7 Die Chomsky-Hierarchie
Formale Sprachen, Grammatiken, Automaten

Σ Alphabet, $L \subseteq \Sigma^*$ formale Sprachen.

- Terme über Signatur (S, Sigma)
- Formeln
- While Programme
- Partielle Korrektheitsformeln
- Ausdrücke (primitiv rekursiv, μ-rekursiv)

Wie beschreibt man Sprachen ?

- Durch **Grammatiken** $G = (N, T, \Pi, Z)$ (spezielle Kalküle)
  N, T disjunkte Alphabete, Π Produktionen über $N \cup T$
  $Z \in N$ Startsymbol. Von G **erzeugte** Sprache:
  $L(G) = \{w \in T^* : Z \underset{\Pi}{\vdash} w\}$, d. h.
  $$Z \overset{1}{\underset{\Pi}{\vdash}} w_1 \overset{1}{\underset{\Pi}{\vdash}} \cdots \overset{1}{\underset{\Pi}{\vdash}} w_n = w \qquad n \geq 1$$

Problem: Wie entscheidet man $w \in L(G)$?

- Durch **Automaten** $A = (Q, N, T, \Pi, i, F)$
  Q endliche Zustandsmenge, Π Produktionen über $N \cup T$, die Übergang zwischen Konfigurationen beschreiben, i Initialkontext, F Finalkonfigurationen. Von A **akzeptierte** Sprache:
  $L(A) = \{w \in T^* : \exists f \in F \;\; i(w) \underset{\Pi}{\vdash} f\}$

Problem: Wie entscheidet man $w \in L(A)$?

7.1 Grammatiken

7.1 Definition Allgemeine Grammatiken

Eine Grammatik ist ein 4 Tupel

$$G = (N, T, \Pi, Z)$$

- Mit N endliche Menge **Nichtterminalsymbole**,
- T endliche Menge **Terminalsymbole**, $N \cap T = \varnothing$,
- Π endliche Menge von **Produktionen** $l \to r$ mit $l, r \in (N \cup T)^*$, wobei l mindestens ein Zeichen aus N enthält und $Z \in N$ Startsymbol ist.

Die von G **erzeugte Sprache** ist die Menge

$$L(G) = \{w \in T^* : Z \underset{\Pi}{\vdash} w\}$$

D.h. es gibt eine Ableitung $\{Z, w_1, \ldots, w_n = w\}$ für w mit $Z \overset{1}{\underset{\Pi}{\vdash}} w_1 \overset{1}{\underset{\Pi}{\vdash}} w_2 \overset{1}{\underset{\Pi}{\vdash}} \cdots \overset{1}{\underset{\Pi}{\vdash}} w$, d. h. $Z \overset{n}{\underset{\Pi}{\vdash}} w$ im Wortersetzungssystem $(N \cup T, \Pi)$, für ein $n \geq 1$.

Zwei G_1, G_2 Grammatiken sind **äquivalent**, falls $L(G_1) = L(G_2)$.

Beispiele

7.2 Beispiel Schreibweisen

a) $G = (N, T, \Pi, Z)$, $N = \{Z, Z_1\}$, $T = \{a, b\}$
$\Pi :: Z \to aZ_1, Z_1 \to bZ_1 \mid a$ 3 Produktionen.
Behauptung: $L(G) = \{ab^n a : n \in \mathbb{N}\}$
Beweis: „$\supseteq$" Gebe Ableitung an.
„$\subseteq$" $L(Z_1, G) = \{w \in T^* : Z_1 \underset{\Pi}{\vdash} w\} = \{b^n a : n \in \mathbb{N}\}$
Induktion nach i : $Z_1 \overset{i}{\underset{\Pi}{\vdash}} w$, $w \in T^*$
$i = 1 \rightsquigarrow w = a$
$i \to i + 1 \quad Z_1 \overset{i}{\vdash} b^i Z_1 \vdash b^i a$

b) $G = (N, T, \Pi, Z)$, $N = \{Z\}$, $T = \{a, b\}$
$\Pi :: Z \to aZb \mid \varepsilon$
Behauptung: $L(G) = \{a^n b^n : n \in \mathbb{N}\}$
Sei $\alpha \in V^* = (N \cup T)^*$, $\alpha \notin T^*$, $Z \overset{n}{\underset{\Pi}{\vdash}} \alpha$, so $\alpha = a^n Z b^n$.
Induktion nach n.
Dann „$\subseteq$" klar, „$\supseteq$" Angabe einer Ableitung.

c) $N = \{Z, T, S, A, B\}$, $T = \{a, b\}$
$\Pi :: Z \to TS$, $T \to aTA \mid bTB \mid \varepsilon$, $S \to \varepsilon$
$Aa \to aA, \quad Ab \to bA, \quad AS \to aS$
$Ba \to aB, \quad Bb \to bB, \quad BS \to bS$

Beispiele (Fort.)

Beispiel einer Ableitung:

$Z \overset{1}{\vdash} TS \overset{1}{\vdash} aTAS \overset{1}{\vdash} abTBAS \overset{1}{\vdash} abBAS \overset{1}{\vdash} abBaS \overset{1}{\vdash}$
$abaBS \overset{1}{\vdash} ababS \overset{1}{\vdash} abab$

Behauptung: $L(G) = \{ww : w \in T^*\}$

Für $w = w(a, b)$, sei $\hat{w} = w(A, B)$ das entsprechende Wort in den Großbuchstaben. Weiterhin sei ρ die Spiegelungsfunktion.

„$\supseteq$“ $Z \underset{\Pi}{\vdash} wT\rho(\hat{w})S \underset{T\to\varepsilon}{\vdash} w\rho(\hat{w})S \underset{\Pi}{\vdash} wwS \vdash ww$

„$\subseteq$“ Normierte Ableitungen: Erst T-Regeln bis $T \to \varepsilon$

$Z \vdash TS \underset{\Pi}{\vdash} wT\rho(\hat{w})S \vdash w\rho(\hat{w})S \underset{\Pi}{\vdash} ww$

Groß $\to$ klein, Vertauschregeln, mit $AS \to aS$, $BS \to bS$

d) $N = \{Z, A, B\}$, $T = \{a, b\}$

$\Pi :: Z \to \varepsilon \mid aAbZ \mid bBaZ, \quad A \to \varepsilon \mid aAbA,$
$\qquad B \to \varepsilon \mid bBaB$

Behauptung: $L(G) = \{w \in T^* : |w|_a = |w|_b\}$

$Z \underset{\Pi}{\vdash} \alpha \in (N \cup T)^*$, $|w|_a = |w|_b$ klar aus Regeln, also $L(G) \subseteq \{w \in T^* \mid |w|_a = |w|_b\}$

„$\supseteq$“ Ableitung angeben + Induktion $|w|_a = |w|_b$.

Eine andere Möglichkeit: $\Pi' : Z \to \varepsilon \mid aZb \mid bZa \mid ZZ$, dann $L(G') = L(G)$. Also sind G und G' äquivalent.

Frage: Einfachste Grammatik, die eine Sprache L erzeugt?

Beispiele (Forts.)

e) $N = \{Z, B, C\}$, $T = \{a, b, c\}$
$\Pi :: Z \to aZBC \mid aBC, \quad CB \to BC,$
$aB \to ab, bB \to bb,$
$bC \to bc, cC \to cc$

Behauptung: $L(G) = \{a^n b^n c^n : n \geq 1\}$

„$\supseteq$" $Z \overset{n-1}{\vdash} a^{n-1}S(BC)^{n-1} \underset{S \to aBC}{\overset{1}{\vdash}} a^n(BC)^n \underset{\Pi}{\vdash} a^n B^n C^n \underset{\Pi}{\vdash} a^n b^n C^n \underset{\Pi}{\vdash} a^n b^n c^n$

„$\subseteq$" Jede Ableitung lässt sich „normieren", erst alle Anwendungen von Z-Regeln (d. h. keine $CB \to BC$ Anwendung), dann die restlichen Regeln.

