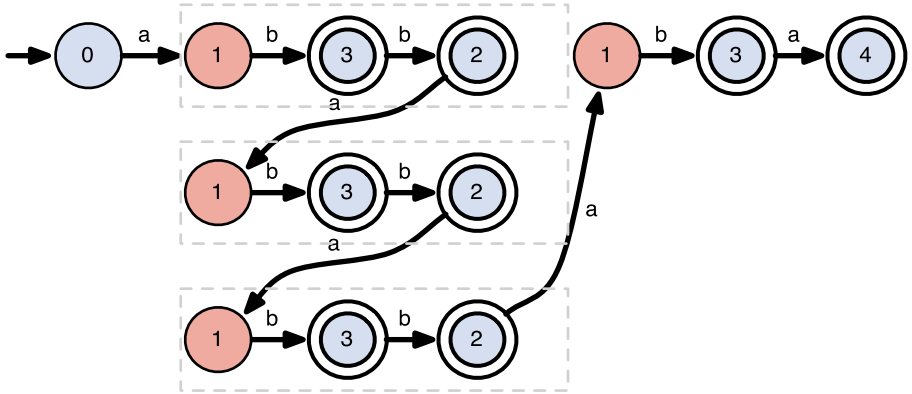
Berechenbarkeitstheorie

8. Vorlesung



Dr. Franziska Jahnke
Institut für Mathematische Logik und Grundlagenforschung
WWU Münster

Kontextfreie Grammatiken

- Grammatiken = Formalismus zum Beschreiben von Sprachen mit Ableitungsregeln
- Ziel: $\{\mathsf{a}^n\mathsf{b}^n \mid n \geq 0\}$ soll erkannt werden
- Ziel: Modell soll *beherrschbar* bleiben

Kontextfreie Grammatik

Eine Kontextfreie Grammatik G ist ein 4 Tupel (V, Σ, R, S), bestehend aus

- V endliche Menge von Variablen,
- $\Sigma \neq \emptyset$ endliche Menge von Terminalsymbolen,
- $R \subseteq V \times (V \cup \Sigma)^*$ einer endliche Menge von Regeln,
- $S \in V$ einem Startsymbol.

Sprache einer kf Grammatik

- wenn $u, v, w \in (\Sigma \cup V)^*$ und $A \to w$ eine Regel, dann leitet uAv zu uwv ab
  Schreibweise: $uAv \Rightarrow uwv$
- wiederholtes Ableiten: $uAv \Rightarrow \cdots \Rightarrow \cdots \Rightarrow t$
  Schreibweise: $uAv \Rightarrow^* t$

Definition

Für eine kf Grammatik G ist die Sprache der Grammatik $L(G)$ definiert als

$$L(G) := \{w \in \Sigma^* \mid S \Rightarrow^* w\}.$$

- also alle Wörter **über** Σ die ich aus dem Startsymbol ableiten kann

Definition

Die Sprache $L(G)$ einer kf. Grammatik G heißt **kontextfreie Sprache**.

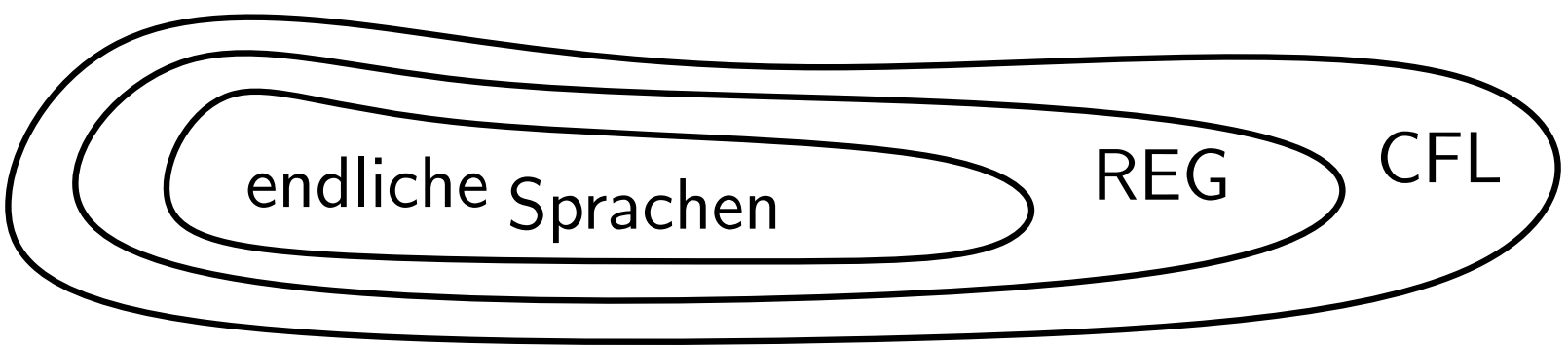
$$\text{CFL} := \{L \mid L \text{ ist kontextfrei}\}$$

