Theoretische Grundlagen der Informatik

Vorlesung am 22. Januar 2015

INSTITUT FÜR THEORETISCHE INFORMATIK

Das Pumping-Lemma für kontextfreie Sprachen

KIT – Karlsruher Institut für Technologie

Pumping-Lemma für kontextfreie Sprachen

Für jede kontextfreie Sprache L
gibt es eine Konstante $n \in \mathbb{N}$,
so dass sich jedes Wort $z \in L$ mit $|z| \geq n$

so als

- $z = uvwxy$

schreiben lässt, dass

- $|vx| \geq 1$,
- $|vwx| \leq n$ und
- für alle $i \geq 0$ das Wort $uv^i wx^i y \in L$ ist.

Ogden's Lemma für kontextfreie Sprachen

Ogden's Lemma für kontextfreie Sprachen

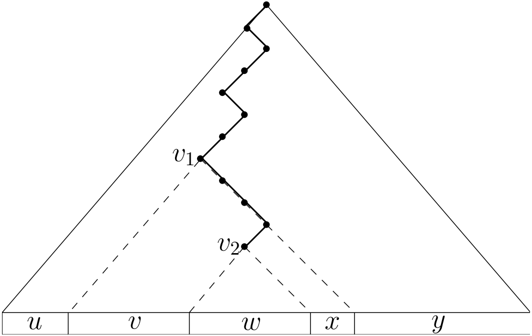
Für jede kontextfreie Sprache L
gibt es eine Konstante $n \in \mathbb{N}$,
so dass für jedes Wort $z \in L$ mit $|z| \geq n$ gilt:

Wenn wir in z mindestens n Buchstaben markieren, so lässt sich z so als $z = uvwxy$ schreiben,

- dass von den mindestens n markierten Buchstaben
  - mindestens einer zu vx gehört und
  - höchstens n zu vwx gehören und
- für alle $i \geq 0$ das Wort $uv^iwx^iy \in L$ ist.

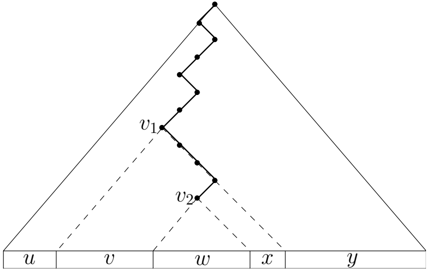
Beweis

- 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 $|z| \geq n$
- Betrachte einen Syntaxbaum T zu z.



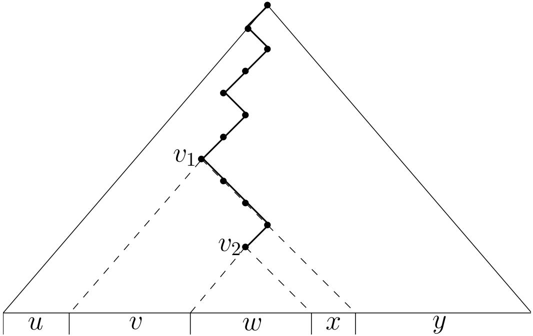
Beweis

- T hat $|z|$ Blätter und alle inneren Knoten außer den Vorgängern der Blätter haben Grad 2, ansonsten Grad 1.
- Seien mindestens n Blätter markiert.
- Durchlaufe einen Weg von der Wurzel zu einem Blatt.
  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

- 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 Wort unter Teilbaum mit Wurzel v_1
- Sei w Wort unter Teilbaum mit Wurzel v_2.
- Damit sind u und y eindeutig bestimmt.



Beweis

- 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 Odgen's Lemma, in dem alle Buchstaben von z markiert sind, ist gerade das Pumping–Lemma.

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

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

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

Beweis - Teil 1

Es gibt eine kontextfreie Sprache, die nicht regulär ist.

Die Sprache

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

ist kontextfrei und wird durch die Grammatik

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{0, 1\} \\ R &= \{S \to 01 \mid 0S1\}\ . \end{aligned}$$

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

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

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

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

ist nicht kontextfrei.

Beweis - Teil 2

Pumping-Lemma für kontextfreie Sprachen

Für jede kontextfreie Sprache L
gibt es eine Konstante $n \in \mathbb{N}$,
so dass sich jedes Wort $z \in L$ mit $|z| \geq n$

so als

- $z = uvwxy$

schreiben lässt, dass

- $|vx| \geq 1$,
- $|vwx| \leq n$ und
- für alle $i \geq 0$ das Wort $uv^i wx^i y \in L$ ist.

