzung von Automaten als Schaltwerk

4. Modellierung

Gone Maggie gone

„The Simpsons“, Staffel 20, Folge 13
Homer will mit Maggie, dem Hund Knecht Ruprecht und einem Glas mit Giftpillen auf die andere Seite des Flusses.

The Simpsons – Modellierung als Automat

Systemkomponenten: Maggie (M), Knecht Ruprecht (K), Gift (G), Homer+Boot (H), linkes/rechtes Flussufer

Situationsbeschreibung (Systemzustand): $\frac{\text{Lebewesen/Dinge links}}{\text{Lebewesen/Dinge rechts}}$

(Wer/Was befindet sich momentan auf welcher Seite des Flusses?)

Anfangszustand: $\frac{MKGH}{}$

Endzustand (Ziel): $\frac{}{MKGH}$

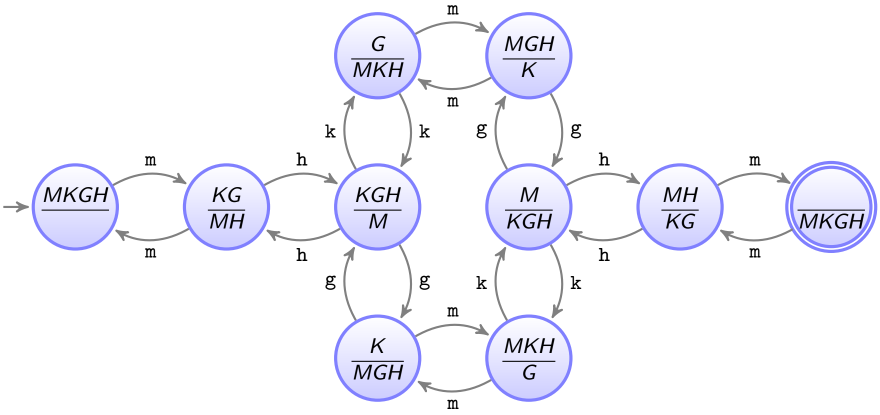
Verbotene Zustände: $\frac{GH}{MK}$, $\frac{MK}{GH}$, $\frac{KH}{MG}$, $\frac{MG}{KH}$, $\frac{H}{MKG}$, $\frac{MKG}{H}$

Zustandsübergänge:

`h`, `m`, `k`, `g` ... Homer fährt alleine/mit Maggie/KR/Gift über den Fluss.

The Simpsons – Modellierung als Automat

Automat (ohne verbotene Zustände und Übergänge):



Mögliche Lösungen: $\{\texttt{mhkmghm}, \texttt{mmmhhhgmkhm}, \dots, \texttt{mhkmgkmgkmghm}, \dots\}$

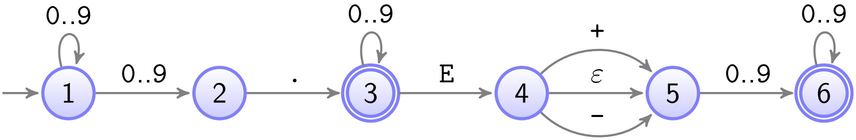
„Sprache des Automaten“

Beispiel: Reelle Numerale mit Exponentialteil

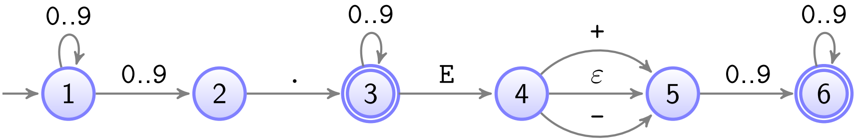
Z.B. `3.14`, `0.314E1` $(= 0.314 \cdot 10^1)$, `314.E-2` $(= 314 \cdot 10^{-2})$

- Mindestens eine Ziffer vor Dezimalpunkt
- Dezimalpunkt
- Nachkommastellen optional
- Exponentialteil optional:
  - eingeleitet durch `E`
  - Vorzeichen optional
  - mindestens eine Ziffer

Endlicher Automat für die reellen Numerale



`0..9` ... Abkürzung für 10 parallele Übergänge beschriftet mit `0` bis `9`.
ε ... Leerwort, „Nichts“

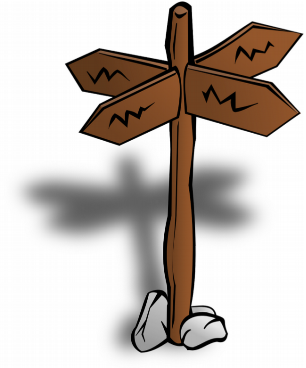


- Zustandsbeschriftungen 1–6 dienen nur der Bezugnahme, irrelevant für das Verhalten des Automaten (sinnvolle Bezeichnungen helfen aber dem Betrachter)
- Kanten sind mit Symbolen beschriftet, die gelesen/geschrieben werden.
- Anfangszustand (1) ist durch einen Pfeil aus dem Nichts markiert.
- Endzustände (3, 6) sind durch einen Doppelkreis markiert.

Zwei Sichtweisen:

- *Akzeptor:* Der Automat *liest* Symbole und akzeptiert alle Zeichenketten, die vom Anfangs- zu einem der Endzustände führen.
- *Generator:* Der Automat *schreibt* Symbole und generiert jene Zeichenketten, die vom Anfangs- zu einem der Endzustände führen.