Satz 10

$$L \in \text{REG} \Rightarrow L \in \text{CFL}$$

- Aus Satz 10 folgt, dass $\text{REG} \subsetneq \text{CFL}$ ...

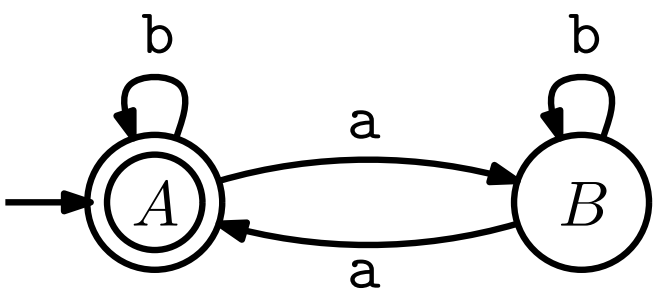
  ... denn $\{\mathsf{a}^n\mathsf{b}^n \mid n \geq 0\}$ ist kontextfrei aber nicht regulär



Beweis $L \in \text{REG} \Rightarrow L \in \text{CFL}$

- sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DEA, welcher L erkennt
- Definiere Grammatik $G_M = (V, \Sigma, R, S)$ mit
  - $V = Q$,
  - $S = q_0$,
  - für alle $q \in Q$ und $a \in \Sigma$ mit $\delta(q, a) = p$ füge die Regel $q \to ap$ zu R hinzu,
  - für alle $q \in F$ füge die Regel $q \to \varepsilon$ zu R hinzu.

Konstruktion am Beispiel



$$V = \{A, B\},\ \Sigma = \{\mathtt{a}, \mathtt{b}\},\ S = A$$

R: $A \to \mathtt{a}B \mid \mathtt{b}A \mid \varepsilon$

$B \to \mathtt{a}A \mid \mathtt{b}B$

- für alle $q \in Q$ und $a \in \Sigma$ mit $\delta(q,a) = p$ füge die Regel $q \to ap$ zu R hinzu,
- für alle $q \in F$ füge die Regel $q \to \varepsilon$ zu R hinzu.

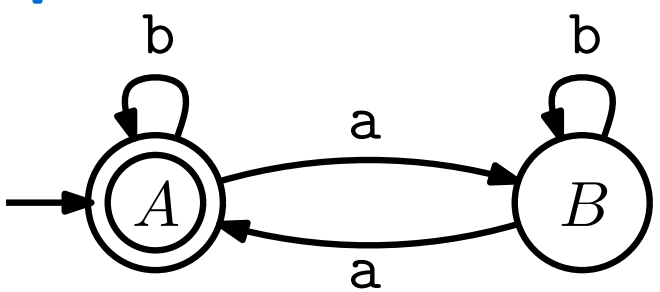
■ $w \in L \iff w$-Lauf $(p_0, p_1, \ldots, p_n)$ in M und $p_n \in F$
$w = x_1 x_2 \ldots$

(i) $p_0 = q_0$ (ii) $\delta(p_i, x_{i+1}) = p_{i+1}$

$$\iff (p_0, p_1, \ldots, p_n) \in V^n \text{ mit } p_n \to \varepsilon$$
$$p_0 = S \text{ und } p_i \to x_{i+1} p_{i+1}$$
$$\iff p_0 \Rightarrow x_1 p_1 \Rightarrow x_1 x_2 p_2 \Rightarrow \cdots \Rightarrow w p_n \Rightarrow w$$
$$\iff w \in L(G_M)$$

□

Bsp.



