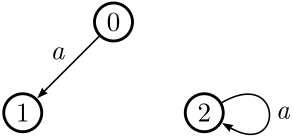
Inhalt

Zustandsdiagramm

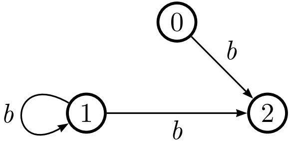
$\delta\colon \Sigma \to (Q \to Q \to \Omega)$

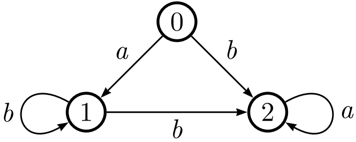
Beispiel

δ_a



δ_b





δ

Automat

Definition

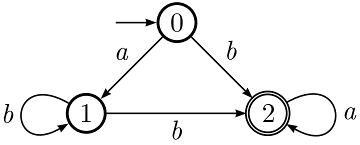
Ein nicht-deterministischer, endlicher Automat besteht aus

- einer endlichen Menge Σ, dem Eingabealphabet;
- einer endlichen Menge Q von Zuständen (states);
- einer entscheidbaren Zustandsüberführungsrelation $\delta\colon \Sigma \to (Q \to Q \to \mathbb{B})$;
- einer Teilmenge $I \subseteq Q$ von Initialzuständen;
- einer Teilmenge $F \subseteq Q$ von Finalzuständen.

Diese Konstruktion wird dann auch mit dem Quintupel $(\Sigma, Q, \delta, I, F)$ bezeichnet.

Erkannte Wörter

Der Automat



erkennt die Wörter: b, ab, $abbbb$, $baaaa$, $abbbbbaaaaa$;
nicht aber: aa $abab$, bb.

Sprache eines Automaten

$(Q \to Q \to \Omega)$ bildet mit dem Relationenprodukt ein Monoid.

$\delta \colon \Sigma \to (Q \to Q \to \Omega)$
wird fortgesetzt zu einem Homomorphismus:
$\delta^* \colon \Sigma^* \to (Q \to Q \to \Omega)$

Definition

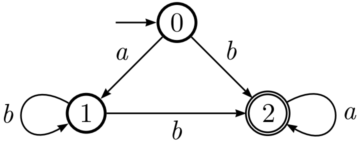
Die Sprache eines Automaten

$$L(\mathscr{A}) := \{\, u \colon \Sigma^* \mid \underset{i\colon I}{\exists}\ \underset{j\colon J}{\exists}\ i \xrightarrow{\delta_u^*} j \,\}$$

besteht aus allen Wörtern, die einen Übergang von einem Initialzustand in einen Finalzustand erlauben.
Man sagt auch, der Automat erkennt oder akzeptiert die Sprache $L(\mathscr{A})$, bzw. jedes Wort in dieser Sprache.

Beispiel

Der Automat



akzeptiert die Sprache, die durch den folgenden regulären Ausdruck

$$(b \mid ab^* b)a^*$$

beschrieben wird.

Dieser Automat arbeitet nicht-deterministisch.

Automat

Definition

Ein deterministischer, endlicher Automat besteht aus

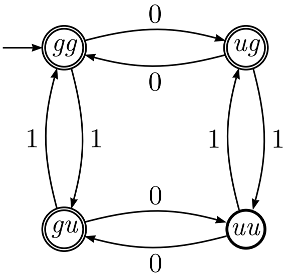
- einer endlichen Menge Q von Zuständen;
- einer endlichen Menge Σ, dem Eingabealphabet;
- einer Zustandsüberführungs**funktion** $\delta\colon \Sigma \to (Q \to Q)$;
- einem Initialzustand $q_0 : Q$;
- einer Teilmenge $F \subseteq Q$ von Finalzuständen.

Diese Konstruktion wird dann auch mit dem Quintupel $(\Sigma, Q, \delta, q_0, F)$ bezeichnet.

Deterministischer Automat

Example

Der Automat



ist deterministisch und akzeptiert die Sprache bestehend allen 0-1-Folgen, welche eine gerade Anzahl von Nullern oder Einsern aufweisen.

Partieller Automat

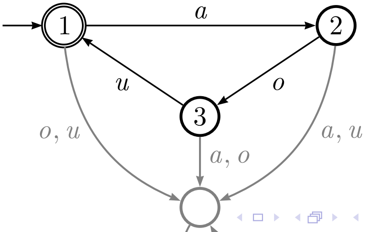
Definition

Ein partieller (deterministischer) Automat hat zu jedem Input höchstens einen Folgezustand.

Ein partieller Automat kann leicht vervollständigt werden, indem man einen neuen Zustand ω (weder initial noch final), der immer dann als Folgezustand verwendet wird, wenn sonst keiner zur Verfügung steht.

