Grammatiken, Definitionen

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Eine **Grammatik** G (vom Typ **Chomsky-0, Rewritingsystem**, **Semi-Thue-System**) ist beschrieben durch ein 4-Tupel (V, Σ, P, S).

- V ist endliches Alphabet von **Variablen**
- Σ ist endliches Alphabet von **Terminalen**
- $S \in V$ ist das **Startsymbol**
- $P \subseteq ((V \cup \Sigma)^+ - \Sigma^*) \times (V \cup \Sigma)^*$ ist eine endliche Menge von **Produktionen** oder **(Ersetzungs-) Regeln**

Für eine Regel $(u,v) \in P$ schreiben wir $u \rightarrow v$.

Die Chomsky-Hierarchie

- Eine Grammatik heißt **kontextsensitiv** oder vom Typ **Chomsky-1**, falls für jede Regel $u \rightarrow v$ gilt: $|u| \leq |v|$.
- Sie heißt **kontextfrei** oder vom Typ **Chomsky-2**, falls alle Regeln vom Typ $u \rightarrow v$ mit $u \in V$ sind.
- Sie heißt **regulär** oder vom Typ **Chomsky-3**, falls alle Regeln vom Typ $u \rightarrow v$ mit $u \in V$ und

  $v \in \{\varepsilon\} \cup \Sigma \cup (\Sigma\ X\ \ V)$ sind.

(Eine allgemeine Grammatik ist vom Typ **Chomsky-0** .)

Satz: L ist genau dann rekursiv aufzählbar, wenn es Chomsky-0-Grammatik G gibt mit L = L (G).

Satz: Kontextsensitive Sprachen sind entscheidbar.

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Sei L regulär. Dann gibt es ein $p \in \mathbb{N}$, so dass für alle $x \in L$ mit $|x| \geq p$ eine Zerlegung $x = uvw$ existiert mit:

(i) $|uv| \leq p$,

(ii) $|v| \geq 1$, und

(iii) $uv^iw \in L$ für alle $i \geq 0$

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Wir wenden die Kontraposition des Pumping Lemmas an:

Falls für jedes $p \in \mathbb{N}$ ein $x \in L$ mit $|x| \geq p$ existiert, so dass für jede Zerlegung $x = uvw$ mit $|uv| \leq p$ und $|v| \geq 1$ ein $i \geq 0$ mit $uv^iw \notin L$ existiert,

so ist L nicht regulär.

Folgende Sprachen sind nicht regulär

$L_1 = \{0^n\, 1^n, n \geq 1\}$

$L_2 = \{ww \mid w \in \{0,1\}^*\}$

$L_3 = \{0^q \mid q \text{ Primzahl}\}$

Korollar: Sei REG die Klasse der regulären, CF die Klasse der kontextfreien Sprachen, dann gilt: REG $\subseteq$ CF und REG $\neq$ CF

Bew: „$\subseteq$“ ist klar. „$\neq$“ gilt, da $L_1 \in$ CF, aber $L_1 \notin$ REG ist.

Wir zeigen:

DFAs können genau die Sprachen akzeptieren,

Die reguläre Grammatiken erzeugen können.

D.h.: Endliche Automaten akzeptieren genau die regulären Sprachen.

Endliche Automaten

… sind Turingmaschinen, die nicht schreiben dürfen
(und nur nach rechts gehen)

Endliche Automaten haben also, neben der Eingabe, nur konstant viel Speicher (Speichern in den Zuständen)

Ein **(deterministischer) endlicher Automat (DFA)** A ist beschrieben durch ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, wobei

- Q eine endliche Menge von Zuständen ist.
- Σ ein endliches Alphabet ist, $Q \cap \Sigma = \emptyset$.
- $\delta : Q \times \Sigma \to Q$ die Übergangsfunktion ist.
- q_0 der Startzustand ist.
- $F \subseteq Q$ die Menge von akzeptierenden Zuständen ist.

Akzeptierte Sprachen

Ein DFA **A akzeptiert w** $\in \Sigma^*$ genau dann,
wenn δ (q_0, w) $\in$ F gilt.

Die **von A akzeptierte Sprache** ist
L (A) := {w $\in \Sigma^*$, δ (q_0, w) $\in$ F}

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Wir zeigen:

DFAs können genau die Sprachen akzeptieren, die reguläre Grammatiken erzeugen können.

Bew.: „$\Rightarrow$“ Sei $A = (Q, \Sigma, \delta, q_0, F)$ DFA für $L \subseteq \Sigma^*$.

Wir definieren $G = (V{:=}Q, \Sigma, P, S{:=}q_0)$

Für $\delta(q,a)=q'$ liegt $q \rightarrow aq'$ in P

Falls $q' \in F$, so liegt $q \rightarrow a$ in P

„$\Leftarrow$“ ?

Nichtdeterministische endliche Automaten (NFA)

Definition 1 *Ein* **nichtdeterministischer endlicher Automat (NFA)** N *ist beschrieben durch ein 5-Tupel* $(Q, \Sigma, \delta, q_0, F)$*, wobei*

- Q *eine endliche Menge von Zuständen ist.*
- Σ *ein endliches Alphabet ist,* $Q \cap \Sigma = \emptyset$*.*
- $\delta : Q \times \Sigma_\epsilon \to 2^Q$ *die Übergangsfunktion ist.*
- q_0 *der Startzustand ist.*
- $F \subseteq Q$ *die Menge von akzeptierenden Zuständen ist.*

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Der NFA N_1:

$Q = \{q_0, q_1, q_2, q_3, q_4\}$,

$\Sigma = \{0,1\}$, $F = \{q_3\}$.