$$R: \quad \begin{array}{lcl} A & \to & \mathtt{a}B \mid \mathtt{b}A \mid \varepsilon \\ B & \to & \mathtt{a}A \mid \mathtt{b}B \end{array}$$

aba hat akz. Lauf (A, B, B, A) $\quad A \Rightarrow \mathtt{a}B \Rightarrow \mathtt{ab}B \Rightarrow \mathtt{aba}A \Rightarrow \mathtt{aba}$

Chomsky Normalform

Definition

Eine kontextfreie Grammatik G ist in **Chomsky Normalform (CNF)** wenn alle ihre Regeln von der folgenden Form sind:

1. $A \to BC$, wobei $B, C \in V$ und $B, C \neq S$
2. $A \to a$, wobei $a \in \Sigma$
3. $S \to \varepsilon$

Bsp.

$$\begin{aligned} S &\to BB \mid AA \mid \varepsilon \\ A &\to AB \mid BA \mid AA \mid \mathtt{a} \\ B &\to \mathtt{b} \end{aligned}$$

Satz 11

Für jede Sprache $L \in$ CFL gibt es eine kf. Grammatik G in CNF mit $L(G) = L$.

Beweis

- Sei $G = (V, \Sigma, R, S)$ eine kfG für L, dann konstruieren wir eine kfG $G' = (V', \Sigma, R', S')$, in CNF für L
- Konstruktion erfolgt in 5 Schritten durch Umbau der Regeln aus R

1. Neuer Startzustand

- Füge die Regel $S' \to S$ ein
- neuer Startzustand tritt auf keiner rechten Seite auf

2. Eliminiere $A \to \varepsilon$ Regeln

- so lange es Regeln $A \to \varepsilon$ gibt, wähle eine dieser Regeln aus
- streiche $A \to \varepsilon$
- modifiziere jede Regel mit A auf der rechten Seite so, als ob dass A auch gestrichen werden kann (alle Kombinationen!)

Bsp. $X \to \mathrm{a}ABA$

wird zu $X \to \mathrm{a}ABA \mid \mathrm{a}BA \mid \mathrm{a}AB \mid \mathrm{a}B$

2. Eliminiere $A \to \varepsilon$ Regeln

- Sonderfall: Regel $X \to A$
  - wird zu $X \to A \mid \varepsilon$ wenn Regel $X \to \varepsilon$ noch nicht eliminiert wurde
  - ansonsten bleibt $X \to A$ unverändert

3. Eliminiere $A \to B$ Regeln

- streiche $A \to B$
- für alle Regeln $B \to w$ mit $w \in (V \cup \Sigma)^*$ füge die Regel $A \to w$ hinzu
- Sonderfall 1: $B \to A$
  - $A \to A$ Regel nicht aufnehmen
- Sonderfall 2: $B \to X$ und $A \to X$ wurde bereits gelöscht
  - $A \to X$ Regel nicht aufnehmen

4. Kürze zu lange rechte Seiten

- Regel $A \to x_1 x_2 \dots x_k$, mit $x_i \in V \cup \Sigma$
- streiche $A \to x_1 x_2 \dots x_k$
- füge folgende Regeln hinzu:

$$\left.\begin{array}{rcl} A & \to & x_1 A_1 \\ A_1 & \to & x_2 A_2 \\ & \vdots & \\ A_{k-1} & \to & x_k \end{array}\right\} \; A_i \textbf{ neue} \text{ Variablen}$$

5. Ersetzte Regeln der Form $A \to xB$, $A \to Bx$ und $A \to xy$

- für jedes Terminal $x \in \Sigma$ füge folgende Regel ein:

$$V_x \to x \qquad\qquad V_x \textbf{ neue} \text{ Variable}$$

- ersetze $x \in \Sigma$ in den rechten Seiten durch V_x
- Bsp.: $A \to \mathtt{x}B$ wird zu $A \to V_{\mathtt{x}}B$ und $V_{\mathtt{x}} \to \mathtt{x}$

Regeländerungen erhalten Sprache und bringen G in CNF!