Beispiel

Ein patieller deterministischer Automat über dem Alphabet $\{ a, o, u \}$:



Deterministisch äquivalent zu Nicht-deterministisch

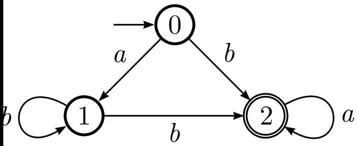
Satz

Zu jedem nicht-deterministischen Automaten gibt es einen deterministischen Automaten, der dieselbe Sprache erkennt.

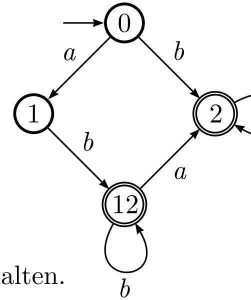
Beweis.

Man betrachte die Teilmengen von Q als Zustände:
$\bar{\delta}\colon \Sigma \to \mathscr{P}(Q) \to \mathscr{P}(Q)$ $\square$

Beispiel



$\bar{\delta}$	a	b
0	1	2
1		12
2	2	
12	2	12



Initialzustand: I.
Finalzustände: Alle, die ein Element von J enthalten.
Betrachte nur erreichbare Zustandsmengen.

Verbindung von Automaten

Definition

Seien $\mathscr{A}_1, \mathscr{A}_2$ zwei Automaten über demselben Eingabealphabet.

Parallelschaltung : $\mathscr{A}_1 + \mathscr{A}_2$

Serienschaltung : $\mathscr{A}_1\mathscr{A}_2$

Rückkoppelung : $\mathscr{A}^*$

Satz

Seinen $\mathscr{A}_1, \mathscr{A}_2$ endliche Automaten. Dann gilt

$$L(\mathscr{A}_1\mathscr{A}_2) = L(\mathscr{A}_1)L(\mathscr{A}_2)$$
$$L(\mathscr{A}_1 + \mathscr{A}_2) = L(\mathscr{A}_1) \cup L(\mathscr{A}_2)$$
$$L(\mathscr{A}_1^*) = L(\mathscr{A}_1)^*.$$

Folgerung

Zu jeder regulären Sprache gibt es einen erkennenden Automaten.

Vom Automaten zum regulären Ausdruck

Satz

Jede von einen endlichen Automaten erkannte Sprache ist regulär.

Beweis.

$R_{i,j}^{Y}$ bestehe aus all jenen Wörtern, die vom Zustand i in den Zustand j führen, und dabei *zwischendurch* nur in Zustände aus der Menge Y führen. Dann gilt

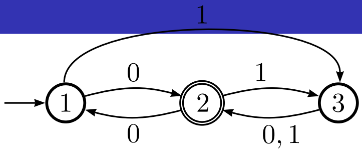
$$L(\mathscr{A}) = \bigcup_{j \in F} R_{0,j}^{Q}.$$

Es reicht also zu zeigen, daß jedes $R_{i,j}^{Y}$ regulär ist.
Rekursion:

$$R_{i,j}^{Y \cup \{ k \}} = R_{i,j}^{Y} \cup R_{i,k}^{Y} (R_{k,k}^{Y})^{*} R_{k,j}^{Y},$$

Und die $R_{i,j}^{\varnothing}$ sind endliche Teilmengen von Σ. $\square$

Beispiel



$$
\begin{aligned}
r_{1,2}^{\{1,2,3\}} &= r_{1,2}^{\{1,2\}} \mid r_{1,3}^{\{1,2\}}\left(r_{3,3}^{\{1,2\}}\right)^* r_{3,2}^{\{1,2\}} \\
r_{1,2}^{\{1,2\}} &= r_{1,2}^{\{1\}} \mid r_{1,2}^{\{1\}}\left(r_{2,2}^{\{1\}}\right)^* r_{2,2}^{\{1\}} \\
&= r_{1,2}^{\{1\}}\left(r_{2,2}^{\{1\}}\right)^* \\
&= 0(\epsilon \mid 00)^* = 0(00)^* \\
r_{1,3}^{\{1,2\}} &= r_{1,3}^{\{1\}} \mid r_{1,2}^{\{1\}}\left(r_{2,2}^{\{1\}}\right)^* r_{2,3}^{\{1\}} \\
&= 1 \mid 0(00)^*(1 \mid 01) = 0^*1 \\
r_{3,3}^{\{1,2\}} &= r_{3,3}^{\{1\}} \mid r_{3,2}^{\{1\}}\left(r_{2,2}^{\{1\}}\right)^* r_{2,3}^{\{1\}} \\
&= \epsilon \mid (0 \mid 1)(00)^*(1 \mid 01) = \epsilon \mid (0 \mid 1)0^*1 \\
r_{3,2}^{\{1,2\}} &= r_{3,2}^{\{1\}}\left(r_{2,2}^{\{1\}}\right)^* \\
&= (0 \mid 1)(00)^* \\
r_{1,2}^{\{1,2,3\}} &= 0(00)^* \mid 0^*1((0 \mid 1)0^*1)^*(0 \mid 1)(00)^*
\end{aligned}
$$