$Z \underset{\Pi}{\vdash} a^n ZW(B,C) \overset{1}{\vdash} a^{n+1}BCW(B,C) \vdash a^{n+1}b^{n+1}c^{n+1}$
mit $|W(B,C)|_B = |W(B,C)|_C = n$

Aus $aW(B,C)$ mit $|W(B,C)|_B = |W(B,C)|_C$ lässt sich nur $ab^n c^n$ ableiten (als terminales Wort).

7.2 Chomsky Hierarchie

7.3 Definition Klassifikation nach Form der Produktionen

Sei $G = (N, T, \Pi, Z)$ Grammatik.

0) G ist vom **Typ 0**, falls keine Einschränkungen.
1) G ist vom **Typ 1 (kontext-sensitiv)**, falls $l \to r \in \Pi$, so $l = xAy$, $r = xzy$ mit $x, y \in (N \cup T)^*$, mit $A \in N$, $z \in (N \cup T)^+$ (d. h. $|l| \leq |r|$).
   Ausnahme: $Z \to \varepsilon$ (ε - Regel) erlaubt, falls Z in keiner rechten Seite einer Produktion vorkommt.
2) G ist vom **Typ 2 (kontext-frei)**, falls $l \to r \in \Pi$, so $l = A$, $r = z$ mit $A \in N, z \in (N \cup T)^*$.
3) G ist vom **Typ 3 (rechts-linear)**, falls $l \to r \in \Pi$, so $l = A, r = aB|a|\varepsilon$, $A, B \in N, a \in T$.

Eine Sprache $L \subseteq T^*$ heißt **vom Typ i**, falls es eine Grammatik G vom Typ i gibt mit $L = L(G)$.

Im **Beispiel** 7.2: a) Typ 3, b) Typ 2, c) Typ 0, d) Typ 2, e) Typ 0.

Beachte: G rechts-linear, so G kontext-frei, G kontext-frei ohne ε-Regeln, so G kontext-sensitiv.

Normierungen für Grammatiken

7.4 Bemerkung Normierte Grammatiken - Eigenschaften

- Es gibt stets eine äquivalente Grammatik vom gleichen Typ, für die das Startsymbol in keiner rechten Seite einer Produktion vorkommt.
  $\Pi_1 = \Pi \cup \{Z_1 \to Z\}$
  Für Typ 3 $\{Z_1 \to \alpha$: für $Z \to \alpha \in \Pi\}$
- Für eine kontext-freie Grammatik G und Wörter $x, y, z, u, v \in (N \cup T)^*$ gilt
  $x \underset{\Pi}{\vdash} y$ so $uxv \underset{\Pi}{\vdash} uyv$ (gilt sogar für beliebige G)
  $xy \overset{n}{\underset{\Pi}{\vdash}} z$, so gibt es $z_1, z_2 \in (N \cup T)^*$ mit $z = z_1 z_2$ und $x \overset{\leq n}{\underset{\Pi}{\vdash}} z_1$, $y \overset{\leq n}{\underset{\Pi}{\vdash}} z_2$ (Ind. nach n).
- Für jede kontext-freie Grammatik G gibt es eine ε-freie kontext-freie Grammatik G_1 mit $L(G_1) = L(G) - \{\varepsilon\}$.
  Ist $\varepsilon \in L(G)$, dann gibt es eine kontext-freie Grammatik G' mit $L(G') = L(G)$, wobei die einzige Regel in G', die ε als rechte Seite hat, $Z' \to \varepsilon$ ist. Hierbei ist Z' Startsymbol von G', und Z' kommt in keiner rechten Seite einer Regel vor.

Normierungen - Abschlusseigenschaften

Beweisidee:

Sei $U_1 = \{X : X \to \varepsilon \in \Pi\}$ und
$U_{i+1} = U_i \cup \{X : X \to \alpha \in \Pi, \alpha \in U_i^*\}$.

Offenbar $U_i \subseteq N$, $U_i \subseteq U_{i+1}$. D. h. es gibt k mit $U_k = U_{k+1}$ und somit $U_k = U_{k+v}$, für $v = 0, 1, 2, 3 \ldots$

Behauptung: $X \underset{\Pi}{\vdash} \varepsilon \text{ gdw } X \in U_k$. (Beweis: Übung).

Insbesondere: $\varepsilon \in L(G) \ \text{ gdw } Z \in U_k$.

Definiere: $G_1 = (N, T, \Pi_1, Z)$ mit
$X \to \alpha' \in \Pi_1 \text{ gdw}$ es gibt $X \to \alpha \in \Pi, \alpha' \neq \varepsilon$ entsteht durch Streichen von Buchstaben in U_k (kein Streichen erlaubt).

7.5 Lemma Abschlusseigenschaften von $\mathcal{L}_i$

$\mathcal{L}_i$ ist abgeschlossen bzgl. $\cup, \circ, *$ für $i = 0, 1, 2, 3$.

Beweis:

$L_1 \circ L_2 = \{uv : u \in L_1, v \in L_2\}$

$L^* = \{u_1 \ldots u_n : n \in \mathbb{N}, u_i \in L\} = \bigcup_{n \geq 0} L^n \quad (L^0 = \{\varepsilon\})$

Sei L_j erzeugt von $G_j = (N_j, T_j, \Pi_j, Z_j)$. G_j vom Typ i
$(i = 0, 1, 2, 3)$, $j = 1, 2$.

Abschlusseigenschaften

O.B.d.A. auf linken Seiten von Produktionen kommen keine terminalen Buchstaben vor. (Für $a \in T$ Platzhalter $A_a \in N$, ersetze Vorkommen von a in linker Seite durch A_a. Hinzunahme von Produktionen $A_a \to a$). $N_1 \cap N_2 = \varnothing$.

a) $\cup$: $G =$
   $(N_1 \cup N_2 \cup \{Z\}, T_1 \cup T_2, \Pi_1 \cup \Pi_2 \cup \{Z \to Z_1 \mid Z_2\})$
   Für Typ (3): $Z \to \alpha$ für $Z_1 \to \alpha \in \Pi_1$ oder $Z_2 \to \alpha \in \Pi_2$.
   G ist vom Typ i und $L(G) = L(G_1) \cup L(G_2)$.

b) $\circ$: $G = (N_1 \cup N_2 \cup \{Z\}, T_1 \cup T_2, \Pi_1 \cup \Pi_2 \cup \{Z \to Z_1 Z_2\})$
   G ist vom Typ i für $i = 0, 1, 2$.
   Behauptung: $L(G) = L(G_1) \circ L(G_2)$.
   „$\supseteq$“ $Z \overset{1}{\underset{\Pi}{\vdash}} Z_1 Z_2 \underset{\Pi}{\vdash} u Z_2 \vdash uv$ für $u \in L(G_1), v \in L(G_2)$.
   „$\subseteq$“ $Z \overset{1}{\underset{\Pi}{\vdash}} Z_1 Z_2 \underset{\Pi}{\vdash} X$ und $X \in (T_1 \cup T_2)^*$.
   Dann $Z_1 \underset{\Pi_1}{\vdash} X_1$ und $Z_2 \underset{\Pi_2}{\vdash} X_2$, $X = X_1 X_2$.
   Da linke Seiten nur aus nichtterminalen Buchstaben und
   $N_1 \cap N_2 = \varnothing$, d. h. keine Vermischungen.
   Für Typ 3 - Grammatiken:
   Π_1' entstehe aus Π_1 durch Ersetzen von jeder Produktion $X \to a|\varepsilon$ durch $X \to a Z_2$ bzw. $X \to \alpha$ für $Z_2 \to \alpha \in \Pi_2$.
   $G = (N_1 \cup N_2, T_1 \cup T_2, \Pi' \cup \Pi_2, Z_1)$ erfüllt Forderung.

Abschlusseigenschaften (Fort.)

c) $* : L^* = \{w : \exists n \in \mathbb{N}, w \in L^n, w = v_1 \dots v_n, v_i \in L\}$
Sei $G =$
$(N_1 \cup \{Z\}, T_1, \Pi_1 \cup \{Z \to \varepsilon, Z \to Z_1, Z_1 \to Z_1 Z_1\})$.
Dann ist G vom Typ i für $i = 0, 1, 2$ und $L(G) = L(G_1)^*$.
Für Typ 3 Grammatiken: Übung.

