Theoretische Grundlagen der Informatik
Vorlesung am 09. Januar 2020

INSTITUT FÜR THEORETISCHE INFORMATIK

Übersicht

Chomsky-Hierarchie	Wortproblem letzte Vorlesung	Beispiele heute
	semi-entscheidbar NTM akzeptiert	
Chomsky-1 kontextsensitiv	$\mathcal{NP}$-schwer $\mathcal{NTAPE}(n)$	
Chomsky-2 kontextfrei	polynomiell CYK-Algorithmus	
Chomsky-3 regulär	linear DEA	

Das Pumping-Lemma für kontextfreie Sprachen

Satz (Pumping-Lemma für reguläre Sprachen):
Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n\,,\ v \neq \varepsilon$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Satz (Pumping-Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ eine Darstellung

$$z = uvwxy \text{ mit } |vwx| \leq n\,,\ vx \neq \varepsilon$$

existiert, bei der auch $uv^iwx^iy \in L$ ist für alle $i \in \mathbb{N}_0$.

Das Pumping-Lemma für kontextfreie Sprachen

Satz (Pumping-Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ eine Darstellung

$$z = uvwxy \text{ mit } |vwx| \leq n\,,\ vx \neq \varepsilon$$

existiert, bei der auch $uv^iwx^iy \in L$ ist für alle $i \in \mathbb{N}_0$.

Für alle	$\forall L \subseteq \Sigma^*$ mit L kontextfrei
existiert	$\exists n \in \mathbb{N}$
für alle	$\forall z \in L$ mit $\|z\| > n$
existiert	$\exists u, v, w, x, y \in \Sigma^*$ mit $z = uvwxy, \|vwx\| \leq n, vx \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0$:
gilt	$uv^iwx^iy \in L$

Ogden's Lemma für kontextfreie Sprachen

Karlsruher Institut für Technologie

Satz (Ogden's Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ gilt:

Wenn wir in z mindestens n Buchstaben markieren, so existiert eine Darstellung

$$z = uvwxy,$$

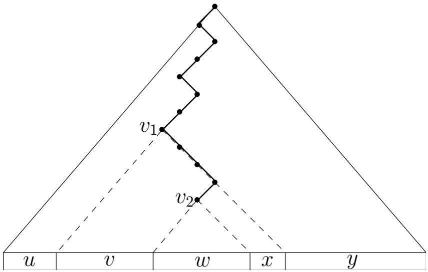
in der von den mindestens n markierten Buchstaben

- höchstens n zu vwx gehören und
- mindestens einer zu vx gehört,

bei der auch $uv^iwx^iy \in L$ ist für alle $i \in \mathbb{N}_0$.

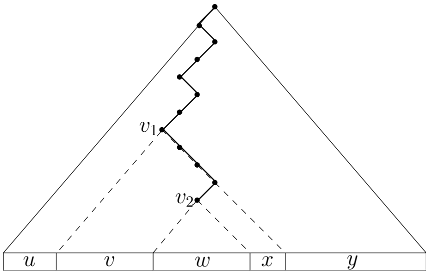
Beweis von Ogden's Lemma

- Sei L kontextfreie Sprache.
- Sei G Grammatik zu L mit Variablen V in Chomsky-Normalform, d.h. alle Regeln sind von der Form $A \to BC$ oder $A \to a$.
- Setze $n := 2^{|V|+1}$.
- Wähle beliebiges Wort $z \in L$ mit $|z| > n$.
- Betrachte einen Syntaxbaum T zu z.



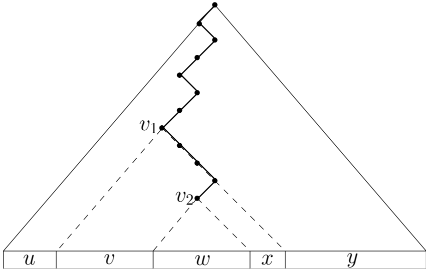
Beweis von Ogden's Lemma

- T hat $|z|$ Blätter, die Vorgänger der Blätter haben 1 Nachfolger und alle weiteren inneren Knoten haben 2 Nachfolger.
- Seien nun mindestens n Blätter markiert.
- Durchlaufe einen Weg von der Wurzel zu einem Blatt wie folgt: Wähle stets den Nachfolger, auf dessen Seite die größere Anzahl markierter Blätter liegt.
- Nenne Knoten auf dem Weg, für die rechter und linker Unterbaum markierte Blätter hat, **Verzweigungsknoten**.



