Das Leerheitsproblem

Definition

Das Leerheitsproblem für CFG:

- Eingabe: Eine CFG $G = (N, T, P, S)$
- Frage: Ist $L(G) = \emptyset$?

Theorem

Das Leerheitsproblem für CFG läßt sich in polynomieller Zeit lösen.

Beweis.

Es ist $L(G) = \emptyset$ gdw. $S \notin pre^*(T^*)$. □

Unerreichbare Symbole

Definition

Finden von unerreichbaren Symbolen einer CFG:

- Eingabe: Eine CFG $G = (N, T, P, S)$
- Ausgabe: Eine Liste der unerreichbaren Symbole von G:
  $\{ A \in N \mid$ es gibt kein $\alpha A \beta \in (N \cup T)^*$ mit $S \overset{*}{\Rightarrow} \alpha A \beta \}$

Theorem

Unerreichbare Symbole einer CFG lassen sich in polynomieller Zeit finden.

Beweis.

A unerreichbar gdw. $S \notin \mathit{pre}^*((N \cup T)^* A (N \cup T)^*)$ □

Nullierbare Symbole

Definition

Finden von nullierbaren Symbolen einer CFG:

- Eingabe: Eine CFG $G = (N, T, P, S)$
- Ausgabe: Eine Liste der nullierbaren Symbole von G: $\{ A \in N \mid A \overset{*}{\Rightarrow} \epsilon \}$

Theorem

Nullierbare Symbole einer CFG lassen sich in polynomieller Zeit finden.

Beweis.

Die Menge der nullierbaren Symbole ist $N \cap pre^*(\{\epsilon\})$. □

Später: Wie entfernt man nullierbare Symbole?

Das Endlichkeitsproblem für CFG

Definition

Das Endlichkeitsproblem für CFG:

- Eingabe: Eine CFG $G = (N, T, P, S)$
- Frage: Ist $L(G)$ endlich?

Theorem

Das Endlichkeitsproblem für CFG läßt sich in polynomieller Zeit lösen.

Beweis.

Ersetze G durch G' mit $L(G') = L(G) \setminus \{\epsilon\}$, aber G' enthält keine unproduktiven, unerreichbare oder nullierbare Symbole.

$|L(G)| = \infty$ gdw. es gibt $A \in N$ mit
$A \in pre^*_{G'}\big((N \cup T)^+ A (N \cup T)^* \cup (N \cup T)^* A (N \cup T)^+\big)$. □

Das Schnittleerheitsproblem für CFG

Definition

Das Schnittleerheitsproblem für CFG:

- Eingabe: Zwei CFG G_1 und G_2
- Frage: Ist $L(G_1) \cap L(G_2) = \emptyset$?

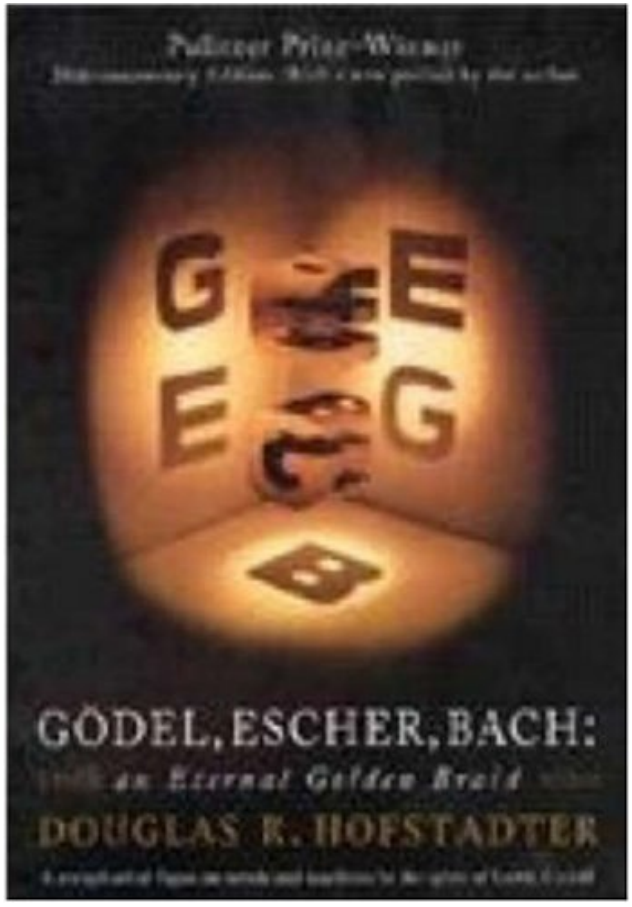
Theorem

Das Schnittleerheitsproblem für CFG ist „nicht berechenbar“.

Beweis.