7.6 Folgerung

- Jede endliche Sprache ist vom Typ 3:
  $w = a_1 \dots a_n \qquad a_i \in T \qquad n \geq 0$
  $Z \to a_1 A_1, A_1 \to a_2 A_2, \dots, A_{n-1} \to a_n A_n, A_n \to \varepsilon$
  $N = \{Z, A_1, \dots, A_n\}$
- $\mathcal{L}_{\text{endl}} \subsetneq \mathcal{L}_{T_3} \subsetneq \mathcal{L}_{T_2} \subsetneq \mathcal{L}_{T_1} \subsetneq \mathcal{L}_{T_0}$
- Wie ordnen sich die Sprachklassen in Hierarchie ein?

$$\mathcal{L}_{\text{endl}} \subsetneq \mathcal{L}_{\text{prim-rek}} \subsetneq \mathcal{L}_{\text{rek-entsch.}} \subsetneq \mathcal{L}_{\text{rek-aufzb.}}$$

Ist $L(G)$ entscheidbar für beliebiges G ?

7.7 Lemma

Sei $G = (N, T, \Pi, Z)$ Grammatik, dann ist $L(G)$ rekursiv aufzählbar.

Idee: Führe systematisch alle Ableitungen aus Z der Länge nach durch.

Ableitbare Wörter aus $(N \cup T)^*$ in $1, 2, 3 \dots$ Ableitungsschritte.

$L(G)$ ist rekursiv aufzählbar

- Verfahren hält mit Eingabe w gdw $s \overset{i}{\underset{\Pi}{\vdash}} w$ für ein i d. h. w kommt in Stufe i vor.
- Verfahren ist effektiv und hält bei Eingabe w gdw $w \in L(G)$.

Formal: Sei $\Sigma = N \dot{\cup} T \dot{\cup} \{\vdash\}, \Pi = \{l_1 \to r_1, \dots, l_n \to r_n\}$ und

$$M = \{w \in \Sigma^*\text{: Es gibt } w_1, \dots, w_m \in (N \cup T)^* \text{ mit } w = \;\vdash Z \vdash w_1 \vdash \cdots \vdash w_m \vdash \text{ und } Z \overset{1}{\underset{\Pi}{\vdash}} w_1, w_i \overset{1}{\underset{\Pi}{\vdash}} w_{i+1} \text{ für } i \geq 1\}$$

M ist die Menge der Ableitungen in G.

Für $\alpha, \beta \in V^*$ sei

$$Q_i(\alpha, \beta) \text{ gdw } \alpha \underset{l_i \to r_i}{\vdash} \beta \text{ gdw } \exists\, \alpha', \alpha'' \leq \alpha.\ \alpha = \alpha' l_i \alpha'' \wedge \beta = \alpha' r_i \alpha''$$

$$Q(\alpha, \beta) \text{ gdw } \alpha \underset{\Pi}{\vdash} \beta \text{ gdw } Q_1(\alpha, \beta) \vee \cdots \vee Q_n(\alpha, \beta).$$

Offenbar $Q_1, \dots, Q_n \in \mathcal{P}(\Sigma), Q \in \mathcal{P}(\Sigma)$.

M ist primitiv rekursiv (verwende Anfangswort, Teil- und Endwort).

- $x \in L(G)$ gdw $\exists w.\ w \in M \wedge$ Endwort$(\vdash x \vdash, w)$.

Umkehrung

7.8 Lemma

$L \subset \Sigma^*$ rekursiv aufzählbar, dann gibt es eine Grammatik $G = (N, \Sigma, \Pi, Z)$ mit $L = L(G)$.

Beweisidee: Simuliere mit der Grammatik die TM-Schritte einer TM die L akzeptiert rückwärts.

Sei o.B.d.A. T eine TM, die L akzeptiert mit nur einem Haltezustand q. D. h. $F = \{q\}$. $T = (Q, \Sigma, \Gamma, \delta, q_0, F)$

Die Konfigurationen von T werden in Klammern eingeschlossen: $[uq_iv]$.

Produktionen von G bewirken:

1-Gruppe: $Z \underset{G}{\vdash} [uqv] \quad u, v \in \Gamma^* \;\; (u, v$ lang genug).

2-Gruppe: $[k_{i+1}] \underset{G}{\vdash} [k_i]$, falls $k_i \underset{T}{\vdash} k_{i+1}$, dabei ist $|k_i| = |k_{i+1}|$.

Dann gilt: $[uqv] \underset{G}{\vdash} [\Box^s q_0 \Box x \Box^t]$, falls $q_0 \Box x \overset{*}{\underset{T}{\vdash}} uqv \quad (x \in \Sigma^*)$.

3-Gruppe: $[\Box^s q_0 \Box x \Box^t] \underset{G}{\vdash} x$ für alle $s, t \in \mathbb{N}, x \in \Sigma^*$.

Wählt man s, t genügend groß, so verlässt die TM bei ihrer Berechnung nie den Block $\Box^s x \Box^t$.

Produktionen (Forts.)

Produktionen 1-Gruppe:
$Z \to [Z_0], Z_0 \to Z_0 b \mid b Z_0 \mid q \ (b \in \Gamma)$.
Produktionen 2-Gruppe: z. B. aus Turing Programm

$q_i : a$	$\rightsquigarrow$	$q_{i+1}a$	$\longrightarrow$	$q_i b$	$b \in \Gamma$
$q_i : R$	$\rightsquigarrow$	$b q_{i+1}$	$\longrightarrow$	$q_i b$	$b \in \Gamma$
$q_i : L$	$\rightsquigarrow$	$q_{i+1} b$	$\longrightarrow$	$b q_i$	$b \in \Gamma$
$q_i : q_k$	$\rightsquigarrow$	q_k	$\longrightarrow$	q_i	
$q_i : a, q_k$	$\rightsquigarrow$	$q_k a$	$\longrightarrow$	$q_i a$	und $q_{i+1} b \to q_i b \ (b \neq a)$

Produktionen 3-Gruppe:

$q_0 \to T_1$, $\Box T_1 \to T_1$, $[T_1 \Box \to T_2$

$T_2 b \to b T_2$, $b \in \Sigma$, $T_2 \to T_3$,

$T_3 \Box \to T_3$, $T_3] \to \varepsilon$.

G **ist Typ-0 Grammatik**!
Hierbei ist $N = \{Z, Z_0, T_1, T_2, T_3, [,]\} \cup Q \cup (\Gamma - \Sigma)$
Es gilt $Z \underset{G}{\vdash} x \in \Sigma^* \text{ gdw } T$ akzeptiert x, d. h. $L(G) = L$.

7.9 Satz

$L \subseteq \Sigma^*$ ist rekursiv aufzählbar gdw es gibt eine Typ-0 Grammatik $G = (N, \Sigma, \Pi, Z)$ mit $L = L(G)$.

Insbesondere sind Typ-0-Sprachen abgeschlossen gegenüber $\cap$ aber nicht gegen $\neg$ (Komplement) und es gibt nicht entscheidbare Typ-0-Sprachen.