δ	0	1	ϵ
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\{q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\{q_3\}$	$\emptyset$
q_3	$\{q_4\}$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

Arbeitsweise eines NFA

Um exakt zu definieren, welche Sprache ein NFA akzeptiert, erweitern wir wie bei DFAs die Definition von δ von $Q \times \Sigma_\epsilon$ auf $Q \times \Sigma_\epsilon^*$. Wir setzen für $w = w_1 \cdots w_n \in \Sigma_\epsilon^*$

$$\begin{aligned}\delta(q,w) &= \delta(q, w_1 \cdots w_n)\\ &= \{r \in Q \mid \text{es gibt ein } q' \in \delta(q, w_1 \cdots w_{n-1}) \text{ mit } r \in \delta(q', w_n)\}\\ &= \bigcup_{q' \in \delta(q, w_1 \cdots w_{n-1})} \delta(q', w_n).\end{aligned}$$

Ein NFA N *akzeptiert* $w \in \Sigma^*$, wenn

1. $w = w_1 \cdots w_n,\ w_i \in \Sigma_\epsilon^*$
2. $\delta(q_0, w_1 \cdots w_n) \cap F \neq \emptyset$.

Die von einem NFA N akzeptierte Sprache ist $L(N) := \{w \in \sum^* \mid \delta(q_0, w) \cap F \neq 0\}$.

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Der NFA N_2:

$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $F = \{q_3\}$

δ	0	1	ϵ
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\{q_3\}$	$\emptyset$
q_3	$\{q_3\}$	$\{q_3\}$	$\emptyset$

Satz: Sei L $\subseteq \Sigma^*$. Folgende Aussagen sind äquivalent:

(i) L ist regulär
(ii) L wird von einer regulären Grammatik erzeugt
(iii) L wird von einem DFA akzeptiert
(iv) L wird von einem NFA akzeptiert.

Satz 3.3.3 Sei N ein NFA und L die von N akzept. Sprache. Dann gibt es auch einen DFA, der L akzeptiert.

Bew.: L werde von NFA N = $(Q, \Sigma, \delta, q_0, F)$ akzeptiert.
Folgender DFA A = $(Q', \Sigma, \delta', q_0', F')$ akzeptiert L.

Potenzmengenkonstruktion:

$Q' = 2^Q$, $q_0' = \{q_0\}$, $F' = \{R \in Q', R \cap F \neq \emptyset\}$

$\delta'(R, a) = \cup_{r \in R}\, \delta(r,a)$ $(= \{q \in Q \mid \exists\, r \in R$ mit $q \in \delta(r,a)\})$

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Der NFA N_3:

$Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$

δ	0	1	ϵ
q_0	$\emptyset$	$\{q_1\}$	$\emptyset$
q_1	$\{q_1, q_2\}$	$\{q_2\}$	$\emptyset$
q_2	$\{q_1\}$	$\emptyset$	$\emptyset$

Satz 3.3.2 L werde von regulärer Grammatik G erzeugt. Dann gibt es einen NFA N, der L akzeptiert.

Bew.: Sei $G=(V,\Sigma,P,S)$ reguläre Grammatik für L.
Folgender NFA $N=(Q,\Sigma,\delta,q0,F)$ akzeptiert L.

$Q = V \cup \{q_f\}$ mit $q_f \notin V$ (neuer Zustand), $q_0 = S$,

Falls $u \rightarrow av \in P$, dann $v \in \delta(u,a)$ $(u,v \in V,\ a \in \Sigma)$

Falls $u \rightarrow a \in P$, dann $q_f \in \delta(u,a)$ $(u \in V,\ a \in \Sigma)$

Falls $u \rightarrow \varepsilon \in P$, dann $u \in F$ $(u \in V)$

$q_f \in F$

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität

Definition 1 *Sei Σ ein beliebiges endliches Alphabet. R ist ein regulärer Ausdruck über Σ, wenn R einer der folgenden Ausdrücke ist*

1. a, *wobei* $a \in \Sigma$
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$, *wobei* R_1, R_2 *reguläre Ausdrücke sind.*
5. $(R_1 \cdot R_2)$, *wobei* R_1, R_2 *reguläre Ausdrücke sind.*
6. (R_1^*), *wobei* R_1 *ein regulärer Ausdruck ist.*

Reguläre Ausdrücke

Die von R beschriebene Sprache L(R):

1. Ist $R = a, a \in \Sigma$, so ist $L(R) = \{a\}$.

2. Ist $R = \epsilon$, so ist $L(R) = \{\epsilon\}$.

3. Ist $R = \emptyset$, so ist $L(R) = \emptyset$.

4. Ist $R = (R_1 \cup R_2)$, so ist $L(R) = L(R_1) \cup L(R_2)$.

5. Ist $R = (R_1 \cdot R_2)$, so ist $L(R) = L(R_1) \cdot L(R_2) = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2$.

6. Ist $R = (R_1^*)$, so ist $L(R) = L(R_1)^*$.

Satz: L ist genau dann regulär, wenn es regulären Ausdruck R
Mit L (R) = L gibt.

Satz: Sei $L \subseteq \Sigma^*$. Folgende Aussagen sind äquivalent:

(i) L ist regulär

(ii) L wird von einer regulären Grammatik erzeugt

(iii) L wird von einem DFA akzeptiert

(iv) L wird von einem NFA akzeptiert.

(v) L ist durch regulären Ausdruck beschreibbar.

(vi) L wird von Zweiwegeautomat (NFA, der auch nach links laufen darf) akzeptiert.

Abschlusseigenschaften regulärer Sprachen

Reguläre Sprachen sind abgeschlossen unter

- Sternbildung
- Komplement
- Konkatenation
- Durchschnitt
- Vereinigung