□

Bsp. $S \to \mathtt{a}\mathtt{a}S \mid \mathtt{a}SS\mathtt{b} \mid \varepsilon$

1. Neuer Startzustand

$$\begin{aligned} S' &\to S \\ S &\to \mathtt{a}\mathtt{a}S \mid \mathtt{a}SS\mathtt{b} \mid \varepsilon \end{aligned}$$

2. Eliminiere $A \to \varepsilon$ Regeln

$$\begin{aligned} S' &\to S \mid \varepsilon \\ S &\to \mathtt{a}\mathtt{a}S \mid \mathtt{a}SS\mathtt{b} \mid \mathtt{a}\mathtt{a} \mid \mathtt{a}S\mathtt{b} \mid \mathtt{a}\mathtt{b} \end{aligned}$$

3. Eliminiere $A \to B$ Regeln

$$\begin{aligned} S' &\to \mathtt{a}\mathtt{a}S \mid \mathtt{a}SS\mathtt{b} \mid \mathtt{a}\mathtt{a} \mid \mathtt{a}S\mathtt{b} \mid \mathtt{a}\mathtt{b} \mid \varepsilon \\ S &\to \mathtt{a}\mathtt{a}S \mid \mathtt{a}SS\mathtt{b} \mid \mathtt{a}\mathtt{a} \mid \mathtt{a}S\mathtt{b} \mid \mathtt{a}\mathtt{b} \end{aligned}$$

4. Kürze zu lange rechte Seiten

$$\begin{aligned} S' &\to \mathtt{a}X_1 \mid \mathtt{a}X_2 \mid \mathtt{a}\mathtt{a} \mid \mathtt{a}X_3 \mid \mathtt{a}\mathtt{b} \mid \varepsilon \\ S &\to \mathtt{a}X_1 \mid \mathtt{a}X_2 \mid \mathtt{a}\mathtt{a} \mid \mathtt{a}X_3 \mid \mathtt{a}\mathtt{b} \\ X_1 &\to \mathtt{a}S \\ X_2 &\to SX_3 \\ X_3 &\to S\mathtt{b} \end{aligned}$$

$$
\begin{array}{lcl}
S' & \to & \mathtt{a}X_1 \mid \mathtt{a}X_2 \mid \mathtt{aa} \mid \mathtt{a}X_3 \mid \mathtt{ab} \mid \varepsilon \\
S & \to & \mathtt{a}X_1 \mid \mathtt{a}X_2 \mid \mathtt{aa} \mid \mathtt{a}X_3 \mid \mathtt{ab} \\
X_1 & \to & \mathtt{a}S \\
X_2 & \to & SX_3 \\
X_3 & \to & S\mathtt{b}
\end{array}
$$

5. Ersetzte Regeln der Form $A \to xB$, $A \to Bx$ und $A \to xy$

$$
\begin{array}{lcl}
S' & \to & V_\mathtt{a}X_1 \mid V_\mathtt{a}X_2 \mid V_\mathtt{a}V_\mathtt{a} \mid V_\mathtt{a}X_3 \mid V_\mathtt{a}V_\mathtt{b} \mid \varepsilon \\
S & \to & V_\mathtt{a}X_1 \mid V_\mathtt{a}X_2 \mid V_\mathtt{a}V_\mathtt{a} \mid V_\mathtt{a}X_3 \mid V_\mathtt{a}V_\mathtt{b} \\
X_1 & \to & V_\mathtt{a}S \\
X_2 & \to & SX_3 \\
X_3 & \to & SV_\mathtt{b} \\
V_\mathtt{a} & \to & \mathtt{a} \\
V_\mathtt{b} & \to & \mathtt{b}
\end{array}
$$

Chomsky Normalform

$$S \to \mathtt{aa}S \mid \mathtt{a}SS\mathtt{b} \mid \varepsilon$$

Das Wortproblem für kf. Sprachen

Wortproblem

Eingabe: $w \in \Sigma^*$

Frage: $w \in L$? für $L \in$ CFL

- Wortproblem für reguläre Sprachen trivial (Benutze DEA)
- für kf. Sprachen: Algorithmus von Cocke, Younger, Kasami (**CYK-Algorithmus**)
- Ausgangspunkt: kf. Grammatik für L in CNF
- Algorithmus folgt dem Paradigma des *Dynamischen Programmierens*