Wortprobleme

7.10 Definition Wortproblem, uniformes Wortproblem

Sei $G = (N, \Sigma, \Pi, Z)$. Das **Wortproblem** für G ist definiert:

$$WP(x) \text{ gdw } x \in L(G) \quad (x \in \Sigma^*)$$

Ist $\mathcal{G}$ eine Klasse von Grammatiken, so ist das **uniforme Wortproblem** für $\mathcal{G}$ definiert durch

$$UWP(G, x) \text{ gdw } x \in L(G) \quad (G \in \mathcal{G}, x \in T_G^*)$$

7.11 Folgerung

- UWP ist nicht entscheidbar für Typ-0 Grammatik.
- Es gibt Typ-0 Grammatik mit unentscheidbaren WP.
- Das uniforme WP für Typ 1 Grammatiken ist primitiv rekursiv.

$$\mathcal{L}_{\text{endl.}} \subsetneq \mathcal{L}_{\text{Typ}-3} \subseteq \mathcal{L}_{\text{Typ}-2} \subseteq \mathcal{L}_{\text{Typ}-1} \subseteq \mathcal{L}_{\text{prim-rek}} \subsetneq \mathcal{L}_{\text{Typ}-0} = \mathcal{L}_{\text{rek-aufzb.}}$$

endliche Automaten EA

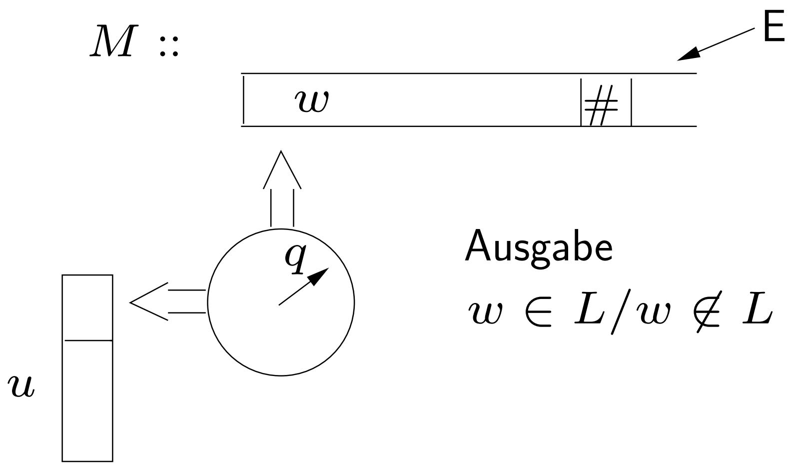
Keller-automaten PDA

linear beschränkte Automaten LBA

TM als akzeptierende Automaten

Formale Sprachen und akzeptierende Automaten

Einschränkungen der Turing-Maschinen: Möglichkeiten



- nur lesen im EB
- lesen dann rechts
- Endmarkierungen
- Hilfsband als Keller
- akzeptieren/verwerfen durch Zustand

Konfigurationen: $uqw \underset{M}{\vdash} u'q'w'$ mit Hilfe von Produktionen.

7.12 Definition Automaten für Sprachen

Ein **Automat** (oder **Akzeptor**) $A = (Q, N, T, \Pi, i, F)$ mit endlicher **Zustandsmenge** Q, endlicher Menge N von **Hilfssymbolen** und endlichem **Eingabealphabet** T, so dass Q, N, T paarweise disjunkt sind, $i : T^* \to (N \cup T)^* \cdot Q \cdot (N \cup T)^*$: **Initialkonfiguration** zur Eingabe $w \in I^*$, einer endlichen Menge von **Finalkonfigurationen** F der Form $lqr \in (N \cup T)^* q (N \cup T)^*$ und einer endlichen Menge Π von **Produktionen** der Form $lqr \to l'q'r'$ $(l, l', r, r' \in (N \cup T)^* q, q' \in Q)$.

$L(A) = \{w \in T^* : \exists f \in F \quad i(w) \underset{\Pi}{\vdash} f\}$ die von A **akzeptierte Sprache**.

7.3 Endliche Automaten - reguläre Sprachen - Typ 3-Sprachen

Typ-3 Grammatik: $G = (N, T, \Pi, \Sigma)$, Π mit Produktionen der Form $A \longrightarrow aB|a|\varepsilon$, $A, B \in N, a \in T$

7.13 Definition Endliche Automaten

a) Ein (deterministischer) **endlicher Automat (DEA)** ist ein 5-Tupel $A = (Q, \Sigma, \Pi, q_0, F)$ mit $q_0 \in Q$ **Startzustand**, $F \subset Q$ Menge der **Finalzustände** (akzeptierende Zustände). $\Pi = \{qa \longrightarrow q' : q, q' \in Q, a \in \Sigma\}$: Für jedes Paar $(q, a) \in Q \times \Sigma$ gibt es genau eine Produktion $qa \longrightarrow q'$.

b) Ein **indeterministischer endlicher Automat (NEA)** ist ebenfalls ein 5-Tupel A wie eben mit dem Unterschied, dass es für jedes Paar $(q, a) \in Q \times \Sigma$ eine endliche (eventuell leere) Menge von Produktionen der Form $qa \longrightarrow q'$ sowie Produktionen der Form $q \longrightarrow q'$ (Spontanübergänge, ε-**Übergänge**) gibt.

c) **Initialkonfiguration** bei Eingabe $w \in \Sigma^*$: $q_0 w$,
d. h. $i(w) = q_0 w$ für $w \in \Sigma^*$.
Finalkonfigurationen: F.

d) Die von A **akzeptierte Sprache** ist die Menge
$L(A) = \{w \in \Sigma^* : q_0 w \underset{\Pi}{\vdash} f \text{ für ein } f \in F\}$.
Schreibe auch $q_0 w \underset{A}{\vdash} f$.

Beispiele - Darstellungsarten
Zustandsgraph oder Automatendiagramme

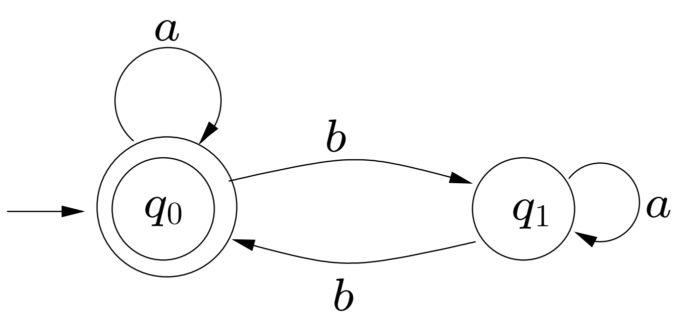
7.14 Beispiel

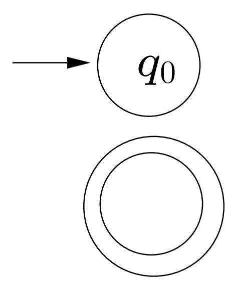
1. $A = (\{q_0, q_1\}, \{a, b\}, \Pi, q_0, \{q_0\})$
   $\Pi :: q_0 a \to q_0, q_0 b \to q_1, q_1 a \to q_1, q_1 b \to q_0$

   Behauptung: $q_0 w \underset{A}{\vdash} q_0 \text{ gdw } |w|_b$ gerade.

   Beweis: Induktion nach $|w|_b$,
   d. h. $L(A) = \{w \in \{a, b\}^* : |w|_b \text{ gerade}\}$.

   Diagramm: Knoten $\leftrightarrow$ Zustand, gerichtete Kante $\leftrightarrow$ Produktion





Anfangszustand

akzeptierender Zustand

   Matrix-Tabelle:

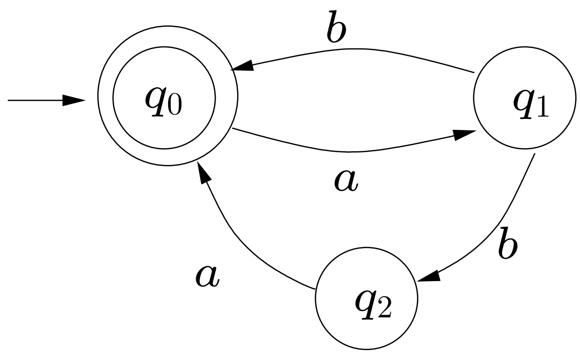
	a	b
q_0	q_0	q_1
q_1	q_1	q_0

Beispiele - Darstellungsarten Zustandsgraph oder Automatendiagramme (Forts.)

Bei indeterminierten Automaten: mehrere Kanten aus Zustand können mit Buchstaben a oder ε markiert sein.

Tabellendarstellung: Zustandsmengen + ε-Spalte.

2. Betrachte



Behauptung: $L(A) = \{ab, aba\}^*$

„$\supseteq$" klar. „$\subseteq$" Es gelte: $q_0 w \underset{A}{\vdash} q_0$.

Dann $w = \varepsilon$ oder w fängt mit a an.