Endliche Automaten – Übersicht



Endliche Automaten modellieren Systeme bzw. Abläufe, die nur eine begrenzte, feste Zahl an unterscheidbaren Zuständen besitzen.

Kennzeichen:

- endliche Menge von *Zuständen*
- *Übergänge* zwischen Zuständen
- *Eingaben*, die die Übergänge steuern.
- *Ausgaben* oder Aktionen, die in den Zuständen oder während der Übergänge getätigt werden.
- *Anfangszustand*
- *Endzustände* (optional)
- *deterministisch:* Der momentane Zustand und die nächste Eingabe bestimmen eindeutig den Folgezustand.
  nichtdeterministisch: Es gibt Zustände, die bei manchen Eingaben mehrere mögliche Folgezustände besitzen.

Arten endlicher Automaten

(Klassischer) Endlicher Automat:

- Anfangs- und Endzustände
- nur Eingaben (bzw. nur Ausgaben)
- Ein-/Ausgaben verknüpft mit Zustandsübergängen
- verarbeitet endliche Symbolfolgen
- Unterarten: deterministisch, nichtdeterministisch mit/ohne ε-Übergängen

Transducer: wie endlicher Automat, aber mit Ein- *und* Ausgaben.

- Mealy-Automat: deterministischer Transducer,
  Ausgabe hängt von Zustand und Eingabe ab
- Moore-Automat: deterministischer Transducer,
  die Ausgaben sind mit den Zuständen verknüpft.

Büchi-Automat: wie endlicher Automat, verarbeitet aber unendliche Symbolfolgen

Weitere Typen: Verallgemeinerter endlicher Automat, Muller-Automat, Rabin-Automat, Baumautomaten, ...

Englische Begriffe: automaton/automata, finite state machine, DFA (Deterministic Finite Automaton), NFA (Non-Determinstic FA)

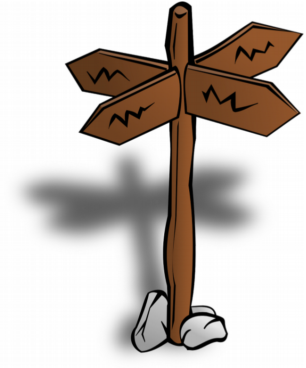
Weitere (nicht-endliche) Automatenarten: Kellerautomaten (Push-down automata), Turing-Maschinen, Registermaschinen etc. können Ausgaben wieder lesen ⇒ zusätzlicher Speicher, mächtiger als endliche Automaten.

Spezifikation von Automaten:

- Graphisch: Zustände sind Knoten, Übergänge sind Kanten, Ein- und Ausgaben sind Beschriftungen von Knoten und Kanten.
- Tabellarisch: Zu jedem Zustand und Eingabesymbol gibt es einen Eintrag mit zugehöriger Ausgabe und den Folgezuständen.

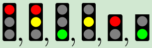
Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung
2. Weitere Typen von Automaten
3. Umsetzung von Automaten als Schaltwerk
4. Modellierung



Formale Sprachen

Alphabet (Σ): endliche, nicht-leere Menge atomarer Symbole

- Menge aller lateinischen Buchstaben, Ziffern und Sonderzeichen
- Menge aller ägyptischen Hieroglyphen
- $\{$  $\}$
- $\{\mathtt{0}, \ldots, \mathtt{9}, \mathtt{.}, \mathtt{E}, \mathtt{+}, \mathtt{-}\}$
- $\{\mathtt{0}, \mathtt{1}\}$
- $\{\mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{11}\}$

Wort über Σ: (endliche) Folge von Zeichen aus dem Alphabet Σ

ε ... Leerwort
$\Sigma^+ = \{\, s_1 \cdots s_n \mid n \in \mathbb{N} \text{ und für } 1 \le i \le n \text{ gilt } s_i \in \Sigma \,\}$ ... Menge aller nicht-leeren Wörter über Σ
$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$ ... Menge aller Wörter über Σ (inklusive Leerwort)

$w_1 \cdot w_2 = w_1 w_2$ ... Verkettung der Wörter $w_1, w_2 \in \Sigma^*$

$\langle \Sigma^*, \cdot, \varepsilon \rangle$ bildet ein Monoid

D.h.: Für alle Wörter $u, v, w \in \Sigma^*$ gelten folgende Gleichungen:

$(u \cdot v) \cdot w = u \cdot (v \cdot w)$	Assoziativität
$w \cdot \varepsilon = \varepsilon \cdot w = w$	Neutralität

$\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$

$10 \cdot \varepsilon \cdot 11101 \cdot 000 = 1011101000$ (Klammerung irrelevant, Assoziativität!)

