Einführung in die Computerlinguistik

Endliche Automaten,
rechtslineare Grammatiken
und reguläre Sprachen (2)

Formale Grammatik

Definition 1. *Eine* **formale Grammatik** *ist ein 4-Tupel* $G = (N, T, S, P)$ *aus*

- *einem Alphabet von Terminalsymbolen* T *(statt* T *werden wir häufig wie bisher* Σ *schreiben)*
- *einem Alphabet von Nichtterminalsymbolen* N *mit* $N \cap T = \emptyset$
- *einem Startsymbol* $S \in N$
- *einer Menge von Regeln/Produktionen*
  $P \subseteq \{\langle \alpha_i, \beta_i \rangle \mid \alpha_i, \beta_i \in (N \cup T)^* \text{ und } \alpha_i \notin T^*\}$.

Für eine Regel $\langle \alpha_i, \beta_i \rangle$ *schreiben wir auch* $\alpha_i \to \beta_i$. *Die Bedingung* $\alpha_i \notin T^*$ *garantiert, daß die linke Seite einer Regel mindestens ein Nichtterminalsymbol enthält; dies ist nötig, damit die Terminalsymbole eine Ableitung terminieren.*

Diese Grammatik heißt auch **Typ0**- oder **allgemeine Regelgrammatik**.

Vokabular der formalen Grammatiken

Sei $G = (N, \Sigma, S, P)$ eine Grammatik und seien $v, w \in (\Sigma \cup N)^*$:

dann heißt v aus w **direkt ableitbar** (in Zeichen $w \rightarrow v$), wenn es Zerlegungen $w = w_1 \alpha w_2$ und $v = v_1 \beta v_2$ gibt, so daß $\langle \alpha, \beta \rangle \in P$.

dann heißt v aus w **ableitbar** (in Zeichen $w \rightarrow^* v$), wenn es $w_0, w_1, \ldots w_k \in (\Sigma \cup N)^*$ gibt $(k \geq 0)$, so daß $w = w_0$, $w_k = v$ und $w_{i-1} \rightarrow w_i$ für jedes $k \geq i \geq 0$. ($\rightarrow^*$ ist die reflexive transitive Hülle von $\rightarrow$)

$L(G) = \{w \in \Sigma^* | S \rightarrow^* w\}$ ist die **von der Grammatik** $G = (N, \Sigma, S, P)$ **erzeugte Sprache**.

Zwei Grammatiken G_i und G_j, die dieselbe Sprache erzeugen $(L(G_i) = L(G_j))$ nennt man **schwach äquivalent**.

Typ 3-Sprachen: rechts- / links-lineare Grammatiken und Sprachen

Definition 2. *Eine Grammatik (N, T, S, P) heißt* **rechts-linear** *(**links-linear**), wenn alle Regeln/Produktionen die folgende Form haben:*

$$A \to a \text{ oder } A \to aB \text{ (bzw. } A \to Ba), \text{ wobei } a \in T^* \text{ und } A, B \in N.$$

Eine durch eine rechts- oder links-lineare Grammatik erzeugte Sprache heißt rechts-/links-linear.

(Vorsicht: Nicht jede lineare Sprache ist rechts-linear)

Endliche Automaten und rechts-lineare Grammatiken

Zu jeder Sprache, die von einem endlichen Automaten akzeptiert wird, gibt es eine rechts-lineare Grammatik, die diese Sprache erzeugt und umgekehrt.

A= $\langle \Phi, \Sigma, \delta, S, F \rangle$ (endlicher Automat)

G= (N, T, S, P) (rechts-lineare Grammatik) mit:

$$N = \Phi$$

$$P = \{ \quad q \to ap \mid q, p \in \Phi \text{ und } a \in \Sigma \text{ und } \delta(q, a) = p \} \cup \{ q \to \epsilon \mid q \in F \}$$

Sprachbeschreibung durch reguläre Ausdrücke

Die Menge der **regulären Ausdrücke** $\mathbf{RA}_\Sigma$ über einem Alphabet $\Sigma = \{a_1, \ldots, a_n\}$ ist definiert durch:

- $\underline{\emptyset}$ ist ein regulärer Ausdruck
- ϵ ist ein regulärer Ausdruck
- $a_1, \ldots, a_n$ sind reguläre Ausdrücke
- Wenn a und b reguläre Ausdrücke über Σ sind, dann auch:
  - $(a + b)$
  - $(a \bullet b)$
  - $(a^\star)$

Die Klammern werden häufig weggelassen, wobei folgende Vorrangregel gilt:

$\star$ geht vor $\bullet$ geht vor $+$.