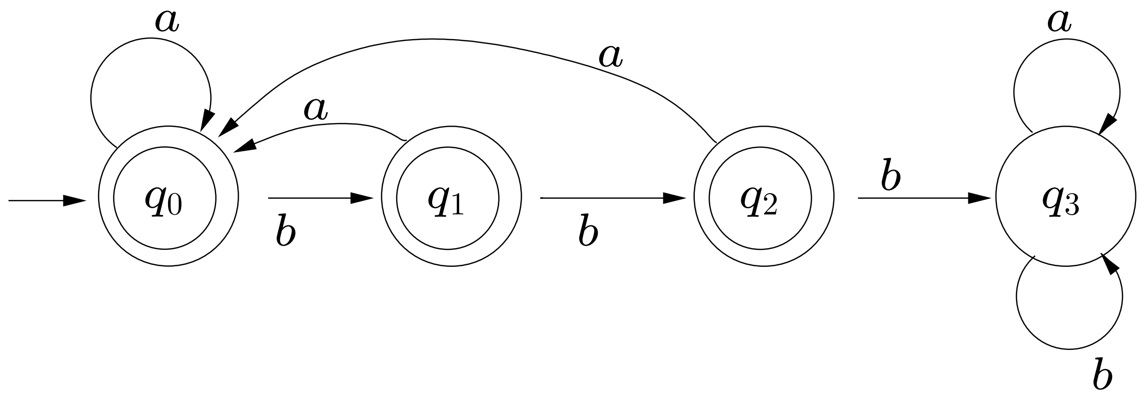
$q_0 a w' \vdash q_1 w' \vdash q_0 \rightsquigarrow w'$ fängt mit b an.

$q_1 b w'' \vdash q_0 w''$ Induktion

$q_1 b w'' \vdash q_2 w''$ w'' mit a +Ind.

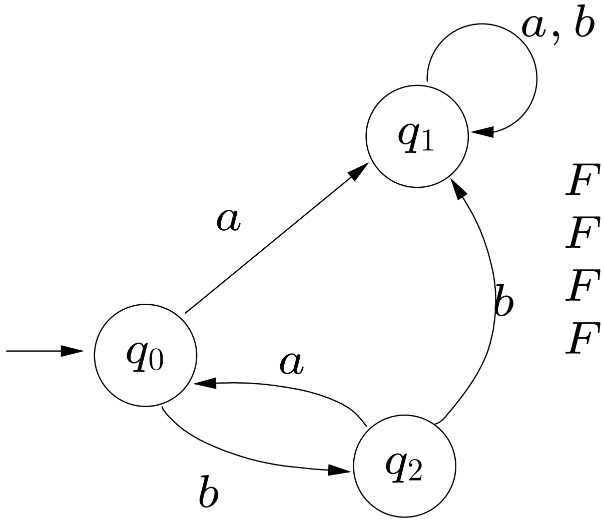
Beispiele (Fort.)

3. $L = \{w \in \{a, b\}^* : w$ enthält nicht bbb als TW$\}$.



   Beschreibung der Wege, die von q_0 nach q_i führen.
   $q_0 \rightsquigarrow q_0 : \varepsilon, \{a\}^*, \{a\}^*\{ba\}^*\{a\}^*, a^*bbaa^*, \ldots$
   Reguläre Ausdrücke zur Beschreibung von Sprachen.

4. Betrachte



$$
\begin{aligned}
F &= \{q_0\} \rightsquigarrow L(A) = \{ba\}^* \\
F &= \{q_2\} \rightsquigarrow L(A) = \{ba\}^*b \\
F &= \{q_1\} \rightsquigarrow L(A) = (ba)^*(a + bb)\Sigma^* \\
F &= \{q_0, q_2\} = (ba)^* + (ba)^*b \\
&= (ba)^*(\varepsilon + b)
\end{aligned}
$$

Vereinigung

   Operationen: Verkettung, Vereinigung, Iteration $(*)$.

Beispiele (Fort.)

5. Dezimalzahlen, die durch 5 teilbar sind.

	0	1	2	3	4	5	6	7	8	9
q_0	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_1	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_2	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_3	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_4	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4

$q_0 w \vdash q_i \text{ gdw } w \equiv i (\text{mod } 5)$, $F = \{q_0\}$
Automat mit 2 Zustände genügt!
$\rightsquigarrow$ **Äquivalente Automaten, minimale Automaten**.

Endliche Automaten und Typ-3-Grammatiken

7.15 Lemma Charakterisierungssatz

Ist $A = (Q, \Sigma, \Pi, q_0, F)$ EA, so ist $L(A)$ eine Typ-3 (rechtslineare) Sprache.

Beweis:

Definiere rl-Grammatik $G = (N, \Sigma, \Pi_G, Z)$ mit $N = Q$, $Z = q_0$, so dass für alle $x \in \Sigma^*$ gilt:

$$(*) \qquad q_0 x \underset{A}{\vdash} q_i \text{ gdw } Z \underset{G}{\vdash} x q_i$$

Endliche Automaten und Typ-3-Grammatiken

Definiere:

$$\begin{aligned}\Pi_G = \quad & \{q_i \to aq_j : q_ia \to q_j \in \Pi\} \\ & \cup \{q_i \to a : q_ia \to q \in \Pi \wedge q \in F\} \\ & \cup \{Z \to \varepsilon : \text{ falls } q_0 \in F\}\end{aligned}$$

G ist rechts-lineare Grammatik.

Behauptung: $(*)$ gilt für G:
Beweis: Induktion nach $|x|$.

„$\Rightarrow$“ $x = \varepsilon$, $q_0\varepsilon \underset{A}{\vdash} q_0$, $Z = q_0 \underset{G}{\vdash} q_0$
$x \rightsquigarrow xa$, $q_0x \underset{A}{\vdash} q_i$, Ind. Vor $Z \underset{G}{\vdash} xq_i$
Sei $q_ia \to q_j \in \Pi$, dann $q_0xa \underset{A}{\vdash} q_ia \underset{A}{\vdash} q_j$
Da $q_i \to aq_j \in \Pi_G$ folgt $Z \underset{G}{\vdash} xq_i \overset{1}{\vdash} xaq_j$

„$\Leftarrow$“ $x = \varepsilon$, $Z \underset{G}{\vdash} q_i$, dann $q_i = q_0$
$x \rightsquigarrow xa$, $Z \underset{G}{\vdash} xaq_j$. Da Π_G rechts-linear ist, folgt
$Z \underset{G}{\vdash} xq_i \overset{1}{\vdash} xaq_j$ mit Regel $q_i \to aq_j \in \Pi_G$.
Dann aber $q_ia \to q_j \in \Pi$.
Nach Ind. Vor: $q_0x \underset{A}{\vdash} q_i$ und somit $q_0xa \underset{A}{\vdash} q_ia \underset{\Pi}{\overset{1}{\vdash}} q_j$.

Endliche Automaten und Typ-3-Grammatiken (2)

Behauptung: $L(A) = L(G)$

„$\subseteq$“ $x \in L(A)$

-: $x = \varepsilon$, so ist $q_0 \in F$, $Z \to \varepsilon \in \Pi_G$, d. h. $x \in L(G)$

-: $x = ya$, $q_0 y \underset{A}{\vdash} q_i$, $q_i a \to q$ mit $q \in F$.

Dann folgt aus $(*)$ $Z \underset{G}{\vdash} yq_i \overset{1}{\vdash} ya$, da $q_i \to a \in \Pi_G$,

d. h. $ya \in L(G)$. Also $x \in L(G)$

„$\supseteq$“ $x \in L(G)$

-: $x = \varepsilon$, so $Z \to \varepsilon \in \Pi_G \rightsquigarrow q_0 \in F \rightsquigarrow x \in L(A)$

-: $x = ya$, $Z \underset{G}{\vdash} yq_i \overset{1}{\underset{G}{\vdash}} ya$. Wegen $(*)$ ist $q_0 y \underset{A}{\vdash} q_i$ und $q_i a \to q$ mit $q \in F$, d. h. $q_0 ya \underset{A}{\vdash} q_i a \underset{A}{\vdash} q \in F$.