$\varepsilon \cdot \varepsilon \cdot \varepsilon = \varepsilon$

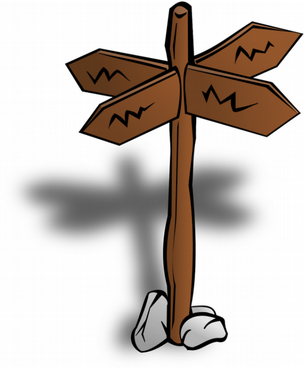
Formale Sprache über Σ: beliebige Teilmenge von Σ^*

- die Menge aller deutschen Sätze (Alphabet: Buchstaben+Satzzeichen)
- die Menge aller Java-Programme (Alphabet: ASCII-Zeichen)
- $\{\}$, $\{\varepsilon\}$, Σ^*

2^{Σ^*} ... Menge aller Sprachen über Σ = Menge aller Teilmengen von Σ^*

Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung
2. Weitere Typen von Automaten
3. Umsetzung von Automaten als Schaltwerk
4. Modellierung



Deterministische endliche Automaten

Deterministischer endlicher Automat (DEA)

... wird beschrieben durch ein 5-Tupel $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, wobei

- Q ... endliche Menge der Zustände
- Σ ... Eingabealphabet (*input alphabet*)
- $\delta \colon Q \times \Sigma \to Q$ ... Übergangsfunktion (total) (*transition function*)
- $q_0 \in Q$ ... Anfangszustand (*initial state*)
- $F \subseteq Q$ ... Menge der Endzustände (*final states*)

δ ist eine totale Funktion: Folgezustand $\delta(q, s)$ ist für jeden Zustand $q \in Q$ und jede Eingabe $s \in \Sigma$ eindeutig definiert. $\Longrightarrow$ „deterministisch“

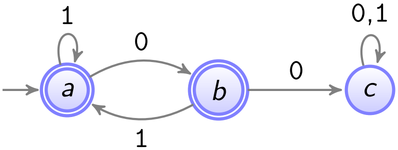
Erweiterte Übergangsfunktion $\delta^* \colon Q \times \Sigma^* \to Q$

$\delta^*(q, \varepsilon) = q, \quad \delta^*(q, sw) = \delta^*(\delta(q, s), w) \quad$ für alle $q \in Q$, $s \in \Sigma$, $w \in \Sigma^*$.

Akzeptierte/Generierte Sprache

$\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^* \mid \delta^*(q_0, w) \in F \,\}$

Beispiel: 00-freie Binärstrings



c . . . „Falle“, Fehlerzustand
wird oft auch weggelassen

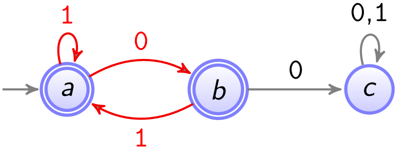
$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, wobei

- $Q = \{a, b, c\}$ . . . Zustandsmenge
- $\Sigma = \{0, 1\}$ . . . Eingabealphabet
- $\delta\colon Q \times \Sigma \to Q$ . . . Übergangsfunktion definiert durch:

δ	0	1
a	b	a
b	c	a
c	c	c

- $q_0 = a$ . . . Anfangszustand
- $F = \{a, b\}$ . . . Endzustände

Beispiel: 00-freie Binärstrings



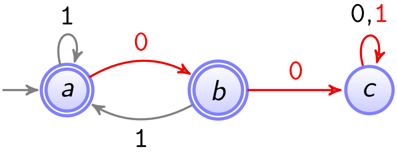
$$\begin{aligned}\delta^*(a, 101) &= \delta^*(\delta(a, 1), 01) \\ &= \delta^*(a, 01) \\ &= \delta^*(\delta(a, 0), 1) \\ &= \delta^*(b, 1) \\ &= \delta^*(\delta(b, 1), \varepsilon) \\ &= \delta^*(a, \varepsilon) \\ &= a\end{aligned}$$

$$\delta^*(q, sw) = \delta^*(\delta(q, s), w)$$

$$\delta^*(q, \varepsilon) = q$$

Das Wort 101 wird von $\mathcal{A}$ akzeptiert/generiert, d.h., $101 \in \mathcal{L}(\mathcal{A})$, weil $\delta^*(a, 101) = a$ ein Endzustand ist.

Beispiel: 00-freie Binärstrings

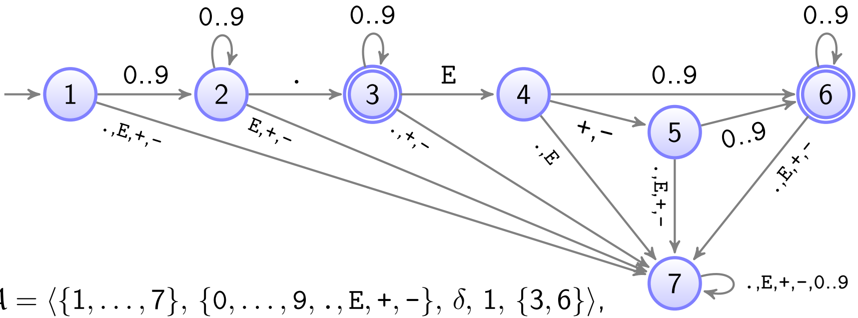


$$\delta^*(a, 001) = \delta^*(\delta(a, 0), 01) \qquad \delta^*(q, sw) = \delta^*(\delta(q, s), w)$$
$$= \delta^*(b, 01)$$
$$= \delta^*(\delta(b, 0), 1)$$
$$= \delta^*(c, 1)$$
$$= \delta^*(\delta(c, 1), \varepsilon)$$
$$= \delta^*(c, \varepsilon) \qquad \delta^*(q, \varepsilon) = q$$
$$= c$$

Das Wort 001 wird von $\mathcal{A}$ nicht akzeptiert/generiert, $001 \notin \mathcal{L}(\mathcal{A})$,
weil $\delta^*(a, 001) = c$ kein Endzustand ist.