Beweis von Ogden's Lemma

- Wegen $n > 2^{|V|}$ liegen auf dem Weg mindestens $|V| + 1$ Verzweigungsknoten.
- Von den letzten $|V| + 1$ Verzweigungsknoten entsprechen mindestens zwei Knoten v_1, v_2 derselben Variablen A.
- Sei vwx das Teilwort von z im Unterbaum von v_1.
- Sei w das Teilwort von z im Unterbaum von v_2.
- Damit sind u und y eindeutig bestimmt.



Beweis von Ogden's Lemma

- Da v_1 Verzweigungsknoten ist, enthält vx mindestens einen markierten Buchstaben.
- Da der Unterbaum von v_1 inkl. v_1 nur $|V| + 1$ Verzweigungsknoten enthält, gibt es in vwx höchstens $2^{|V|+1} = n$ markierte Buchstaben.
- Zu G existieren die Ableitungen

$$S \xrightarrow{*} uAy, \qquad A \xrightarrow{*} vAx, \qquad A \xrightarrow{*} w.$$

Daraus kann z abgeleitet werden durch

$$S \xrightarrow{*} uAy \xrightarrow{*} uvAxy \xrightarrow{*} uvwxy = z,$$

aber auch uv^iwx^iy für jedes $i \geq 1$ durch

$$S \xrightarrow{*} uAy \xrightarrow{*} uvAxy \xrightarrow{*} uv^2Ax^2y \xrightarrow{*} \cdots \rightarrow uv^iAx^iy \rightarrow uv^iwx^iy.$$

Also ist auch $uv^iwx^iy \in L$ für $i \geq 0$.

Bemerkung

- Der Spezialfall von Ogden's Lemma, in dem alle Buchstaben von z markiert sind, ist gerade das Pumping-Lemma für kontextfreie Sprachen.

Satz (Ogden's Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ gilt:

Wenn wir in z mindestens n Buchstaben markieren, so existiert eine Darstellung

$$z = uvwxy,$$

in der von den mindestens n markierten Buchstaben

- höchstens n zu vwx gehören und
- mindestens einer zu vx gehört,

bei der auch $uv^iwx^iy \in L$ ist für alle $i \in \mathbb{N}_0$.

Übersicht

Chomsky-Hierarchie	**Wortproblem** letzte Vorlesung	**Beispiele** heute
Chomsky-0	semi-entscheidbar NTM akzeptiert	
Chomsky-1 kontextsensitiv	$\mathcal{NP}$-schwer $\mathcal{NTAPE}(n)$	
Chomsky-2 kontextfrei	polynomiell CYK-Algorithmus	
Chomsky-3 regulär	linear DEA	

Echtheit der Chomsky-Hierarchie

Satz:
Die Chomsky-Hierarchie ist echt, d.h.

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0 \;,$$

wobei $\mathcal{L}_i$, $0 \leq i \leq 3$, die Klasse der durch Typ-i-Grammatiken erzeugten Sprachen bezeichnet.

Beweis

Teil 1:	Es gibt eine kontextfreie Sprache L,	$\rightsquigarrow L \in \mathcal{L}_2$
	die nicht regulär ist.	$\rightsquigarrow L \notin \mathcal{L}_3$
Teil 2:	Es gibt eine kontextsensitive Sprache L,	$\rightsquigarrow L \in \mathcal{L}_1$
	die nicht kontextfrei ist.	$\rightsquigarrow L \notin \mathcal{L}_2$
Teil 3:	Es gibt eine semi-entscheidbare Sprache L,	$\rightsquigarrow L \in \mathcal{L}_0$
	die nicht kontextsensitiv ist.	$\rightsquigarrow L \notin \mathcal{L}_1$

Beweis – Teil 1

Es gibt eine kontextfreie Sprache, die nicht regulär ist.

Die Sprache

$$L = \{a^i b^i \mid i \geq 1\}$$

ist kontextfrei und wird durch die Grammatik

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{a, b\} \\ R &= \{S \to ab \mid aSb\}\ . \end{aligned}$$

erzeugt. Sie ist aber nicht regulär.
(Siehe auch Beispiele zum Pumping-Lemma für reguläre Sprachen.)

Beweis – Teil 2

Es gibt eine kontextsensitive Sprache, die nicht kontextfrei ist.