Semantik regulärer Ausdrücke / reguläre Sprachen

Jedem regulärem Ausdruck $r \in \mathbf{RA}_{\Sigma}$ über einem Alphabet Σ wird eine formale Sprache $L(r) \subseteq \Sigma^*$ als Bedeutung zugewiesen. Diese Sprachen heißen **reguläre Sprachen**.

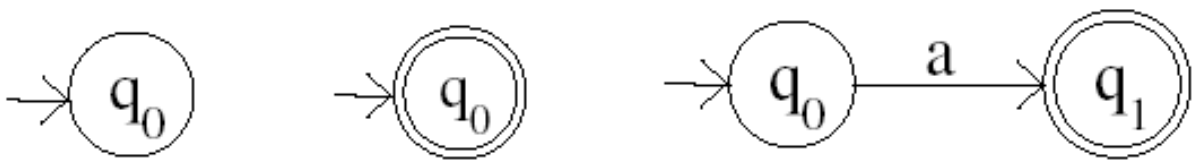
$L : \mathbf{RA}_{\Sigma} \to \mathcal{POT}(\Sigma^*)$ wird induktiv über den Aufbau der regulären Ausdrücke definiert:

- $L(\underline{\emptyset}) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a_i) = \{a_i\}$
- $L(a + b) = L(a) \cup L(b)$
- $L(a \bullet b) = L(a) \circ L(b)$
- $L(a^\star) = L(a)^*$

Reguläre Sprachen und endliche Automaten

[Satz von Kleene] *Jede von einem DEA akzeptierte Sprache ist regulär und zu jeder regulären Sprache gibt es einen DEA, der sie akzeptiert.*

Beweisskizze: jede reguläre Sprache wird von einem NDEA akzeptiert.



Abschlußeigenschaften regulärer Sprachen

Vereinigung	+ ✓
Schnittmenge	+
Komplementmenge	+
Konkatenation	+ ✓
Kleene Stern	+ ✓
Schnittmenge mit einer regulären Sprache	+

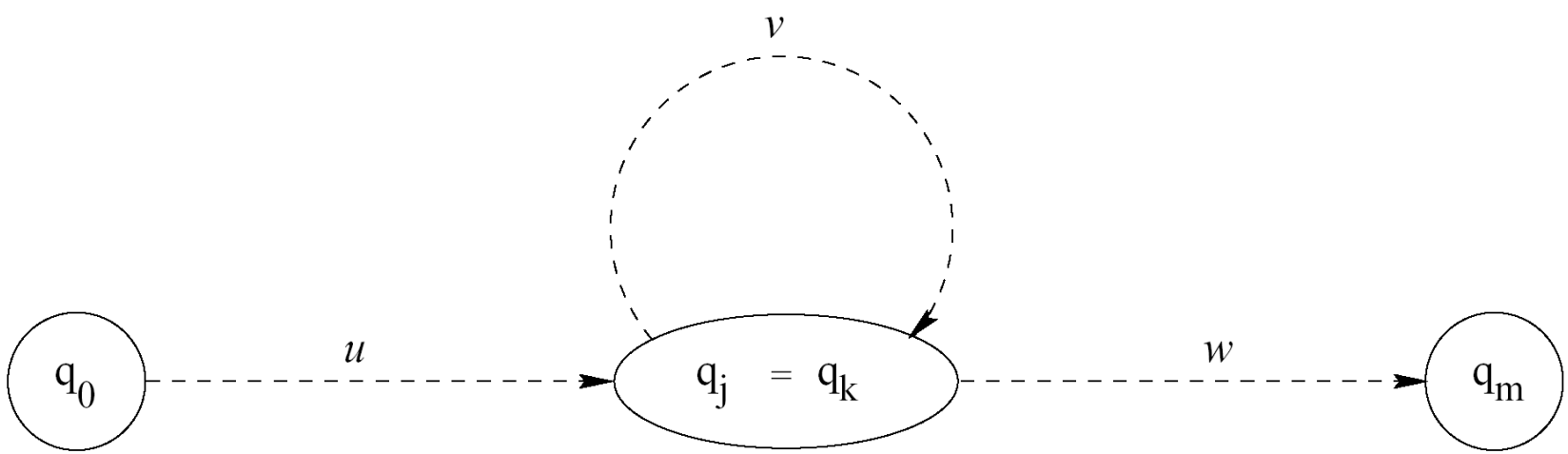
Komplement: konstruiere komplementären DEA

Schnittmenge: folgt aus *de Morgan*

Pumping-Lemma für reguläre Sprachen

Lemma 9. [Pumping-Lemma] *Wenn L eine unendliche reguläre Sprache ist, dann gibt es Wörter $u, v, w \in \Sigma^*$, so daß $v \neq \epsilon$ und $uv^i w \in L$ für beliebiges $i \geq 0$.*