Beweis - Teil 2

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

Beweis

- Annahme: L sei kontextfrei. Sei dann n wie im PL gefordert.
- Wähle das Wort $z = a^n b^n c^n \in L$.
- Wir betrachten eine Zerlegung $z = uvwxy$ wie im PL gefordert:
  - $|vx| \geq 1$,
  - $|vwx| \leq n$ und
  - für alle $i \geq 0$ ist das Wort $uv^i wx^i y \in L$.

Beweis - Teil 2

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

Beweis

- Annahme: L sei kontextfrei. Sei dann n wie im PL gefordert.
- Wähle das Wort $z = a^n b^n c^n \in L$.
- Wir betrachten eine Zerlegung $z = uvwxy$ wie im PL gefordert:
  - $|vx| \geq 1$,
  - $|vwx| \leq n$ und
  - für alle $i \geq 0$ ist das Wort $uv^i wx^i y \in L$.
- Fallunterscheidung, Fall 1: vwx besteht nur aus a und b
  - Dann enthalt vx mindestens ein a oder b.
  - Damit ist $uv^0 wx^0 y = a^i b^j c^n \notin L$ weil entweder $i < n$ oder $j < n$.
  - Dies ist ein Widerspruch zum PL.

Beweis - Teil 2

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

Beweis

- Annahme: L sei kontextfrei. Sei dann n wie im PL gefordert.
- Wähle das Wort $z = a^n b^n c^n \in L$.
- Wir betrachten eine Zerlegung $z = uvwxy$ wie im PL gefordert:
  - $|vx| \geq 1$,
  - $|vwx| \leq n$ und
  - für alle $i \geq 0$ ist das Wort $uv^i wx^i y \in L$.
- Fallunterscheidung, Fall 2: vwx besteht nur aus b und c
  - Dann enthalt vx mindestens ein b oder c.
  - Damit ist $uv^0 wx^0 y = a^n b^i c^j \notin L$ weil entweder $i < n$ oder $j < n$.
  - Dies ist ein Widerspruch zum PL.

Beweis - Teil 2

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

Ogden's Lemma für kontextfreie Sprachen

Für jede kontextfreie Sprache L
gibt es eine Konstante $n \in \mathbb{N}$,
so dass für jedes Wort $z \in L$ mit $|z| \geq n$ gilt:

Wenn wir in z mindestens n Buchstaben markieren, so lässt sich z so als $z = uvwxy$ schreiben,

- dass von den mindestens n markierten Buchstaben
  - mindestens einer zu vx gehört und
  - höchstens n zu vwx gehören und
- für alle $i \geq 0$ das Wort $uv^i wx^i y \in L$ ist.

Beweis - Teil 2

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

Alternativer Beweis mit Odgen's Lemma

- Annahme: L sei kontextfrei.
- Sei dann n wie in Odgen's Lemma gefordert.
- Wähle das Wort $z = a^{n+1} b^{n+1} c^{n+1} \in L$.
- Markiere alle b.
- Damit enthält vwx mindestens ein b aber kein a oder kein c.
- Es enthalte vwx kein c (anderer Fall analog)
- Damit ist $uv^0 wx^0 y = a^i b^j c^n \notin L$ weil entweder $i < n$ oder $j < n$.
- Dies ist ein Widerspruch zu Odgen's Lemma.

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 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.

- L_u ist semi-entscheidbar, aber nicht entscheidbar (Kapitel 3).
- Wegen der Semi-entscheidbarkeit gilt $L_u \in \mathcal{L}_0$.
- Annahme: $L_u \in \mathcal{L}_1$.
- Dann gibt es eine NTM, die L_u mit linearem Speicher erkennt.
- Mit linearem Speicher können nur exponentiell viele verschiedene Konfigurationen auftreten.
- Diese könnte durch eine DTM durch Ausprobieren aller möglichen Konfigurationen simuliert werden.
- Dies wäre ein Widerspruch zur Nichtentscheidbarkeit von L_u.

Nutzlose Variablen

Sei G eine kontextfreie Grammatik. Eine Variable A heißt **nutzlos**, falls es keine Ableitung $S \xrightarrow{*} w$ gibt, $w \in \Sigma^*$, 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 $V' = \{A \in V \mid \exists w \in \Sigma^* : A \stackrel{*}{\to} w\}$

- 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
    $$B \to \alpha A \beta \text{ mit } \alpha, \beta \in (V \cup \Sigma)^*$$
    durch die Regeln
    $$B \to \alpha w \beta, \text{ wobei } w \in \Sigma^* \text{ und } A \to w \text{ Regel.}$$
  - 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.
- G erzeugt dann die leere Sprache und alle Variablen sind nutzlos.