Die Sprache

$$L = \{a^i b^i c^i \mid i \geq 1\}$$

ist kontextsensitiv.

Beweis

- L kontextsensitiv $\Leftrightarrow$ es gibt NTM mit linearem Speicherbedarf für L
- Eingabe $w \in \{a, b, c\}^*$
- Überprüfe deterministisch, ob $w = a^i b^j c^k$
- Überprüfe deterministisch, ob $j = i$ und $k = i$
- Speicherbedarf: $i + j + k$, also linear
- $\Rightarrow$ L kontextsensitiv

Beweis – Teil 2

Es gibt eine kontextsensitive Sprache, die nicht kontextfrei ist.

Die Sprache

$$L = \{a^i b^i c^i \mid i \geq 1\}$$

ist nicht kontextfrei.

Satz (Pumping-Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ eine Darstellung

$$z = uvwxy \text{ mit } |vwx| \leq n\,,\ vx \neq \varepsilon$$

existiert, bei der auch $uv^i wx^i y \in L$ ist für alle $i \in \mathbb{N}_0$.

Durch **Widerlegen** der Aussage des Pumping-Lemmas für eine gegebene Sprache L zeigen wir, dass L **nicht kontextfrei** ist.

Beweis – Teil 2

Aussage des kontextfreien PL für Sprache L:
$\exists n \quad \forall z \in L, |z| > n \quad \exists uvwxy = z, |vwx| \leq n, vx \neq \varepsilon$
$\forall i \in \mathbb{N}_0: \quad uv^i wx^i y \in L$

Widerlegen der Aussage
$\forall n \quad \exists z \in L, |z| > n \quad \forall uvwxy = z, |vwx| \leq n, vx \neq \varepsilon \quad \exists i \in \mathbb{N}_0:$
$uv^i wx^i y \notin L$

Die Sprache $L = \{a^i b^i c^i \mid i \geq 1\}$ ist nicht kontextfrei.

Beweis

“$\forall$” Betrachte beliebiges $n \in \mathbb{N}$.
“$\exists$” Wähle $z = a^n b^n c^n$. Beachte: $|z| = 3n > n$ und $z \in L$.
“$\forall$” Betrachte beliebige Zerlegung $z = uvwxy$, $|vwx| \leq n$, $vx \neq \varepsilon$.
“$\exists$” Wähle $i = 0$.

Beweis – Teil 2

Widerlegen der Aussage

$\forall n \quad \exists z \in L, |z| > n \quad \forall uvwxy = z, |vwx| \leq n, vx \neq \varepsilon \quad \exists i \in \mathbb{N}_0 :$
$uv^i wx^i y \notin L$

Die Sprache $L = \{a^i b^i c^i \mid i \geq 1\}$ ist nicht kontextfrei.

Beweis

“$\forall$” Betrachte beliebiges $n \in \mathbb{N}$.
“$\exists$” Wähle $z = a^n b^n c^n$. Beachte: $|z| = 3n > n$ und $z \in L$.
“$\forall$” Betrachte beliebige Zerlegung $z = uvwxy$, $|vwx| \leq n$, $vx \neq \varepsilon$.
“$\exists$” Wähle $i = 0$.

Fallunterscheidung, Fall 1: vwx enthält kein c.
Dann ist $uv^0wx^0y = a^r b^s c^n \notin L$, weil entweder $r < n$ oder $s < n$.

Fallunterscheidung, Fall 2: vwx enthält kein a.
Dann ist $uv^0wx^0y = a^n b^r c^s \notin L$, weil entweder $r < n$ oder $s < n$.

Beweis – Teil 2

Die Sprache $L = \{a^i b^i c^i \mid i \geq 1\}$ ist nicht kontextfrei.

Alternativer Beweis mit Ogden's Lemma

Satz (Ogden's Lemma für kontextfreie Sprachen):
Sei L eine kontextfreie Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $z \in L$ mit $|z| > n$ gilt:

Wenn wir in z mindestens n Buchstaben markieren, so existiert eine Darstellung

$$z = uvwxy,$$

in der von den mindestens n markierten Buchstaben

- höchstens n zu vwx gehören und
- mindestens einer zu vx gehört,

bei der auch $uv^i wx^i y \in L$ ist für alle $i \in \mathbb{N}_0$.

Beweis – Teil 2

Die Sprache $L = \{a^i b^i c^i \mid i \geq 1\}$ ist nicht kontextfrei.