$\mathcal{L}(\mathcal{A}) = \{\, w \in \{0, 1\}^* \mid 00 \text{ kommt nicht in } w \text{ vor} \,\}$

Beispiel: reelle Numerale

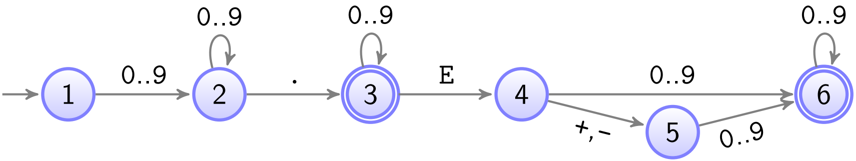


$\mathcal{A} = \langle\{1,\ldots,7\},\ \{0,\ldots,9,\ .,\ \mathtt{E},\ +,\ -\},\ \delta,\ 1,\ \{3,6\}\rangle$,

wobei

δ	$0 \cdots 9$	.	E	+	−
1	$2 \cdots 2$	7	7	7	7
2	$2 \cdots 2$	3	7	7	7
3	$3 \cdots 3$	7	4	7	7
4	$6 \cdots 6$	7	7	5	5
5	$6 \cdots 6$	7	7	7	7
6	$6 \cdots 6$	7	7	7	7
7	$7 \cdots 7$	7	7	7	7

Beispiel: reelle Numerale



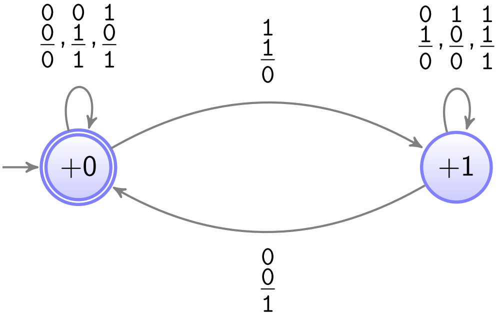
	δ	$0 \cdots 9$	.	E	+	−
AZ	1	$2 \cdots 2$	7	7	7	7
	2	$2 \cdots 2$	3	7	7	7
EZ	3	$3 \cdots 3$	7	4	7	7
	4	$6 \cdots 6$	7	7	5	5
	5	$6 \cdots 6$	7	7	7	7
EZ	6	$6 \cdots 6$	7	7	7	7
	7	$7 \cdots 7$	7	7	7	7

Beispiel: Binäraddition von rechts nach links (Kontrolle)

$$\begin{array}{cccccccr} 0 & 0 & 1 & 0 & 1 & 1 & = & 11_{10} \\ 0 & 0 & 0 & 1 & 0 & 1 & = & 5_{10} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & = & 16_{10} \end{array}$$

$$\longleftarrow$$

$Q = \{+0, +1\}$ $\qquad \Sigma = \{\frac{\substack{0\\0}}{0}, \frac{\substack{0\\0}}{1}, \frac{\substack{0\\1}}{0}, \frac{\substack{0\\1}}{1}, \frac{\substack{1\\0}}{0}, \frac{\substack{1\\0}}{1}, \frac{\substack{1\\1}}{0}, \frac{\substack{1\\1}}{1}\}$



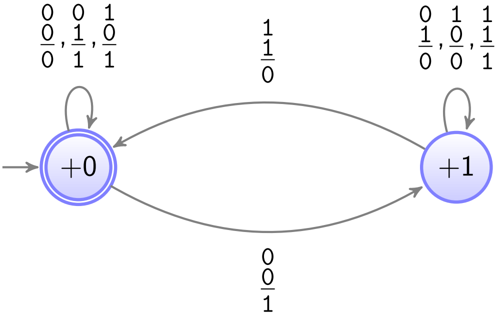
$+0$ . . . kein Übertrag
$+1$ . . . Übertrag

Beispiel: Binäraddition von links nach rechts (Kontrolle)

$$\begin{array}{cccccccl} & 0 & 0 & 1 & 0 & 1 & 1 & = 11_{10} \\ & 0 & 0 & 0 & 1 & 0 & 1 & = 5_{10} \\ \hline & 0 & 1 & 0 & 0 & 0 & 0 & = 16_{10} \end{array}$$

$\longrightarrow$

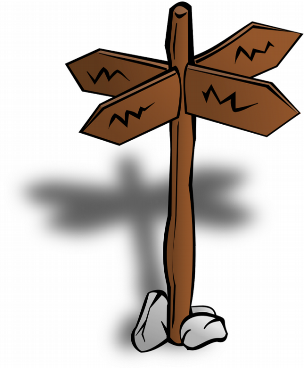
$Q = \{+0, +1\}$ $\qquad$ $\Sigma = \{ \frac{\overset{0}{0}}{0}, \frac{\overset{0}{0}}{1}, \frac{\overset{0}{1}}{0}, \frac{\overset{0}{1}}{1}, \frac{\overset{1}{0}}{0}, \frac{\overset{1}{0}}{1}, \frac{\overset{1}{1}}{0}, \frac{\overset{1}{1}}{1} \}$

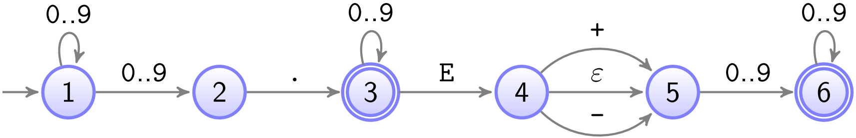


$+0$ . . . kein Übertrag
$+1$ . . . Übertrag

Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung
2. Weitere Typen von Automaten
3. Umsetzung von Automaten als Schaltwerk
4. Modellierung





Kein deterministischer Automat!