Abschlußeigenschaften regulärer Sprachen

Satz

Die Klasse der regulären Sprachen ist gegenüber allen boolschen Mengenoperationen (also auch Durchschnitt und Komplement)

Beweis.

Es ist nicht offensichtlich, wie ein regulärer Ausdruck für das Komplement einer regulärer Sprache ausschauen soll. Dafür kann man zu jedem Automaten $\mathscr{A}$ ganz einfach jenen konstruieren, der alle Wörter erkennt, die $\mathscr{A}$ nicht erkennt: man verwendet einfach $Q \setminus F$ als Finalzustände. Für den Durchschnitt kann man dann etwa ein De Morgan Gesetz verwenden:

$$L_1 \cap L_2 = \complement(\complement L_1 \cup \complement L_2).$$

$\square$

Ferner ist die Klasse der regulären Sprachen gegenüber Spiegelung, homomorphen Bildern und Substitutionen abgeschlossen.

Für die Spiegelung muß man lediglich den regulären Ausdruck spiegeln.

Minimierung endlicher Automaten

Definition

Ein vollständiger deterministischer endlicher Automat $\mathscr{A}$ heißt minimal wenn jeder andere Automat, der dieselbe Sprache erkennt, mindestens soviel Zustände wie $\mathscr{A}$ hat.

Definition

Äquivalente Zustände: $i \sim j \iff \underset{u\,:\,\Sigma^*}{\forall} \left(\delta^*_u(i) \in F \iff \delta^*_u(j) \in F\right)$

Satz

Zu jeder regulären Sprache gibt es, bis auf Isomorphie, genau einen diese erkennenden minimalen Automaten.

Beweis.

Äquivalente Zustände lassen sich identifizieren.
Zu jedem erreichbaren Zustand k gibt es einen Inputstring $u_k : \Sigma^*$, sodaß $\delta^*_{u_k}(q_0) = k$.
Ist ein $\mathscr{A}'$ ein weiterer Automat, der dieselbe Sprache erkennt, dann entspricht der Zustand $\delta'^*_{u_k}(q'_0)$ in $\mathscr{A}'$ dem Zustand k in $\mathscr{A}$.
Gilt $\delta^*_u(q_0) = \delta^*_v(q_0)$, dann ist $\delta'^*_u(q'_0) \sim \delta'^*_v(q'_0)$. $\square$

Entscheidungsprobleme für reguläre Sprachen

Satz

Alle folgenden Probleme sind für reguläre Sprachen entscheidbar:

1. $L = \varnothing$;
2. L *ist endlich;*
3. $w \in L$;
4. $L_1 = L_2$;
5. $L_1 \subseteq L_2$.

Beweis.

3 Festzustellen, ob ein Wort zu einer Sprache gehört oder nicht, ist gerade die Aufgabe, die ein erkennender Automat (optimal) beherrscht.

4 Man kann von beiden einen minimalen Automaten berechnen und feststellen, ob diese isomorph sind. Alternative: Zurückführung auf Durchschnittsberechnung und Punkt 1.

5 $L_1 \subseteq L_2$ gilt genau dann wenn $L_1 \cup L_2 = L_2$ (oder wenn $L_1 \cap \complement L_2 = \varnothing$).

Vervielfachen von Wortteilen

Pumping Lemma

Sei L eine reguläre Sprache.

- *Dann gibt es $n \in \mathbb{N}$, sodaß*
- *jedes $z \in L$ mit $|z| \geq n$*
- *derart in $z = uvw$ zerlegt werden kann, daß gilt:*
  - $|v| \geq 1$;
  - $|uv| \leq n$;
  - $uv^k w \in L$, *für alle* $k \in \mathbb{N}$.

Jedes ausreichend lange Wort einer regulären Sprache läßt sich „aufpumpen".

Beweis.

Deterministischer endlicher Automat.

- Sei $n = |Q|$.
- Wenn $|z| \geq n$, werden mindestens $n+1$ Zustände durchlaufen.
- Sei j der erste wiederholte Zustand.
- Zerlege $z = uvw$ gemäß: $i \xrightarrow{u} j \xrightarrow{v} j \xrightarrow{w} f$.
  Dann gilt:
  - $|v| \geq 1$;
  - $|uv| \leq n$;
  - Jedes $uv^k w$ führt nach f.