Alternativer Beweis mit Ogden's Lemma

Aussage von Ogden's Lemma für Sprache L:
$\exists n \quad \forall z \in L, |z| > n$, mind. n Markierungen
$\exists uvwxy = z$, vwx mit höchst. n Markierungen, vx mit mind. 1 Markierung $\quad \forall i \in \mathbb{N}_0: \quad uv^iwx^iy \in L$

Widerlegen der Aussage
$\forall n \quad \exists z \in L, |z| > n$, mind. n Markierungen
$\forall uvwxy = z$, vwx höchst. n Markierungen, vx mind. 1 Markierung
$\exists i \in \mathbb{N}_0: \quad uv^iwx^iy \notin L$

Beweis – Teil 2

Die Sprache $L = \{a^i b^i c^i \mid i \geq 1\}$ ist nicht kontextfrei.

Alternativer Beweis mit Ogden's Lemma

Widerlegen der Aussage

$\forall n \quad \exists z \in L, |z| > n$, mind. n Markierungen

$\forall uvwxy = z$, vwx höchst. n Markierungen, vx mind. 1 Markierung

$\exists i \in \mathbb{N}_0: \quad uv^i wx^i y \notin L$

"$\forall$" Betrachte beliebiges $n \in \mathbb{N}$.

"$\exists$" Wähle $z = a^{n+1} b^{n+1} c^{n+1}$ und markiere alle b.

(Beachte: $|z| > n$, $z \in L$, mind. n Markierungen.)

"$\forall$" Betrachte beliebige Zerlegung $z = uvwxy$, sodass vwx höchst. n Markierungen, vx mind. 1 Markierung hat.

"$\exists$" Wähle $i = 0$.

Da vwx kein a oder kein c hat, gilt $uv^0 wx^0 y \notin L$.

Echtheit der Chomsky-Hierarchie

Satz:
Die Chomsky-Hierarchie ist echt, d.h.

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0 ,$$

wobei $\mathcal{L}_i$, $0 \leq i \leq 3$, die Klasse der durch Typ-i-Grammatiken erzeugten Sprachen bezeichnet.

Beweis

Teil 1: Es gibt eine kontextfreie Sprache L, $\rightsquigarrow L \in \mathcal{L}_2$
die nicht regulär ist. $\rightsquigarrow L \notin \mathcal{L}_3$

Teil 2: Es gibt eine kontextsensitive Sprache L, $\rightsquigarrow L \in \mathcal{L}_1$
die nicht kontextfrei ist. $\rightsquigarrow L \notin \mathcal{L}_2$

Teil 3: Es gibt eine semi-entscheidbare Sprache L, $\rightsquigarrow L \in \mathcal{L}_0$
die nicht kontextsensitiv ist. $\rightsquigarrow L \notin \mathcal{L}_1$

Beweis – Teil 3

Es gibt eine semi-entscheidbare Sprache, die nicht kontextsensitiv ist.

Es sei L_u die universelle Sprache.

Wiederholung
Die **universelle Sprache** L_u über $\{0, 1\}$ ist definiert durch

$$L_u := \{wv \mid v \in L(T_w)\}.$$

L_u ist also die Menge aller Wörter wv, für die die DTM T_w bei der Eingabe v hält und v akzeptiert.

Beweis – Teil 3

Es gibt eine semi-entscheidbare Sprache, die nicht kontextsensitiv ist.

Es sei L_u die universelle Sprache.

- Kapitel 3: L_u ist semi-entscheidbar (aber nicht entscheidbar).
- Wegen der Semi-Entscheidbarkeit gilt $L_u \in \mathcal{L}_0$.
- **Da alle Sprachen in $\mathcal{L}_1$ entscheidbar sind, gilt $L_u \notin \mathcal{L}_1$.**

- Sei L eine Sprache in $\mathcal{L}_1$. (Zum Beispiel $L = L_u$.)
- Dann gibt es eine NTM, die L mit linearem Speicher akzeptiert.
- Diese kann durch eine DTM simuliert werden.
- Mit beschränktem Speicher können nur endlich viele verschiedene Konfigurationen auftreten.
- Dann können Endlosschleifen erkannt werden.
- $\Longrightarrow$ Sprache L kann sogar entschieden werden.