- $\delta(1, \mathtt{0}) = 1$?
  $\delta(1, \mathtt{0}) = 2$?
  Die Übergangsfunktion muss ein eindeutiges Ergebnis besitzen.
- $\delta(4, \varepsilon) = 5$?
  Die Übergangsfunktion ist vom Typ $Q \times \Sigma \to Q$, aber $\varepsilon \notin \Sigma$!

Indeterminismus: Der momentane Zustand und das Eingabesymbol legen den nächsten Zustand bzw. die nächste Aktion nicht eindeutig fest.

- In Zustand 1 sind bei Eingabe `0` die Folgezustände 1 und 2 möglich.
- In Zustand 3 sind bei Eingabe `E` die Folgezustände 4 und 5 möglich.
- In Zustand 4 ist Zustand 5 mit und ohne Eingabe erreichbar.

Ob die richtige Entscheidung getroffen wurde, wird erst später klar.

Nichtdeterministische endliche Automaten

Nichtdeterministischer endlicher Automat (NEA)

… wird beschrieben durch ein 5-Tupel $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, wobei

- Q, Σ, q_0, F … siehe DEAs
- $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ … Übergangsrelation

vgl. DEA: $\delta \colon Q \times \Sigma \to Q$ … totale Übergangsfunktion

Erweiterte Übergangsrelation $\delta^* \subseteq Q \times \Sigma^* \times Q$

δ^* ist die kleinste Menge mit folgenden Eigenschaften:

- $(q, \varepsilon, q) \in \delta^*$ für alle $q \in Q$
- Wenn $(q_1, w, q_2) \in \delta^*$ und $(q_2, s, q_3) \in \delta$, dann $(q_1, ws, q_3) \in \delta^*$ $(s \in \Sigma,\ w \in \Sigma^*)$.

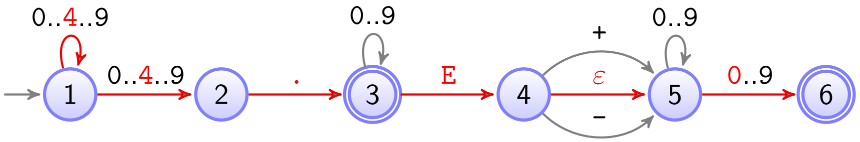
vgl. DEA: $\delta^*(q, \varepsilon) = q, \quad \delta^*(q, sw) = \delta^*(\delta(q, s), w)$

Akzeptierte/Generierte Sprache

$\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^* \mid (q_0, w, q_f) \in \delta^* \text{ für ein } q_f \in F \,\}$

vgl. DEA: $\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^* \mid \delta^*(q_0, w) \in F \,\}$

Beispiel: `42.E0` ist ein reelles Numeral



$\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^* \mid (1, w, 3) \in \delta^* \text{ oder } (1, w, 6) \in \delta^* \,\}$

Zu zeigen: $\texttt{42.E0} \in \mathcal{L}(\mathcal{A})$

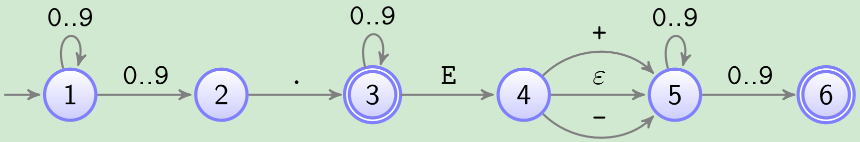
Wenn $(q_1, w, q_2) \in \delta^*$	und $(q_2, s, q_3) \in \delta$,	dann $(q_1, ws, q_3) \in \delta^*$.
$(1, \varepsilon, 1)$	$(1, 4, 2)$	$(1, 4, 2)$
$(1, \varepsilon, 1)$	$(1, 4, 1)$	$(1, 4, 1)$
$(1, 4, 1)$	$(1, 2, 2)$	$(1, 42, 2)$
$(1, 42, 2)$	$(2, \texttt{.}, 3)$	$(1, \texttt{42.}, 3)$
$(1, \texttt{42.}, 3)$	$(3, \texttt{E}, 4)$	$(1, \texttt{42.E}, 4)$
$(1, \texttt{42.E}, 4)$	$(4, \varepsilon, 5)$	$(1, \texttt{42.E}, 5)$
$(1, \texttt{42.E}, 5)$	$(5, 0, 6)$	$(1, \texttt{42.E0}, 6)$

$(1, \texttt{42.E0}, 6) \in \delta^*$, 1 ist Anfangs- und 6 Endzustand $\Longrightarrow \texttt{42.E0} \in \mathcal{L}(\mathcal{A})$

Tabellarische Darstellung der Übergangsrelation $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$

