Kapitel: Die Chomsky Hierarchie

Allgemeine Grammatiken

Definition

Eine Grammatik $G = (\Sigma, V, S, P)$ besteht aus:

- einem endlichen Alphabet Σ,
- einer endlichen Menge V von Variablen (oder Nichtterminalen) mit $\Sigma \cap V = \emptyset$,
- dem Startsymbol $S \in V$ und
- einer endlichen Menge P von Produktionen der Form $u \to v$ mit

$$u \in (\Sigma \cup V)^* \, V \, (\Sigma \cup V)^* \quad \text{und} \quad v \in (\Sigma \cup V)^*$$

Eine Grammatik G erzeugt eine Sprache $L(G) \subseteq \Sigma^*$ wie folgt:

Beginne mit dem Startsymbol S und wende dann Produktionen an:

Eine Produktion $u \to v$ ersetzt ein Vorkommen von u durch ein Vorkommen von v.

Details: nächste Folie

Die von einer Grammatik erzeugte Sprache

Sei $G = (\Sigma, V, S, P)$ eine Grammatik.

- Für eine Produktion $u \to v$ und ein Wort $w_1 = xuy$ wird das Wort $w_2 = xvy$ abgeleitet. Wir schreiben
$$xuy \Rightarrow xvy.$$

- Für Worte $r, s \in (\Sigma \cup V)^*$ schreiben wir
$$r \overset{*}{\Rightarrow} s \quad :\Longleftrightarrow \quad \text{Es gibt Worte } w_1 = r,\ w_2,\ \ldots,\ w_k = s, \text{ für } k \geqslant 1, \text{ so dass } w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_k.$$
s kann in null, einem oder mehreren Schritten aus r abgeleitet werden.

- $L(G) = \left\{ w \in \Sigma^* \ : \ S \overset{*}{\Rightarrow} w \right\}$ ist die von der Grammatik G erzeugte Sprache.

Beispiel: Eine Grammatik, die die Sprache $L := \{a^n b^n c^n : n \in \mathbb{N}_{>0}\}$ erzeugt

Wir wissen bereits, dass diese Sprache nicht kontextfrei ist — somit haben wir keine Chance, eine kontextfreie Grammatik zu finden, die L erzeugt.

Eine nicht-kontextfreie Grammatik $G = (\Sigma, V, S, P)$ mit $L(G) = L$:

- $\Sigma = \{a, b, c\}$
- $V = \{S, B, C\}$
- Startsymbol S
- P besteht aus folgenden Produktionen:

$$\begin{aligned} S &\to aSBC \mid aBC \\ CB &\to BC \\ aB &\to ab \\ bB &\to bb \\ bC &\to bc \\ cC &\to cc. \end{aligned}$$

Behauptung: $L(G) = \{a^n b^n c^n : n \in \mathbb{N}_{>0}\}$.

Beweis: siehe Tafel.

Die Chomsky Hierarchie (1/2)

(0) Allgemeine Grammatiken, wie sie am Anfang dieses Kapitels definiert wurden, heißen Typ 0-Grammatiken.

(1) Eine Grammatik $G = (\Sigma, V, S, P)$ heißt monoton oder Typ 1-Grammatik, falls für jede Produktion $(u \to v) \in P$ gilt: $|u| \leqslant |v|$.

Beispiel: Die Grammatik für $\{a^n b^n c^n : n \geqslant 1\}$ der vorherigen Folie ist monoton.

Eine Grammatik $G = (\Sigma, V, S, P)$ heißt kontextsensitiv, falls jede Produktion in P von der Form $uXv \to uxv$ ist, mit $X \in V$, $u, x, v \in (\Sigma \cup V)^*$ und $x \neq \varepsilon$.

Beachte: Per Definition ist jede kontextsensitive Grammatik ist monoton. Umgekehrt kann man zeigen (hier ohne Beweis), dass es für jede monotone Grammatik eine kontextsensitive Grammatik gibt, die dieselbe Sprache erzeugt.

In der Literatur wird daher der Begriff "kontextsensitive Grammatik" oft als Synonym für "monotone Grammatik" verwendet.

Sonderfall: Erzeugen des leeren Worts
Per Definition gilt für monotone Grammatiken G, dass $\varepsilon \notin L(G)$.
Um auch Sprachen erzeugen zu können, die das leere Wort enthalten, erlauben wir als Typ 1-Grammatiken auch Grammatiken der Form

$$(\Sigma,\ V \dot\cup \{S'\},\ S',\ P \cup \{S' \to \varepsilon \mid S\}),$$

wobei (Σ, V, S, P) eine monotone Grammatik ist.

Die Chomsky Hierarchie (2/2)

(2) Eine Grammatik $G = (\Sigma, V, S, P)$ heißt kontextfrei oder Typ 2-Grammatik, falls jede Produktion in P von der Form $X \to x$ ist, mit $X \in V$ und $x \in (\Sigma \cup V)^*$.