Übersicht

Chomsky-Hierarchie	Wortproblem letzte Vorlesung	Beispiele heute
Chomsky-0	semi-entscheidbar NTM akzeptiert	universelle Sprache
Chomsky-1 kontextsensitiv	$\mathcal{NP}$-schwer $\mathcal{NTAPE}(n)$	$L = \{a^i b^i c^i \mid i \geq 1\}$
Chomsky-2 kontextfrei	polynomiell CYK-Algorithmus	$L = \{a^i b^i \mid i \geq 1\}$
Chomsky-3 regulär	linear DEA	$L = \{a^i \mid i \geq 1\}$

Eigenschaften kontextfreier Sprachen

Satz:
Für eine kontextfreie Grammatik G kann in polynomieller Zeit entschieden werden, ob $L(G) = \emptyset$ ist.

Bemerkung: Für Chomsky-0-Grammatiken ist das nicht entscheidbar.

Satz:
Die Klasse der kontextfreien Sprachen ist abgeschlossen bzgl. Vereinigung, Konkatenation und Kleene'schem Abschluss.

Testen Sie sich: Für welche Sprachen gilt das auch?

Semi-entscheidbare Sprachen?
Entscheidbare Sprachen?
Kontextsensitive Sprachen?
Reguläre Sprachen?

Nutzlose Variablen

Sei G eine kontextfreie Grammatik. Eine Variable A heißt **nutzlos**, falls es keine Ableitung $S \xrightarrow{*} w$ mit $w \in \Sigma^*$ gibt, in der A vorkommt.

Satz:
Für eine kontextfreie Grammatik kann die Menge der nutzlosen Variablen (in polynomialer Zeit) berechnet werden.

Beweis:

- Wir benutzen ein zweistufiges Verfahren.

Schritt 1

Bestimme alle Variablen, die ein Wort erzeugen können.

Formal: Berechne $V' = \{A \in V \mid \exists w \in \Sigma^*\colon A \stackrel{*}{\to} w\}$.

- Initialisiere eine leere Queue Q.
- Füge alle $A \in V$ mit $A \to w$ für ein $w \in \Sigma^*$ in Q und V' ein.
- Entferne der Reihe nach jedes Element A aus Q.
  - Ersetze jede Regel

    $B \to \alpha A \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$

    durch die Regeln

    $B \to \alpha w \beta$, wobei $w \in \Sigma^*$ und $A \to w$ Regel.
  - Wenn dabei eine Regel der Form

    $B \to w'$, $w' \in \Sigma^*$

    entsteht und $B \notin V'$, füge B in Q und V' ein.
- Das Verfahren endet, wenn Q leer ist.

Schritt 1

Bemerkung 1

- Falls $S \notin V'$, breche das Verfahren ab (kein Schritt 2).
- G erzeugt dann die leere Sprache und alle Variablen sind nutzlos.

Bemerkung 2

- Für jede Variable A mit $A \overset{*}{\rightarrow} w$ für ein $w \in \Sigma^*$ gilt:
- Per Induktion über die Länge der kürzesten Ableitungsregel der Form $A \overset{*}{\rightarrow} w$ kann für A gezeigt werden, dass $A \in V'$.

Beispiel: Schritt 1

Grammatik $G = (\Sigma, V, S, R)$ mit Produktionen R gegeben durch

$$\begin{aligned}
S &\rightarrow Aa|B|Cab \\
A &\rightarrow bc|A \\
B &\rightarrow Bd|Cd \\
C &\rightarrow aBc \\
D &\rightarrow Ab \\
E &\rightarrow SD
\end{aligned}$$

Beispiel: Schritt 1

Füge alle $A \in V$ mit $A \to w$ für ein $w \in \Sigma^*$ in Q und V' ein.

$$\begin{aligned} S &\to Aa|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to Ab \\ E &\to SD \\ \\ V' &= \emptyset \\ Q &= \emptyset \end{aligned}$$

$$\begin{aligned} S &\to Aa|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to Ab \\ E &\to SD \\ \\ V' &= \{A\} \\ Q &= \{A\} \end{aligned}$$

Beispiel: Schritt 1

Entferne der Reihe nach jedes Element A aus Q.