Bemerkung 2

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

Beispiel: Schritt 1

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

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

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

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

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.

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

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

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.

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

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.

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

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

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

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

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.

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

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

Beispiel: Schritt 2

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

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

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

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

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

Korollar

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.

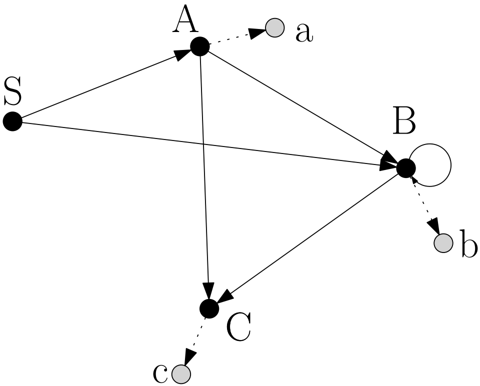
Satz:
Für eine kontextfreie Grammatik $G = (\Sigma, V, S, R)$ kann (in polynomialer Zeit) entschieden werden, ob $L(G)$ endlich ist.

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 $E = \{(A, B) \mid \exists C \in V : A \to BC \in R \vee A \to CB \in R\}$
- Mit Tiefensuche kann entschieden werden, ob dieser Graph einen Kreis enthält.
- Man kann sich leicht überlegen, dass $L(G)$ genau dann endlich ist, wenn der entsprechende Graph keinen Kreis enthält.

Beispielgraph

$$S \rightarrow AB$$
$$A \rightarrow BC$$
$$B \rightarrow BC$$
$$A \rightarrow a$$
$$B \rightarrow b$$
$$C \rightarrow c$$



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

Beweis:

- Seien L_1 kontextfreie Sprache mit Grammatik $G_1 = (\Sigma, V_1, S_1, R_1)$
- Seien L_2 kontextfreie Sprache mit Grammatik $G_2 = (\Sigma, V_2, S_2, R_2)$
- o.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Vereinigung: Die Grammatik

$$\begin{aligned} V &= V_1 \cup V_2 \cup \{S\} \\ S & \quad \text{neues Startsymbol} \\ R &= R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\} \end{aligned}$$

erzeugt $L_1 \cup L_2$.

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

Beweis:

- Seien L_1 kontextfreie Sprache mit Grammatik $G_1 = (\Sigma, V_1, S_1, R_1)$
- Seien L_2 kontextfreie Sprache mit Grammatik $G_2 = (\Sigma, V_2, S_2, R_2)$
- o.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Konkatenation: Die Grammatik

$$\begin{array}{lcl} V & = & V_1 \cup V_2 \cup \{S\} \\ S & & \text{neues Startsymbol} \\ R & = & R_1 \cup R_2 \cup \{S \to S_1 S_2\} \end{array}$$

erzeugt $L_1 \cdot L_2$.

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

Beweis:

- Seien L_1 kontextfreie Sprache mit Grammatik $G_1 = (\Sigma, V_1, S_1, R_1)$
- Seien L_2 kontextfreie Sprache mit Grammatik $G_2 = (\Sigma, V_2, S_2, R_2)$
- o.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Kleenscher Abschluss: Die Grammatik

$$\begin{array}{lcl} V & = & V_1 \cup \{S\} \\ S & & \text{neues Startsymbol} \\ R & = & R_1 \cup \{S \to \varepsilon, S \to SS, S \to S_1\} \end{array}$$

erzeugt L_1^*.

Satz:
Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen bzgl. Komplementbildung und Durchschnitt.

Beweis Schnitt: Betrachte die kontextfreien Sprachen

$$L_1 = \{a^n b^n | n \geq 1\} \qquad L_2 = \{c\}^*$$
$$L_3 = \{a\}^* \qquad L_4 = \{b^n c^n | n \geq 1\}$$

Nach dem letzen Satz sind dann auch $L_1 \cdot L_2$ und $L_3 \cdot L_4$ kontextfrei.
Es ist dann

$$L := L_1 L_2 \cap L_3 L_4 = \{a^n b^n c^n | n \geq 1\} \ .$$

Diese Sprache ist nicht kontextfrei.

Satz:
Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen bzgl. Komplementbildung und Durchschnitt.

Beweis Komplementbildung:

- Angenommen, die Klasse der kontextfreien Sprachen wäre bzgl. Komplementbildung abgeschlossen.
- Dann würde für beliebige kontextfreie Sprachen L_1, L_2 gelten $(L_1^c \cup L_2^c)^c = L_1 \cap L_2$ ist wieder kontextfrei.
- Dies ist ein Widerspruch zur ersten Aussage des Satzes.