Beweisskizze:



$L = \{a^n b^n : n \geq 0\}$ ist nicht regulär

$L = \{a^n b^n : n \geq 0\}$:

L ist unendlich. Wäre L regulär, dann gäbe es $u, v, w \in \{a, b\}^*$, $v \neq \epsilon$ mit $uv^n w \in L$ für alle $n \geq 0$. Für v ergeben sich folgende Fälle:

- angenommen v besteht aus a's und b's, dann gibt es in v^n $(n \geq 2)$ b's, die vor a's stehen (Widerspruch),
- angenommen v besteht nur aus a's, dann sind alle b's in dem w-Teil; wenn man v aufpumpt, steigt die Zahl der a's ohne daß die der b's steigen würde (Widerspruch).
- angenommen v besteht nur aus b's, dann analog zu v besteht nur aus a's,

Natürliche Sprachen sind nicht regulär (1)

1. Der Hund starb.
2. Der Hund, der den Vogel jagte, starb.
3. Der Hund, der den Vogel, der den Wurm fraß, jagte, starb.
4. Der Hund, der den Vogel, der den Wurm, der den Rasen durchquerte, fraß, jagte, starb.
5. Der Hund, der den Vogel, der den Wurm, der den Rasen, der den Garten bedeckte, durchquerte, fraß, jagte, starb.

   ...

- Allgemeine Form: der Hund (der den *maskulines Nomen*)^n (*transitives Verb*)^n starb.

Natürliche Sprachen sind nicht regulär (2)

- Sei $A = \left\{ \begin{array}{c} \text{der den Hund, der den Vogel, der den Kühlschrank,} \\ \text{der den Wurm, der den Rasen, der den Garten} \end{array} \right\}$ und
- $B = \{$fraß,beschenkte,durchquerte, jagte, liebte, sah,trank$\}$ und
- $w =$ *der Hund* und $v =$ *starb*.
- wx^*y^*v mit $x \in A$ und $y \in B$ ist eine reguläre Sprache.
- $\textsc{Deutsch} \cap wx^*y^*v = wx^ny^nv$.
- Wäre $\textsc{Deutsch}$ regulär, dann müßte auch wx^ny^nv regulär sein, da die Schnittmenge zweier regulärer Sprachen regulär ist (Widerspruch).

intuitive Merkregeln über reguläre Sprachen

- L ist regulär, wenn der Test auf Mitgliedschaft eines Wortes w in L durch buchstabenweises Durchlesen unter Zuhilfenahme eines beschränkten Speichers möglich ist.

- Endliche Automaten sind zu schwach für:
  - Worteigenschaften, deren Überprüfung eine Zählervariable mit Wertebereich $\mathbb{N}$ erfordert
  - Spezifikation der Wiederholung fester Muster beliebiger Länge
  - Mengen von Ausdrücken mit beliebiger Klammertiefe

Übungsaufgaben

Welche Aussagen kann man mit Hilfe der Abschlußeigenschaften der regulären Sprachen und dem Pumping-Lemma über die Komplexität folgender formaler Sprachen machen:

1. $L_1 = \{w \in \{a,b\}^* : w$ enthält eine ungerade Anzahl von $b's\}$.

2. $L_2 = \{w \in \{a,b\}^* : w$ enthält die gleiche Anzahl von $b's$ und $a's\}$.

3. w^R ist das Wort w in umgekehrter Reihenfolge.
   $L_3 = \{ww^R : w \in \{a,b\}^*\}$.

zu den Hausaufgaben

Welche Aussagen kann man mit Hilfe der Abschlußeigenschaften der regulären Sprachen und dem Pumping-Lemma über die Komplexität folgender formaler Sprachen machen:

1. $L_1 = \{w \in \{a,b\}^* : w$ enthält eine ungerade Anzahl von $b's\}$.

   L_1 ist regulär (L_1 akzeptierenden Automaten / L_1 erzeugende rechts-lineare Grammatik / regulärer Ausdruck)

2. $L_2 = \{w \in \{a,b\}^* : w$ enthält die gleiche Anzahl von $b's$ und $a's\}$.

   L_2 ist nicht regulär (Schnitt mit regulärer Sprache und Pumping-Lemma), obwohl für alle $v \in L_2$ auch $v^i \in L_2$

3. w^R ist das Wort w in umgekehrter Reihenfolge.
   $L_3 = \{ww^R : w \in \{a,b\}^*\}$.

   L_3 ist nicht regulär (Schnitt mit regulärer Sprache und Pumping-Lemma)