- Ersetze jede Regel $B \to \alpha A \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$ durch die Regeln $B \to \alpha w \beta$, wobei $w \in \Sigma^*$ und $A \to w$ Regel.
- Wenn dabei eine Regel der Form $B \to w'$, $w' \in \Sigma^*$ entsteht und $B \notin V'$, füge B in Q und V' ein.

$$\begin{array}{lcl} S & \to & Aa|B|Cab \\ A & \to & bc|A \\ B & \to & Bd|Cd \\ C & \to & aBc \\ D & \to & Ab \\ E & \to & SD \\ \\ V' & = & \{A\} \\ Q & = & \{A\} \end{array} \qquad \begin{array}{lcl} S & \to & bca|B|Cab \\ A & \to & bc|A \\ B & \to & Bd|Cd \\ C & \to & aBc \\ D & \to & bcb \\ E & \to & SD \\ \\ V' & = & \{A, S, D\} \\ Q & = & \{S, D\} \end{array}$$

Beispiel: Schritt 1

Entferne der Reihe nach jedes Element A aus Q.

- Ersetze jede Regel $B \to \alpha A \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$ durch die Regeln $B \to \alpha w \beta$, wobei $w \in \Sigma^*$ und $A \to w$ Regel.
- Wenn dabei eine Regel der Form $B \to w'$, $w' \in \Sigma^*$ entsteht und $B \notin V'$, füge B in Q und V' ein.

$$\begin{aligned} S &\to bca|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to bcb \\ E &\to SD \end{aligned}$$

$$\begin{aligned} V' &= \{A, S, D\} \\ Q &= \{S, D\} \end{aligned}$$

$$\begin{aligned} S &\to bca|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to bcb \\ E &\to \color{red}{bca}D \end{aligned}$$

$$\begin{aligned} V' &= \{A, S, D\} \\ Q &= \{D\} \end{aligned}$$

Beispiel: Schritt 1

Entferne der Reihe nach jedes Element A aus Q.

- Ersetze jede Regel $B \to \alpha A \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$ durch die Regeln $B \to \alpha w \beta$, wobei $w \in \Sigma^*$ und $A \to w$ Regel.
- Wenn dabei eine Regel der Form $B \to w'$, $w' \in \Sigma^*$ entsteht und $B \notin V'$, füge B in Q und V' ein.

$$\begin{aligned} S &\to bca|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to bcb \\ E &\to bcaD \end{aligned} \qquad \begin{aligned} S &\to bca|B|Cab \\ A &\to bc|A \\ B &\to Bd|Cd \\ C &\to aBc \\ D &\to bcb \\ E &\to bca\color{red}{bcb} \end{aligned}$$

$$\begin{aligned} V' &= \{A, S, D\} \\ Q &= \{D\} \end{aligned} \qquad \begin{aligned} V' &= \{A, S, D, \color{red}{E}\} \\ Q &= \{\color{red}{E}\} \end{aligned}$$

Beispiel: Schritt 1

Entferne der Reihe nach jedes Element A aus Q.

- Ersetze jede Regel $B \to \alpha A \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$ durch die Regeln $B \to \alpha w \beta$, wobei $w \in \Sigma^*$ und $A \to w$ Regel.
- Wenn dabei eine Regel der Form $B \to w'$, $w' \in \Sigma^*$ entsteht und $B \notin V'$, füge B in Q und V' ein.

$$\begin{aligned}
S &\to bca|B|Cab \\
A &\to bc|A \\
B &\to Bd|Cd \\
C &\to aBc \\
D &\to bcb \\
E &\to bcabcb \\
\\
V' &= \{A, S, D, E\} \\
Q &= \{E\}
\end{aligned}$$

$$\begin{aligned}
S &\to bca|B|Cab \\
A &\to bc|A \\
B &\to Bd|Cd \\
C &\to aBc \\
D &\to bcb \\
E &\to bcabcb \\
\\
V' &= \{A, S, D, E\} \\
Q &= \{\}
\end{aligned}$$

Schritt 2

Bestimme alle Variablen in V', die vom Startsymbol aus „erreicht“ werden können.

Formal: Berechne $\{A \in V' \mid S = A \text{ oder } \exists \alpha, \beta \in (V' \cup \Sigma)^* : S \xrightarrow{*} \alpha A \beta\}$.

- Starte mit $V'' = \{S\}$.
- Füge zu allen Regeln $A \to \alpha B \beta$ mit $\alpha, \beta \in (V' \cup \Sigma)^*$, $A \in V''$, $B \in V'$ die Variable B in V'' ein.
- Wiederhole den letzen Schritt, bis sich V'' nicht mehr ändert.