Also $x \in L(A)$.

Beachte:

G ist rechts-linear und „eindeutig“, d. h. ist $w \in L(G)$, so gibt es genau eine Ableitung für w.

Falls A NEA, so Problem mit Spontanübergängen, diese würden Regeln der Form $q_i \to q_j$ bedeuten. Sonst ok.

Beispielkonstruktion

7.16 Beispiel Sei $A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \Pi, q_0, \{q_0\})$.

Π	a	b
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

$G_A = (N, \Sigma, \Pi_G, Z)$, $N = \{q_0, \ldots, q_3\}$, $Z = q_0$

$$\begin{array}{lll} \Pi_G : & q_0 \longrightarrow aq_2|bq_1|\varepsilon & (q_0 \in F) \\ & q_1 \longrightarrow aq_3|bq_0|b & (q_0 \in F) \\ & q_2 \longrightarrow aq_0|a|bq_3 & (q_0 \in F) \\ & q_3 \longrightarrow aq_1|bq_2 & \end{array}$$

Beachte: $|\Pi_G| \leq 2 \cdot |\Sigma| \cdot |Q| + 1$.

Frage: Wird jede Typ-3 Sprache von einem DEA akzeptiert?

Problem: Bei Typ-3 Grammatiken ist $A \longrightarrow aB$ und $A \longrightarrow aC$ erlaubt, d. h. Indeterminismus.

Endliche Automaten und Typ-3-Grammatiken (3)

7.17 Lemma Charakterisierungssatz

Zu jeder Typ-3 Sprache L gibt es NEA A mit $L = L(A)$.

Beweis: Sei G Typ-3 Grammatik $G = (N, T, \Pi_G, Z)$ mit $L = L(G)$.

Definiere:

$A = (Q, T, \Pi_A, q_0, F)$ mit $Q = N \dot{\cup} \{S\}$, $q_0 = Z$.

$$\begin{aligned} \Pi_A : &\{Xa \to Y : \text{ für } X \to aY \in \Pi_G\} \\ &\cup \{Xa \to S : \text{ für } X \to a \in \Pi_G\} \end{aligned}$$

$F = \{S\} \cup \{X \mid X \to \varepsilon \in \Pi_G\}$

Behauptung:

a) $q_0 w \underset{A}{\vdash} X \text{ gdw } Z \underset{G}{\vdash} wX$ für $X \in N, w \in T^*$.

b) $w \in L(A) \text{ gdw } w \in L(G) \text{ gdw } Z \underset{G}{\vdash} w$ für $w \in T^*$.

Beweis:

a) Induktion nach $|w|$ :: -: $w = \varepsilon$

„⇒“ $X = q_0 = Z$,

„⇐“ dito.

-: $w = va$

„⇒“ $q_0 va \underset{A}{\vdash} X$, $x \in N$: Dann $q_0 v \underset{A}{\vdash} Y$, $Y \in N$ und $ya \vdash X$.

D. h. nach Ind. Vor. $Z \underset{G}{\vdash} vY \overset{1}{\underset{G}{\vdash}} vaX$.

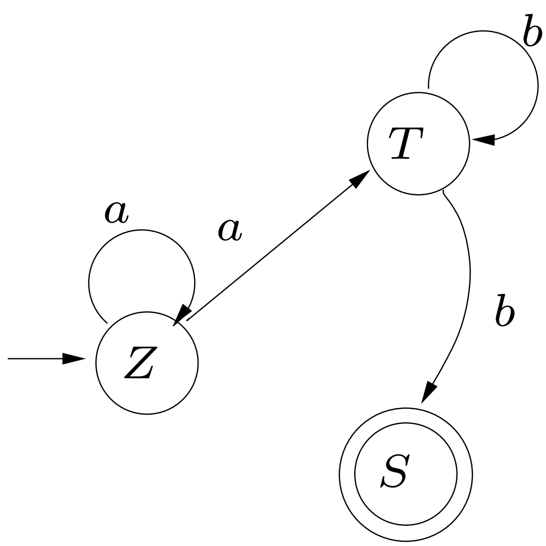
Konstruktion-Beispiele

„$\Leftarrow$“ $Z \underset{G}{\vdash} vaX$, $X \in N$. Dann $Z \underset{G}{\vdash} vY$, für ein $Y \in N$ und $Y \to aX \in \Pi_G$. Dann $q_0 va \underset{A}{\vdash} Ya \underset{A}{\vdash} X$.

b) $w \in L(A)$.
Dann $q_0 w \underset{A}{\vdash} S$ oder $q_0 w \vdash X$ mit $X \to \varepsilon \in \Pi_G$. Dann aber $w = va$, $q_0 w \underset{A}{\vdash} xa \vdash S$.
$x \in N \rightsquigarrow Z \underset{G}{\vdash} vX \underset{G}{\vdash} va \in L(G)$ oder $Z \underset{G}{\vdash} wX \underset{G}{\vdash} w \in L(G)$. $\rightsquigarrow$ Behauptung.

7.18 Beispiel

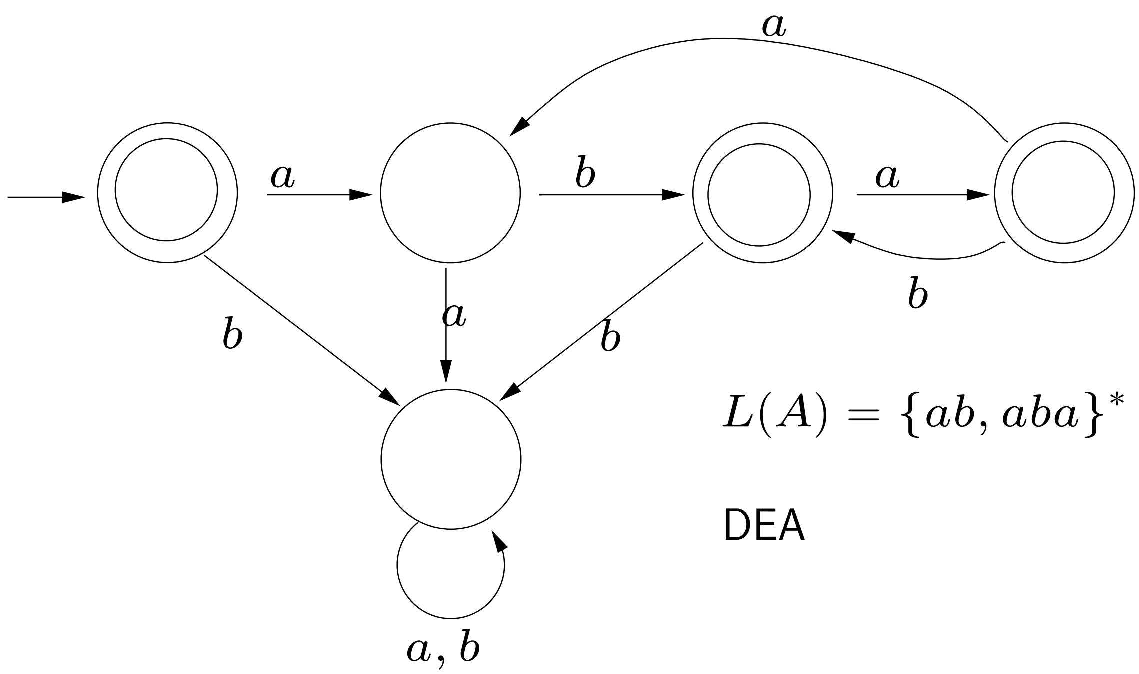
1. $G = (N, \Sigma, \Pi_G, Z)$, $N = \{Z, T\}$, $\Sigma = \{a, b\}$
   $\Pi_G :: Z \to aZ|aT$, $T \to bT|b$
   Behauptung: $L(G) = \{a^n b^m : n, m \geq 1\}$ (klar).
   Konstruktion:



ohne - ε Übergänge

Beispiele

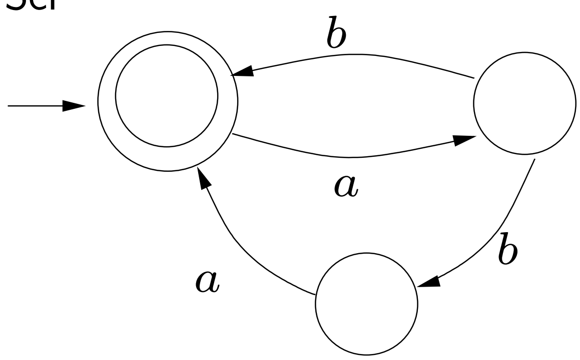
2. Betrachte



$$L(A) = \{ab, aba\}^*$$

DEA

3. Sei

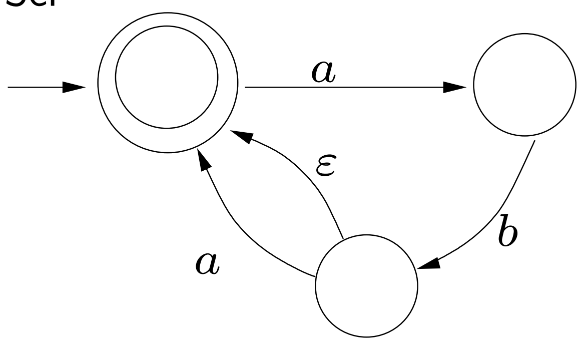


$$L(A) = \{ab, aba\}^*$$

NEA

Beispiele

4. Sei



$L(A) = \{ab, aba\}^*$
NEA
fast deterministisch

Kann man Spontanübergänge vermeiden?
JA: Idee $q \sim q'$ gdw es gibt $q_0, \ldots, q_n \in Q$ $q_0 = q, q_n = q'$, $q_i \to q_{i+1} \in \Pi$. Lässt sich effektiv berechnen!

$$\Pi^* = \{qa \to q' : \exists q''(q \sim q'' \wedge q''a \to q' \in \Pi)\}$$

$$F^* = \{q : \exists f \in F : q \sim f\}$$

Dann $L(A) = L(A^*)$.

Wir haben somit:

7.19 Lemma

$L \subseteq T^*$ ist Typ-3 Sprache gdw $L = L(A)$ für ein NEA A.

Charakterisierungssatz für r.l. Sprachen

7.20 Satz

Zu jedem NEA A gibt es einen DEA A' mit $L(A) = L(A')$.

Beweis: Sei $A = (Q, \Sigma, \Pi, q_0, F)$ ein NEA. A enthalte keine ε-Übergänge. Definition DEA $A' = (Q', \Sigma, \Pi', q_0', F')$ mit

- $Q' =$ Potenzmenge von $Q = \{T : T \subseteq Q\}$
- $\Pi' = \{T_a \to \{q' \in Q : \exists q \in T \;\; qa \to q' \in \Pi\} : T \in Q', a \in \Sigma\}$
- $q_0' = \{q_0\}$
- $F' = \{T \subseteq Q : T \cap F \neq \varnothing\}$

Behauptung: $L(A') = L(A)$.

Beweis: Es gilt $Ty \underset{A'}{\vdash} \{q' \in Q : \exists q \in T \;\; qy \underset{A}{\vdash} q'\} =: T'$ für $T \subseteq Q, y \in \Sigma^*$.

Ind. nach $|y| : y = \varepsilon$, so $T' = T$, da keine Spontanübergänge.

Sei $y = az$, $a \in \Sigma$, dann

$$\begin{aligned} Taz &\underset{A'}{\vdash} \{q' : \exists q \in T \;\; qa \to q' \in \Pi\}z \\ &\underset{\substack{A' \\ \text{Ind.Vor.}}}{\vdash} \{q'' : \exists q' \exists q \in T \;\; qa \to q' \in \Pi, q'z \underset{A}{\vdash} q''\} \\ &= \{q'' : \exists q \in T \;\; qaz \underset{A}{\vdash} q''\} \end{aligned}$$

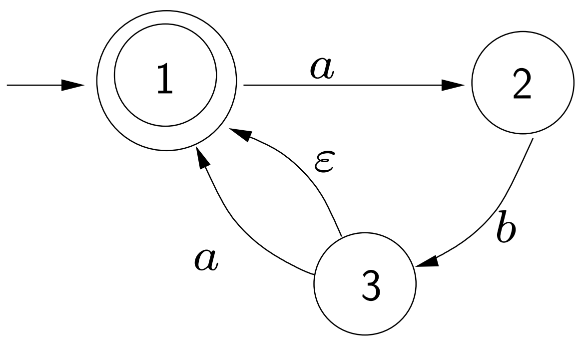
Beispiele

Sei

$$
\begin{aligned}
y \in L(A') \quad & \text{gdw } \exists T \in Q \; (T \cap F \neq \varnothing \wedge \{q_0\}y \underset{A'}{\vdash} T) \\
& \text{gdw } \{q \in Q : q_0 y \underset{A}{\vdash} q\} \cap F \neq \varnothing \\
& \text{gdw } y \in L(A)
\end{aligned}
$$