Für jeden Zustand $q \in Q$ und jede Eingabe $s \in \Sigma \cup \{\varepsilon\}$:

Tabelleneintrag mit der *Menge* $\{\, q' \in Q \mid (q, s, q') \in \delta \,\}$ der Folgezustände



	δ	0	$\cdots$	9	.	E	+	-	ε
AZ	1	$\{1,2\}$	$\cdots$	$\{1,2\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
	2	$\{\}$	$\cdots$	$\{\}$	$\{3\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
EZ	3	$\{3\}$	$\cdots$	$\{3\}$	$\{\}$	$\{4\}$	$\{\}$	$\{\}$	$\{\}$
	4	$\{\}$	$\cdots$	$\{\}$	$\{\}$	$\{\}$	$\{5\}$	$\{5\}$	$\{5\}$
	5	$\{5,6\}$	$\cdots$	$\{5,6\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
EZ	6	$\{\}$	$\cdots$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$

Alternative Definition von NEAs:

Übergangs*funktion* $\delta \colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ (ist total!) an Stelle von

Übergangs*relation* $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$

Vergleich DEA – NEA

NEAs sind flexibler:

- Mehrere Folgezustände pro Zustand und Eingabe möglich;
- Kein Folgezustand zu einem Zustand und einer Eingabe erlaubt;
- Zustandswechsel ohne Eingabe möglich (ε-Übergang).

DEAs und NEAs besitzen dieselbe Ausdrucksstärke.

- Jeder DEA ist per Definition auch ein NEA.
- Zu jedem NEA gibt es einen DEA, der dieselbe Sprache akzeptiert.
  (Lässt sich automatisch finden, siehe später.)

Vorteile von NEAs:

- Benötigen teilweise erheblich weniger Zustände und Übergänge als äquivalente DEAs.
  Die Zustandszahl im DEA kann exponentiell größer sein als im NEA.
- Bei Modellierungsaufgaben leichter zu konstruieren.

Vorteile von DEAs:

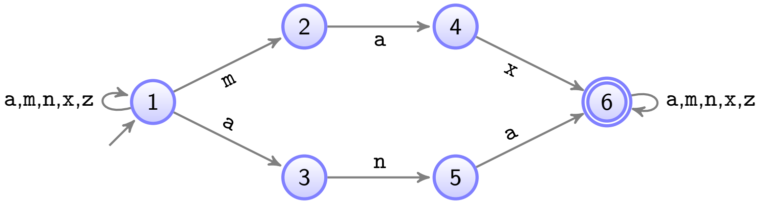
- Effiziente Abarbeitung, kein Backtracking.

Beispiel: Suche nach Max und Ana

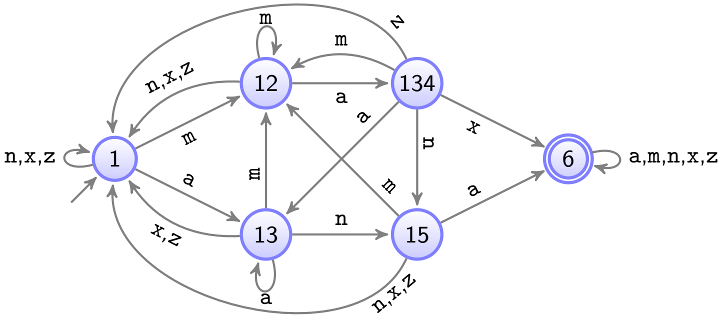
Gesucht: Automat zur Suche nach „max“ und „ana“ in einem Text

$\Sigma = \{\mathtt{a}, \mathtt{m}, \mathtt{n}, \mathtt{x}, \mathtt{z}\}$ (z . . . Stellvertreter für b–l, o–w, y, z, . . .)

NEA:

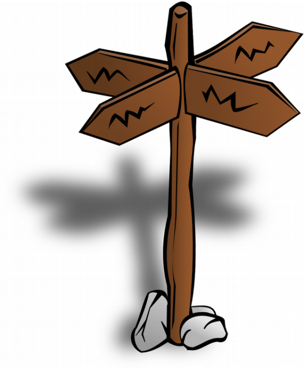


DEA:

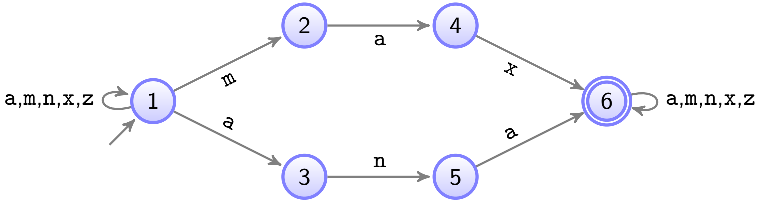


Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung
2. Weitere Typen von Automaten
3. Umsetzung von Automaten als Schaltwerk
4. Modellierung