Per Induktion über die Länge der kürzesten Ableitungsregel der Form $S \to \alpha A \beta$, $\alpha, \beta \in (V' \cup \Sigma)^*$ kann dann wieder die Korrektheit bewiesen werden.

Fazit: Nach Ende von Schritt 2 ist V'' die Menge aller nützlichen Variablen.

Beispiel: Schritt 2

Starte mit $V'' = \{S\}$

$$
\begin{aligned}
S &\rightarrow Aa|B|Cab \\
A &\rightarrow bc|A \\
B &\rightarrow Bd|Cd \\
C &\rightarrow aBc \\
D &\rightarrow Ab \\
E &\rightarrow SD
\end{aligned}
$$

$$
\begin{aligned}
V' &= \{A, S, D, E\} \\
V'' &= \{\}
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow Aa|B|Cab \\
A &\rightarrow bc|A \\
B &\rightarrow Bd|Cd \\
C &\rightarrow aBc \\
D &\rightarrow Ab \\
E &\rightarrow SD
\end{aligned}
$$

$$
\begin{aligned}
V' &= \{A, S, D, E\} \\
V'' &= \{S\}
\end{aligned}
$$

Beispiel: Schritt 2

Füge zu allen Regeln $A \to \alpha B \beta$ mit $\alpha, \beta \in (V' \cup \Sigma)^*$, $A \in V''$, $B \in V'$ die Variable B in V'' ein.

$$\begin{array}{lcl} S & \to & Aa|B|Cab \\ A & \to & bc|A \\ B & \to & Bd|Cd \\ C & \to & aBc \\ D & \to & Ab \\ E & \to & SD \end{array} \qquad \begin{array}{lcl} S & \to & Aa|B|Cab \\ A & \to & bc|A \\ B & \to & Bd|Cd \\ C & \to & aBc \\ D & \to & Ab \\ E & \to & SD \end{array}$$

$$\begin{array}{lcl} V' & = & \{A, S, D, E\} \\ V'' & = & \{S\} \end{array} \qquad \begin{array}{lcl} V' & = & \{A, S, D, E\} \\ V'' & = & \{S, A\} \end{array}$$

Beispiel: Schritt 2

Wiederhole den letzen Schritt, bis sich V'' nicht mehr ändert.

$$
\begin{aligned}
S &\rightarrow Aa|B|Cab \\
A &\rightarrow bc|A \\
B &\rightarrow Bd|Cd \\
C &\rightarrow aBc \\
D &\rightarrow Ab \\
E &\rightarrow SD
\end{aligned}
$$

$$
\begin{aligned}
V' &= \{A, S, D, E\} \\
V'' &= \{S, A\}
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow Aa|B|Cab \\
A &\rightarrow bc|A \\
B &\rightarrow Bd|Cd \\
C &\rightarrow aBc \\
D &\rightarrow Ab \\
E &\rightarrow SD
\end{aligned}
$$

$$
\begin{aligned}
V' &= \{A, S, D, E\} \\
V'' &= \{S, A\}
\end{aligned}
$$

Leere kontextfreie Sprachen

Korollar
Für eine kontextfreie Grammatik G kann (in polynomialer Zeit) entschieden werden, ob $L(G) = \emptyset$ ist.

Beweis:

- $L(G) = \emptyset$ genau dann, wenn S nutzlos.

Endliche kontextfreie Sprachen

Satz:
Für eine kontextfreie Grammatik $G = (\Sigma, V, S, R)$ kann (in polynomialer Zeit) entschieden werden, ob $L(G)$ endlich ist.

Beweis:

- Entferne alle nutzlosen Variablen.
- Überführe G in eine äquivalente Grammatik in Chomsky-Normalform.
- Betrachte den gerichteten Graphen (V, E) mit
  - Knotenmenge V ist gleich der Variablenmenge von G,
  - Kantenmenge $E = \{(A, B) \mid \exists C \in V\colon A \to BC \in R \vee A \to CB \in R\}$.
- Mit Tiefensuche kann entschieden werden, ob dieser Graph einen gerichteten Kreis enthält.
- Man kann sich leicht überlegen, dass $L(G)$ genau dann endlich ist, wenn der entsprechende Graph keinen gerichteten Kreis enthält.

Beispielgraph

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow BC \\
B &\rightarrow BC \\
A &\rightarrow a \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}
$$