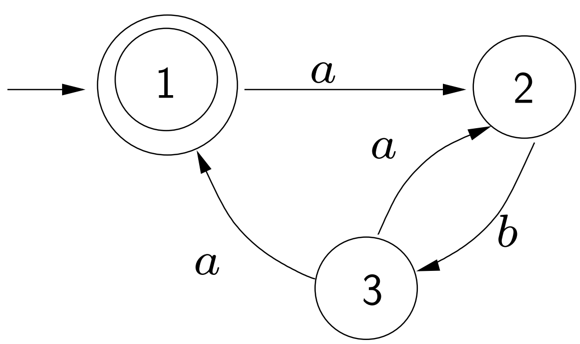
7.21 Beispiel

- Sei



hat Spontanübergänge
$3 \sim 1$

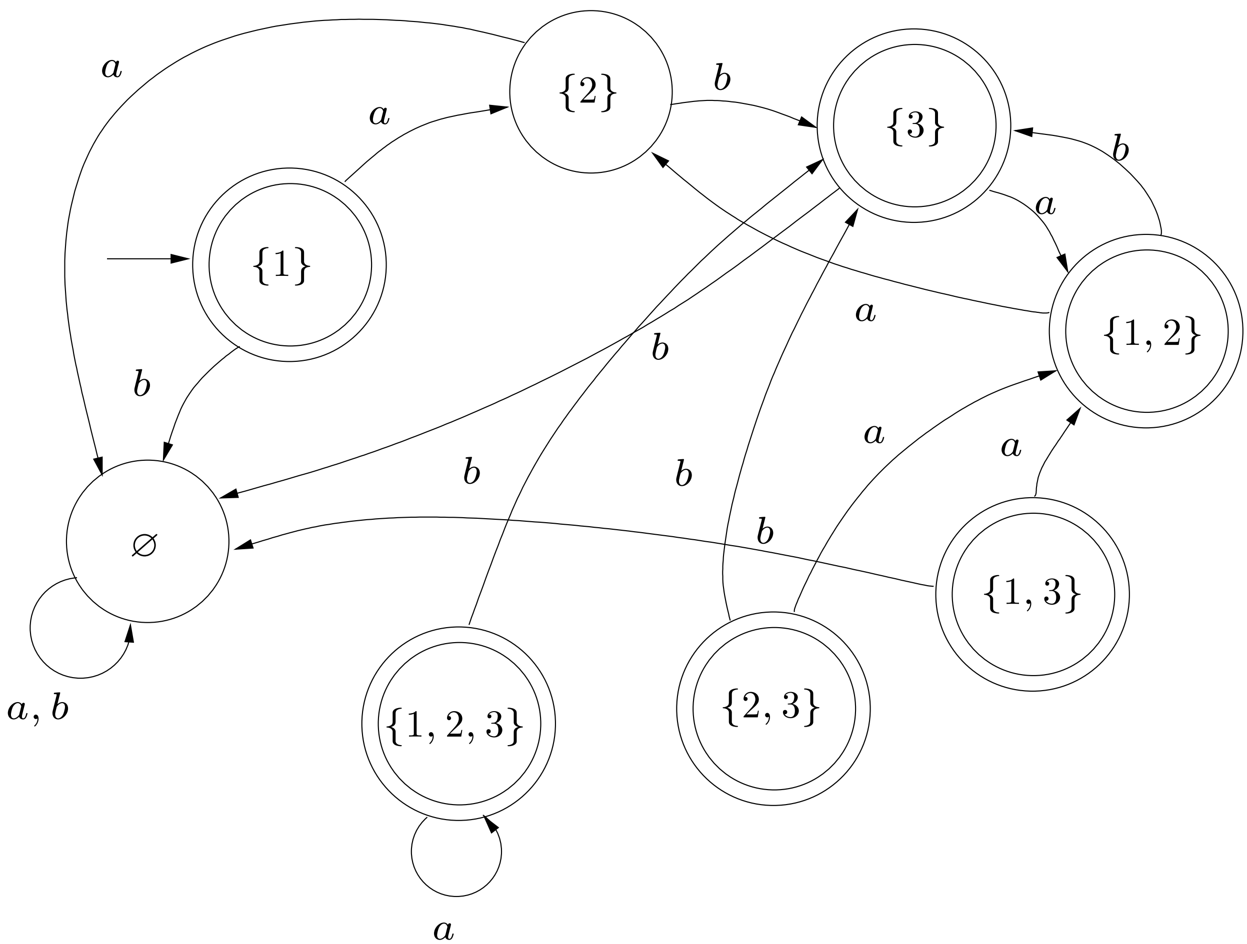
ohne ε-Übergänge



Neue Zustandsmenge:

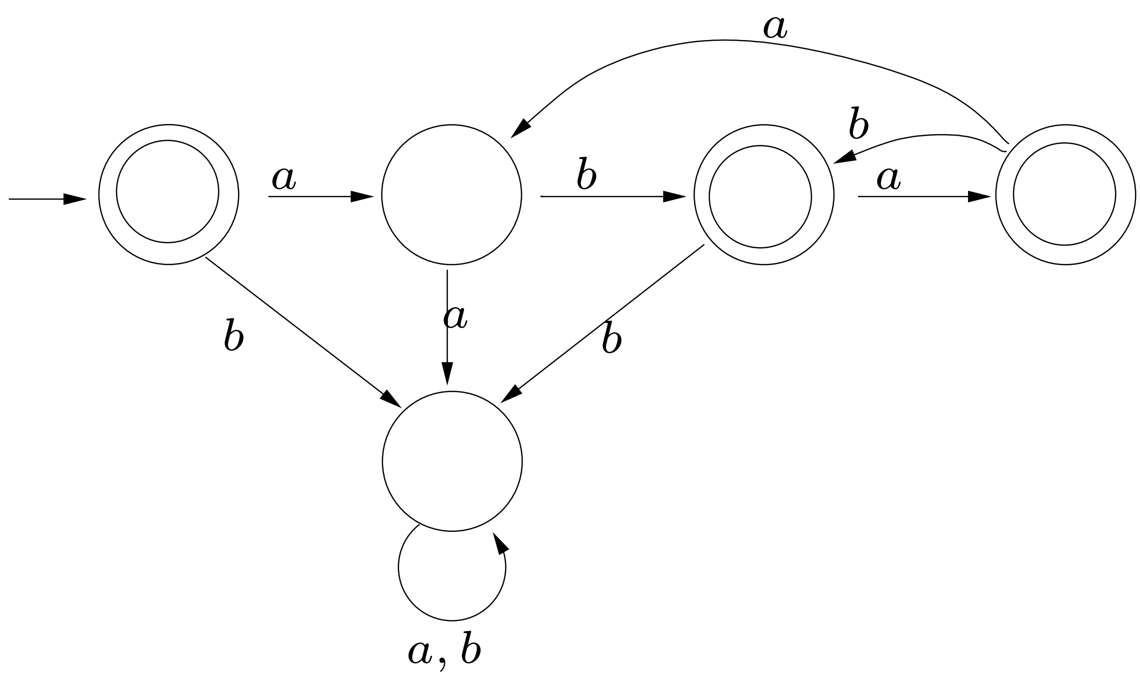
$\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$

Beispiele (Fort.)



Konstruktion liefert oft zu viele Zustände. Nicht erreichbare Zustände (vom Startzustand aus) streichen.

Beispiele (Fort.)



Ist dies minimaler DEA der $L(A)$ akzeptiert, d. h. minimale Anzahl von Zuständen? JA.

$x \underset{A}{\sim} y$ gdw $(q_0x \models_A q$ gdw $q_0y \underset{A}{\vdash} q)$.

$\underset{A}{\sim}$ ist rechtsinvariant, d.h.

$x \underset{A}{\sim} y \rightarrow xz \sim yz$ für alle $z \in \Sigma^*$.

Index $=$ Anzahl der Äquivalenzklassen.

$L(A)$ ist Vereinigung von Äquivalenzklassen (Myhill-Nerode).

Folgerungen

7.22 Folgerung

a) Rechts-lineare Sprachen sind abgeschlossen gegenüber Komplement und Durchschnitt.
$A = (Q, \Sigma, \Pi, q_0, F)$ DEA $L = L(A)$.
$A' = (Q, \Sigma, \Pi, q_0, Q - F)$ DEA mit $L(A') = \neg L$.
$L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}$ oder direkt mit Produktautomaten.
$A_1 \times A_2 = (Q_1 \times Q_2, \Sigma, \Pi_1 \times \Pi_2, (q_{0_1}, q_{0_2}), F_1 \times F_2)$.

b) Jede Typ-3 Sprache kann von Typ-3 Grammatik G erzeugt werden mit: Π enthält für $X \in N, a \in \Sigma$ $X \to aY$ oder $X \to a$ (genau eine Produktion $X \to aY$). D. h. G ist eindeutig und somit ist jede Typ-3 Sprache eindeutig.

c) Das WP für Typ-3 Grammatiken ist in linearer Zeit entscheidbar.

d) **Pumping-Lemma** für Typ-3 Sprachen.
Zu jeder Typ-3 Sprache L gibt es ein $n \in \mathbb{N}$, so dass für alle $y \in L$ gilt: Ist $|y| \geq n$. Dann lässt sich y zerlegen in $y = uvw$ mit $0 < |v| \leq n$, so dass für alle $i \in \mathbb{N}$ $uv^iw \in L$.
Beweis:
Sei A DEA mit $L(A) = L$ und $n := |Q|$. Ist $y \in L(A)$, $|y| \geq n$. Betrachte
$q_0y \overset{1}{\vdash} q_1y_1 \vdash \cdots \overset{1}{\vdash} q_{n-1}y_{n-1} \overset{1}{\vdash} q_ny_n \vdash \cdots \overset{1}{\vdash} q \in F$, $\{q_0, dots, q_n\} \subseteq Q$. Es gibt Zustand q', der zweimal vorkommt $q_0uvw \underset{A}{\vdash} q'vw \underset{A}{\vdash} q'w \vdash q_0, |uv| \leq n$. Dann aber $q_0uv^iw \vdash q$ für alle $i \geq 0$.

Beispiel

7.23 Beispiel

$L = \{w \in \{a,b\}^* : |w|_a = |w|_b\}$ nicht Typ 3 Sprache.

Angenommen, L ist rechts-linear, sei n Konstante für L.
Betrachte $y = a^n b^n \in L$
Pumping-Lemma $\rightsquigarrow a^{k_0}(a^k)^i a^{k_1} b^n \in L$ für alle i $(k_0 + k + k_1 = n, k > 0)$ $\lightning$

Oder: $L \cap \{a\}^*\{b\}^* = \{a^n b^n \mid n \geq 0\}$ wäre rechts-linear, falls L es ist. $\lightning$

e) Für eine Typ-3 Sprache sind folgende Probleme entscheidbar. Dabei soll L durch eine Typ-3 Grammatik, oder durch ein DEA, oder durch ein NEA gegeben sein.

- Ist L leer.
- Ist $L = \Sigma^*$.
- Ist L endlich.
- Ist $L = L_1$ für eine Typ-3 Sprache L_1.