Beispiel: Suche nach Max und Ana

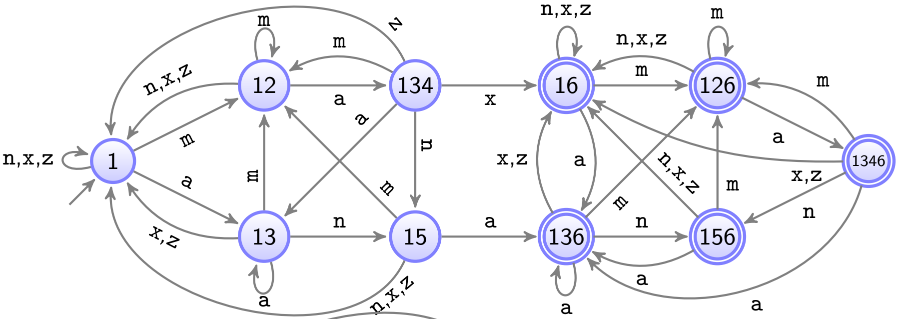


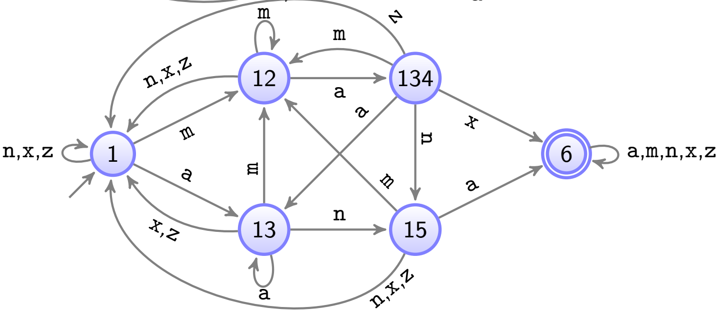
	δ^*	a	m	n	x	z
SZ	1	$\{1,3\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	$\{1\}$
	2	$\{4\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
	3	$\{\}$	$\{\}$	$\{5\}$	$\{\}$	$\{\}$
	4	$\{\}$	$\{\}$	$\{\}$	$\{6\}$	$\{\}$
	5	$\{6\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
EZ	6	$\{6\}$	$\{6\}$	$\{6\}$	$\{6\}$	$\{6\}$

Identisch mit der Tabelle für δ, wenn es keine ε-Kanten gibt.

	δ^*	a	m	n	x	z
SZ	1	$\{1,3\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	$\{1\}$
	2	$\{4\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
	3	$\{\}$	$\{\}$	$\{5\}$	$\{\}$	$\{\}$
	4	$\{\}$	$\{\}$	$\{\}$	$\{6\}$	$\{\}$
	5	$\{6\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
EZ	6	$\{6\}$	$\{6\}$	$\{6\}$	$\{6\}$	$\{6\}$

	$\widehat{\delta}$	a	m	n	x	z
SZ	$\{1\}$	$\{1,3\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	$\{1\}$
	$\{1,2\}$	$\{1,3,4\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	$\{1\}$
	$\{1,3\}$	$\{1,3\}$	$\{1,2\}$	$\{1,5\}$	$\{1\}$	$\{1\}$
	$\{1,3,4\}$	$\{1,3\}$	$\{1,2\}$	$\{1,5\}$	$\{1,6\}$	$\{1\}$
	$\{1,5\}$	$\{1,3,6\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	$\{1\}$
EZ	$\{1,6\}$	$\{1,3,6\}$	$\{1,2,6\}$	$\{1,6\}$	$\{1,6\}$	$\{1,6\}$
EZ	$\{1,3,6\}$	$\{1,3,6\}$	$\{1,2,6\}$	$\{1,5,6\}$	$\{1,6\}$	$\{1,6\}$
EZ	$\{1,2,6\}$	$\{1,3,4,6\}$	$\{1,2,6\}$	$\{1,6\}$	$\{1,6\}$	$\{1,6\}$
EZ	$\{1,5,6\}$	$\{1,3,6\}$	$\{1,2,6\}$	$\{1,6\}$	$\{1,6\}$	$\{1,6\}$
EZ	$\{1,3,4,6\}$	$\{1,3,6\}$	$\{1,2,6\}$	$\{1,5,6\}$	$\{1,6\}$	$\{1,6\}$


1
12
13
134
15
16
126
136
156
1346
n,x,z
m
a
x,z
n
x


1
12
13
134
15
6
n,x,z
a,m,n,x,z
m
a
x,z
n
x

Determinisierung

Gegeben: NEA $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ mit $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$

Gesucht: DEA $\widehat{\mathcal{A}} = \langle \widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q_0}, \widehat{F} \rangle$ mit $\widehat{\delta}\colon \widehat{Q} \times \Sigma \to \widehat{Q}$,
sodass $\mathcal{A}$ und $\widehat{\mathcal{A}}$ dieselbe Sprache akzeptieren.

Wir definieren den deterministischen Automaten $\widehat{\mathcal{A}}$ folgendermaßen:

- $\widehat{Q} = 2^Q$
- $\widehat{q_0} = \{q_0\}$
- $\widehat{F} = \begin{cases} \{\, \widehat{q} \in \widehat{Q} \mid \widehat{q} \cap F \neq \emptyset \,\} \cup \{\widehat{q_0}\} & \text{falls } \varepsilon \in \mathcal{L}(\mathcal{A}) \\ \{\, \widehat{q} \in \widehat{Q} \mid \widehat{q} \cap F \neq \emptyset \,\} & \text{sonst} \end{cases}$
- Für alle Zustände $\widehat{q} \in \widehat{Q}$ und alle Symbole $s \in \Sigma$:

  $\widehat{\delta}(\widehat{q}, s) = \{\, q' \in Q \mid \text{es gibt } q \in \widehat{q}, \text{ sodass } (q, s, q') \in \delta^* \,\}$

$\mathcal{A}$ und $\widehat{\mathcal{A}}$ sind äquivalent, d.h., sie akzeptieren dieselbe Sprache: $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\widehat{\mathcal{A}})$.