CYK Algorithmus

- für eine Wort w bezeichnet $w[i,j]$ das Teilwort vom $i$ten bis $j$ten Zeichen

  Bsp.: $w = \texttt{autobahn} \longrightarrow w[2,5] = \texttt{utob}$

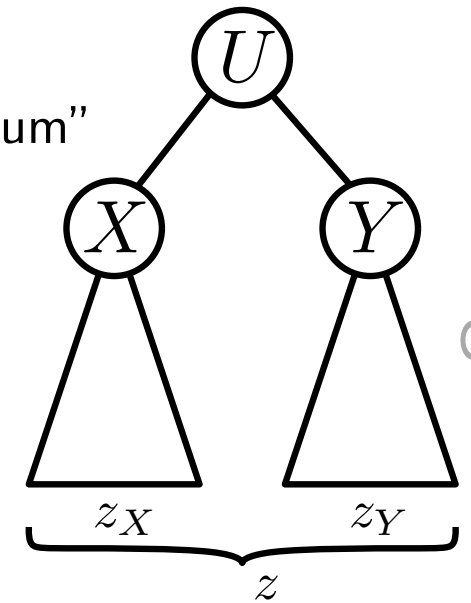
- $V_{ij} := \{U \in V \mid U \Rightarrow^* w[i,j]\}$
- **Teilprobleme des DP:** Bestimme die Mengen V_{ij}
- $w \in L \iff S \in V_{1,n}$, mit $|w| = n$
- Initialisierung: $V_{ii} = \{X \mid X \to w[i,i]\}$ für $i = 1, \ldots, n$
- Rekursiver Schritt: Bestimme V_{ij} mit Hilfe bereits bekannter Teilprobleme
- Abstand zwischen i und j wird in jeder *Runde* um 1 größer

Finden einer Rekursion

- sei $w[i,j] = z$, wir suchen Variablen U mit

  $U \Rightarrow XY \Rightarrow \cdots \Rightarrow z$

  2 Variablen da CNF



"Ableitungsbaum"

- es muss möglich sein $z = z_1 z_2$ zu zerteilen mit
  1. $U \Rightarrow U_1 U_2$
  2. $U_1 \Rightarrow^* z_1$
  3. $U_2 \Rightarrow^* z_2$

  } Teilprobleme schon gelöst

- ich darf keine Zerteilung ausschließen $\rightarrow$ ich muss alle Möglichkeiten z zu teilen in Betracht ziehen
- **Rekursion:**
  $V_{ij} = \bigcup_{k=i}^{j-1} \{X \mid X \rightarrow AB \text{ mit } A \in V_{i,k}, B \in V_{k+1,j}\}$

Algorithmus für $w \neq \varepsilon$

Datenstruktur: T, 2d Array, $T[i,j]$ enthält V_{ij}

Algorithm 2: CYK Algorithmus

```
1  for i = 1 to n do T[i,i] = {X ∈ V | X → w[i,i]} ;
2  for d = 1 to n − 1 do
3      for i = 1 to n − d do
4          j = i + d;
5          for t = i to j − 1 do
6              for all A → XY ∈ R do
7                  if X ∈ T[i,t] and Y ∈ T[t+1,j] then
                   T[i,j] = T[i,j] ∪ {A};
8              end
9          end
10     end
11 end
12 if S ∈ T[1,n] then return true;
13 else return false;
```

Algorithmus für $w \neq \varepsilon$

Datenstruktur: T, 2d Array, $T[i,j]$ enthält V_{ij}

Algorithm 2: CYK Algorithmus

```
1  for i = 1 to n do T[i,i] = {X ∈ V | X → w[i,i]} ;   Initialisierung
2  for d = 1 to n − 1 do                               d ist Abstand i/j
3      for i = 1 to n − d do
4          j = i + d;                                  aktueller Eintrag T[i,j]
5          for t = i to j − 1 do                       t ist Trennstelle
6              for all A → XY ∈ R do
7                  if X ∈ T[i,t] and Y ∈ T[t+1,j] then
                   T[i,j] = T[i,j] ∪ {A};
8              end
9          end
10     end
11 end
12 if S ∈ T[1,n] then return true;
13 else return false;
```

Bsp. 1

$$
\begin{array}{lcl}
S & \to & AB \mid CA \\
A & \to & AA \mid CB \mid \texttt{a} \\
B & \to & AC \mid \texttt{b} \\
C & \to & \texttt{c}
\end{array}
\qquad w = \texttt{cbaac}
$$

	1	2	3	4	5
5					C
4				A	B
3			A	A	S, B
2		B	–	–	–
1	C	A	A	A	S, B

$S \in T[1, n]$ deshalb $w \in L(G)$