□

Vervielfachen von Wortteilen

Pumping Lemma

Sei L eine reguläre Sprache.

- *Dann gibt es $n \in \mathbb{N}$, sodaß*
- *jedes $z \in L$ mit $|z| \geq n$*
- *derart in $z = uvw$ zerlegt werden kann, daß gilt:*
  - $|v| \geq 1$;
  - $|uv| \leq n$;
  - $uv^k w \in L$, *für alle* $k \in \mathbb{N}$.

Jedes ausreichend lange Wort einer regulären Sprache läßt sich „aufpumpen“.

Folgerung

Sei L eine formale Sprache.

- *Kann zu jedem $n \in \mathbb{N}$*
- *ein $z \in L$ mit $|z| \geq n$ gefunden werden,*
- *sodaß für jede Zerlegung $z = uvw$ mit*
  - $|v| \geq 1$;
  - $|uv| \leq n$;
- *ein k existiert, sodaß*
- $uv^k w \notin L$,
- *dann ist L nicht regulär.*

Grammatik

Definition

Eine Grammatik $\mathcal{G} = (T, N, S, \mathcal{P})$ besteht aus

- einem Alphabet Σ von Terminalzeichen;
- einem Alphabet N von Nicht-Terminalzeichen;
- einem Startsymbol $S \in N$;
- einer Menge von Produktionsregeln der Form

  $$l \Rightarrow r,$$

  mit $l \in (N \cup \Sigma)^* \setminus \Sigma^*$ und $r \in (N \cup \Sigma)^*$.

Definition

Sei $\mathcal{G} = (\Sigma, N, S, \mathcal{P})$ eine Grammatik, und $u, w \in (N \cup \Sigma)^*$.

- Dann ist w aus u ableitbar (Schreibweise: $u \rightarrow w$) wenn es
  - Zerlegungen $u = ale$ und $w = are$ gibt,
  - sodaß $l \Rightarrow r$ eine Produktionsregel der Grammatik ist.
- Die von $\mathcal{G}$ erzeugte Sprache ist definiert durch

  $$L(\mathcal{G}) = \{u \in \Sigma^* \mid S \rightarrow^* u\}.$$

Kontextfreie Grammtik

Definition

- Eine Grammatik heißt kontextfrei wenn jede Produktionsregel die Form

  $$A \Rightarrow r$$

  hat, wobei
  - A ein Nichtterminalsymbol;
  - r beliebig.
- Sie heißt rechtslinear wenn darüberhinaus r die Form x oder xB hat, wobei x terminal und B nichtterminal ist.

Definition

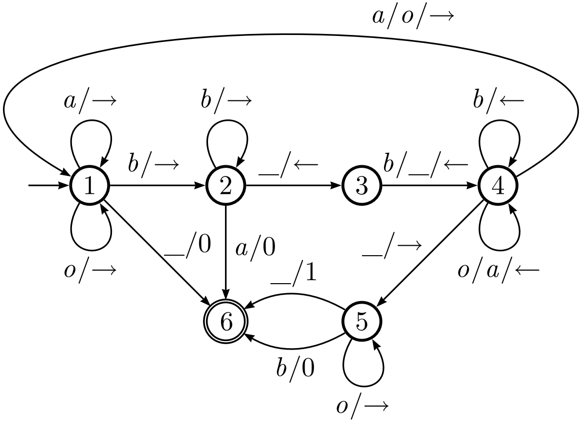
Eine *Sprache* heißt kontextfrei bzw rechtlinear wenn sie durch eine kontextfreie bzw. rechtslineare Grammatik definierbar ist.

Satz

Kontextfreie Sprachen werden durch Kellerautomaten erkannt.

Turingmaschine

Beispiel



erkennt $\{a^n b^{2^n} \mid n : \mathbb{N}\}$

Abschluß- und Entscheidbarkeitseigenschaften

Operation	reg	ktf	kts	rek	ra
Verkettung	ja	ja	ja	ja	ja
Iteration	ja	ja	ja	ja	ja
Vereinigung	ja	ja	ja	ja	ja
Durchschnitt	ja	nein	ja	ja	ja
Komplement	ja	nein	ja	ja	nein
$w \in L$	ja	ja	ja	ja	nein
$L = \varnothing$	ja	ja	nein	nein	nein
L endlich	ja	ja	nein	nein	nein
$L_1 = L_2$	ja	nein	nein	nein	nein
$L_1 \subseteq L_2$	ja	nein	nein	nein	nein