Determinisierung

Anmerkungen

Neue Endzustände: Ein neuer Zustand $\widehat{q}$ ist Endzustand, ...

- wenn seine Bezeichnung einen der alten Endzustände enthält,
  d.h., wenn $\widehat{q} \cap F \neq \emptyset$, oder
- wenn es sich um den neuen Startzustand handelt und der alte Automat das Leerwort akzeptiert,
  d.h., wenn $\widehat{q} = \widehat{q_0}$ und $\varepsilon \in \mathcal{L}(\mathcal{A})$,
  d.h., wenn $\widehat{q} = \widehat{q_0}$ und $(q_0, \varepsilon, f) \in \delta^*$ für einen Endzustand $f \in F$.

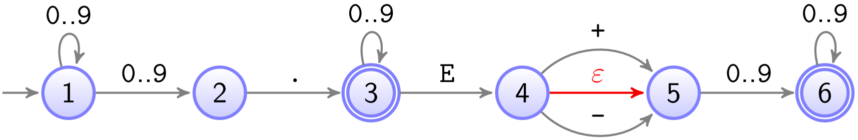
Neue Übergangsfunktion: Wegen

$$\widehat{\delta}(\widehat{q}, s) = \bigcup_{q \in \widehat{q}} \{\, q' \in Q \mid (q, s, q') \in \delta^* \,\} = \bigcup_{q \in \widehat{q}} \delta^*(q, s)$$

spart es Arbeit, wenn man zuerst $\delta^*(q, s)$ für alle $q \in Q$ und alle $s \in \Sigma$ berechnet. Danach müssen nur mehr die Zeilen, die $\widehat{q}$ entsprechen, vereinigt werden.
Neue Zustände: Betrachte nur jene Zustände aus 2^Q, die von $\widehat{q_0}$ aus erreichbar sind.

Determinisierung

Beispiel: Reelle Numerale mit Exponentialteil



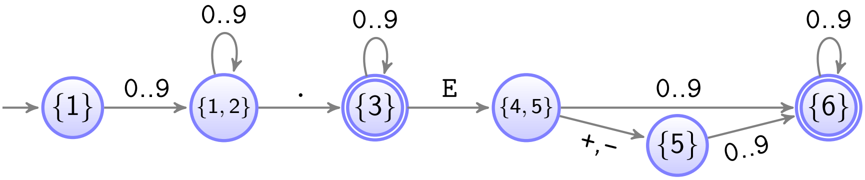
	δ^*	0..9	.	E	+, -
SZ	1	{1, 2}	{}	{}	{}
	2	{}	{3}	{}	{}
EZ	3	{3}	{}	{4, 5}	{}
	4	{6}	{}	{}	{5}
	5	{6}	{}	{}	{}
EZ	6	{6}	{}	{}	{}

Determinisierung

Beispiel: Reelle Numerale mit Exponentialteil

	δ^*	0..9	.	E	+, -
SZ	1	{1, 2}	{}	{}	{}
	2	{}	{3}	{}	{}
EZ	3	{3}	{}	{4, 5}	{}
	4	{6}	{}	{}	{5}
	5	{6}	{}	{}	{}
EZ	6	{6}	{}	{}	{}

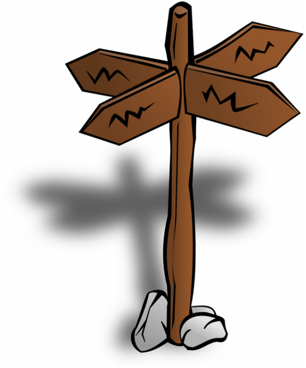
	$\widehat{\delta}$	0..9	.	E	+, -
SZ	{1}	{1, 2}	{}	{}	{}
	{1, 2}	{1, 2}	{3}	{}	{}
	{}	{}	{}	{}	{}
EZ	{3}	{3}	{}	{4, 5}	{}
	{4, 5}	{6}	{}	{}	{5}
	{5}	{6}	{}	{}	{}
EZ	{6}	{6}	{}	{}	{}



In der graphischen Darstellung lassen wir der Übersichtlichkeit wegen die Falle {} samt aller Übergänge dorthin weg.

Endliche Automaten – Übersicht

1. Klassische Endliche Automaten
   - Beispiel: Gone Maggie gone (revisited)
   - Klassifikation von Automaten
   - Grundlagen formaler Sprachen
   - Deterministische endliche Automaten
   - Nichtdeterministische endliche Automaten
   - Determinisierung
   - Zusammenfassung
2. Weitere Typen von Automaten
3. Umsetzung von Automaten als Schaltwerk
4. Modellierung



Zusammenfassung

- Endliche Automaten zur Modellierung von Abläufen
- Grundlagen formaler Sprachen
- Automaten als Akzeptor von formalen Sprachen
  - Deterministische endliche Automaten (DEA)
  - Nichtdeterministische endliche Automaten (NEA)
  - Determinisierung von NEAs