Wir wissen bereits, dass die Typ 2-Grammatiken genau die kontextfreien Sprachen erzeugen.

Beachte: Für jede Typ 2-Grammatik G kann man leicht eine Typ 1-Grammatik konstruieren, die dieselbe Sprache erzeugt:

1. Sei $V_\varepsilon = \{X \in V : X \overset{*}{\Rightarrow} \varepsilon\}$.
2. Für jedes $A \in V_\varepsilon$ tue Folgendes:
   Entferne aus P alle Produktionen der Form $A \to \varepsilon$ und
   füge für jede Produktion der Form $(B \to uAv) \in P$ mit $uv \neq \varepsilon$
   eine zusätzliche Produktion der Form $B \to uv$ ein.
3. Falls $S \in V_\varepsilon$, so wähle ein neues Startsymbol S' und füge die zusätzlichen Produktionen $S' \to S$ und $S' \to \varepsilon$ ein.

(3) Eine Grammatik $G = (V, \Sigma, S, P)$ heißt regulär oder Typ 3-Grammatik, falls jede Produktion in P von der Form $X \to \varepsilon$ oder $X \to aY$ mit $X, Y \in V$ und $a \in \Sigma$ ist.

Wir wissen bereits, dass die Typ 3-Grammatiken genau die regulären Sprachen erzeugen.

Die Sprachen der Chomsky Hierarchie

Für jedes $i \in \{0, 1, 2, 3\}$ sei
$\mathcal{L}_i$ die Klasse aller Sprachen, die von Typ i-Grammatiken erzeugt werden.

Es gilt:

- $\mathcal{L}_3$ ist die Klasse aller regulären Sprachen (kurz auch: REG oder Typ 3-Sprachen).
- $\mathcal{L}_2$ ist die Klasse aller kontextfreien Sprachen
  (kurz auch: CFL, für "context-free language", oder Typ 2-Sprachen).
- $\mathcal{L}_1$ wird die Klasse aller kontextsensitiven Sprachen genannt
  (kurz auch: CSL, für "context-sensitive language", oder Typ 1-Sprachen).
- $\mathcal{L}_0$ wird die Klasse aller Typ 0-Sprachen genannt.

Es gilt:

$$\mathcal{L}_0 \underbrace{\supsetneq}_{H \in \mathcal{L}_0 \setminus \mathcal{L}_1} \mathcal{L}_1 \underbrace{\supsetneq}_{\{a^n b^n c^n : n \geqslant 1\} \in \mathcal{L}_1 \setminus \mathcal{L}_2} \mathcal{L}_2 \underbrace{\supsetneq}_{\{a^n b^n : n \geqslant 1\} \in \mathcal{L}_2 \setminus \mathcal{L}_3} \mathcal{L}_3.$$

Charakterisierung der Typ 0- und der Typ 1-Sprachen

Frage:

- Was genau sind die Typ 0-Sprachen?
- Was genau sind die Typ 1-Sprachen?

Antwort:

Satz:

(a) Die Typ 0-Sprachen sind genau die semi-entscheidbaren Sprachen $L \subseteq \Sigma^*$.

(b) Die Typ 1-Sprachen sind genau die Sprachen $L \subseteq \Sigma^*$, die von einer nichtdeterministischen Turingmaschine mit linear beschränktem Platz entschieden werden können. Solche Turingmaschinen heißen auch linear beschränkte Automaten (kurz: LBA).

Beweis von (a):

- Jede Typ 0-Sprache ist semi-entscheidbar:

  Dies erhält man leicht durch einen Algorithmus, der bei Eingabe von $w \in \Sigma^*$ nach und nach sämtliche möglichen Ableitungen der Typ 0-Grammatik G durchprobiert und anhält, falls er eine Ableitung für w gefunden hat.

- Jede semi-entscheidbare Sprache wird von einer Typ 0-Grammatik erzeugt:

Sei T eine Turingmaschine, die $L \subseteq \Sigma^*$ semi-entscheidet.
Wir konstruieren eine Grammatik G mit $L(G) = L$.

- Berechnungen von T beginnen natürlich stets mit der Eingabe w, aber Ableitungen von w enden mit w.
- Also sollten wir die Grammatik G so konstruieren, dass die Berechnungen von T „rückwärts“ simuliert werden.

O.B.d.A gibt es nur einen Zustand q_h, in dem T anhält,
und wenn T hält, ist das Band leer.

Angenommen, wir unterbrechen die Berechnung von T auf Eingabe w zu einem beliebigen Zeitpunkt.

Wir beschreiben die aktuelle Konfiguration der TM genauso wie im Beweis der Unentscheidbarkeit des Postschen Korrespondenzproblems:

> Die Situation, in der T im Zustand q ist, die Bandbeschriftung $\alpha_1 \cdots \alpha_{i-1}\alpha_i \cdots \alpha_N$ hat und der Kopf auf Position i steht, beschreiben wir durch das Wort
>
> $$\alpha_1 \cdots \alpha_{i-1}\, q\, \alpha_i \cdots \alpha_N \quad \in \quad (\Gamma \cup Q)^*.$$

- Unsere Grammatik G erzeugt diese Beschreibungen als Zwischenschritte.
- G erzeugt zuerst die Endkonfiguration, in der T mit leerem Band anhält, indem sie die folgende Produktion nutzt:

$$S \rightarrow \square q_h \square$$

Typ 0-Sprachen und Semi-Entscheidbare Sprachen (3/3)

- Für jeden Befehl $\delta(q, a) = (q', b, \leftarrow)$ nehmen wir die Produktion

$$q'cb \to cqa$$

für alle $c \in \Gamma$ auf:

Wenn die Konfiguration $* \cdots * q'cb * \cdots *$ schon erzeugt wurde, können wir damit die mögliche Vorgänger-Konfiguration $* \cdots * cqa * \cdots *$ erzeugen.

- Für jeden Befehl $\delta(q, a) = (q', b, \to)$ nehmen wir die Produktion $bq' \to qa$ auf.

- Für jeden Befehl $\delta(q, a) = (q', b, \downarrow)$ nehmen wir die Produktion $q'b \to qa$ auf.

- Am Ende der Ableitung haben wir ein Wort der Form $\square^* q_0 w \square^*$ erzeugt.
- Jetzt lösche q_0 und alle $\square$-Symbole mit weiteren Produktionen.

Die dadurch entstehende Grammatik erzeugt genau die Worte, die von T akzeptiert werden. $\square$

Typ 1-Sprachen und linearer Speicherplatz

Satz:

$L \subseteq \Sigma^*$ ist genau dann kontextsensitiv, wenn $L \subseteq \Sigma^*$ von einer nichtdeterministischen Turingmaschinen auf linearem Speicherplatz entschieden wird.

Insbesondere ist jede kontextsensitive Sprache entscheidbar.

Beweis:

"$\Longrightarrow$": Es gelte $L = L(G)$, für eine Typ-1 Grammatik G.
Warum kann L von einer nichtdeterministischen Turingmaschine T auf linearem Platz akzeptiert werden?

- T "rät" eine Ableitung $S \stackrel{*}{\Rightarrow} w$ von w.
- Falls $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_\ell$ mit $u_\ell = w$ eine Ableitung von w ist, so ist $|u_i| \leqslant |u_{i+1}| \leqslant |w|$, für alle $i < \ell$, da G monoton ist.
- Daher kann T zu jedem Zeitpunkt i die Worte u_i, u_{i+1}, w auf seinem Band speichern.
- Insgesamt reicht linearar Speicherplatz $O(|w|)$ aus!

"$\Longleftarrow$": Übung (ähnlich wie der entsprechende Beweis für Typ 0-Sprachen). □

Zusammenfassung (1/2)

Die Chomsky-Hierarchie besteht aus folgenden Klassen von Sprachen:

- Typ 0-Sprachen, d.h. Sprachen, die von allgemeinen Grammatiken erzeugt werden.

  Dies sind genau die semi-entscheidbaren Sprachen.

- Typ 1-Sprachen, d.h. Sprachen, die von monotonen (oder kontextsensitiven) Grammatiken erzeugt werden.

  Dies sind genau die Sprachen, die von linear beschränkten Automaten (d.h. nichtdeterministische Turingmaschinen mit linear beschränktem Platz) entschieden werden.

- Typ 2-Sprachen, d.h. Sprachen, die von kontextfreien Grammatiken erzeugt werden.

  Dies sind genau die Sprachen, die von nichtdeterministischen Kellerautomaten akzeptiert werden.

- Typ 3-Sprachen, d.h. Sprachen, die von regulären Grammatiken erzeugt werden.

  Dies sind genau die Sprachen, die von endlichen Automaten akzeptiert werden.

Zusammenfassung (2/2)

Trennende Beispiele:

- Eine Typ 2-Sprache (kontextfrei), die keine Typ 3-Sprache (regulär) ist:

$$\{a^n b^n : n \geqslant 1\}.$$

- Eine Typ 1-Sprache (kontextsensitiv), die keine Typ 2-Sprache (kontextfrei) ist:

$$\{a^n b^n c^n : n \geqslant 1\}.$$

- Eine Typ 0-Sprache (semi-entscheidbar), die keine Typ 1-Sprache ist:

$$\text{Halteproblem} \quad H := \{\langle T \rangle w : T \text{ ist eine TM, die bei Eingabe } w \text{ hält}\} \subseteq \{0,1\}^*$$

- Eine Sprache, die keine Typ 0-Sprache ist:

$$\overline{H} := \{0,1\}^* \setminus H.$$