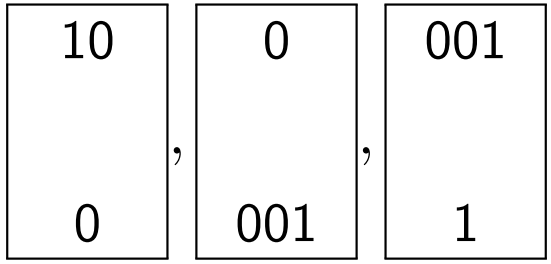
Idee: Gäbe es einen Algorithmus für dieses Problem, dann wäre auch das Post'sche Korrespondenzproblem lösbar. □

Weitere Buchempfehlung



Das Post'sche Korrespondenzproblem

Gegeben ist eine Menge von Karten dieser Form:



Frage:
Kann ich Karten dieser Art so aneinanderlegen, daß oben und unten dasselbe Wort entsteht?

(Jede Karte kann beliebig oft verwendet werden)

In *Berechenbarkeit und Komplexität* wird folgendes bewiesen:

Theorem

Das Post'sche Korrespondenzproblem ist nicht entscheidbar.

Theorem

Das Schnittleerheitsproblem für CFG ist nicht entscheidbar.

Beweis.

Es sei

$$I := \left\{ \begin{array}{|c|}\hline u_1 \\ \\ v_1 \\ \hline \end{array}, \begin{array}{|c|}\hline u_2 \\ \\ v_2 \\ \hline \end{array}, \ldots, \begin{array}{|c|}\hline u_n \\ \\ v_n \\ \hline \end{array} \right\}$$

eine PCP-Instanz.

Konstruiere zwei Grammatiken:

G_1: $S \to u_1 S v_1^R \mid \cdots \mid u_n S v_n^R \mid \$$

G_2: $S \to 0S0 \mid 1S1 \mid \$$

Behauptung: $L(G_1) \cap L(G_2) \neq \emptyset$ gdw. I lösbar. □

Das Eindeutigkeitsproblem für CFG

Definition

Das Eindeutigkeitsproblem für CFG:

- Eingabe: Eine CFG G
- Frage: Ist G eine eindeutige CFG?

Theorem

Das Eindeutigkeitsproblem für CFG ist nicht berechenbar.

Beweis.

Idee: Gäbe es einen Algorithmus für dieses Problem, dann wäre auch das Post'sche Korrespondenzproblem lösbar. □

Beweis.

Es sei

$$I := \left\{ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}, \ldots, \begin{bmatrix} u_n \\ v_n \end{bmatrix} \right\}$$

eine PCP-Instanz.

$$S \to A \mid B$$
$$A \to u_1 A x_1 \mid u_2 A x_2 \mid \cdots \mid u_n A x_n \mid u_1 x_1 \mid u_2 x_2 \mid \cdots \mid u_n x_n$$
$$B \to v_1 B x_1 \mid v_2 B x_2 \mid \cdots \mid v_n B x_n \mid v_1 x_1 \mid v_2 x_2 \mid \cdots \mid v_n x_n$$

Behauptung:

Die Grammatik ist eindeutig gdw. I keine Lösung besitzt. □

Das Universalitätsproblem für CFG

Definition

Das Universalitätsproblem für CFG:

- Eingabe: Eine CFG $G = (N, T, P, S)$
- Frage: Ist $L(G) = T^*$?

Theorem

Das Universalitätsproblem für CFG ist nicht berechenbar.

Beweis.

Idee: Post'sches Korrespondenzproblem. □

Beweis.

$$I := \left\{ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}, \ldots, \begin{bmatrix} u_n \\ v_n \end{bmatrix} \right\}$$

- $L_1 = \{ h^{-1}(w\$w^R) \mid w \in \{0,1\}^* \}$,
  $h\colon 0 \mapsto 0, 1 \mapsto 1, ¢ \mapsto \epsilon, \$ \mapsto \$$
- $L_2 = \{u_{i_1} ¢ \ldots ¢ u_{i_k} \$ v_{i_k}^R ¢ \ldots ¢ v_{i_1}^R \mid 1 \leq i_1, \ldots i_k \leq n\}$

Technisch aufwendig, aber gar nicht so schwer:

$\{0,1,¢,\$\}^* \setminus L_1$ und $\{0,1,¢,\$\}^* \setminus L_2$ sind kontextfreie Sprachen.

Dann auch $L := \{0,1,¢,\$\}^* \setminus (L_1 \cap L_2)$ eine CFL.

Es gilt aber $L = \{0,1,¢,\$\}^*$ gdw. I keine Lösung hat. □

Das Sprachäquivalenzproblem für CFG

Definition

Das Sprachäquivalenzproblem für CFG:

- Eingabe: Zwei CFGs G_1 und G_2
- Frage: Ist $L(G_1) = L(G_2)$?

Theorem

Das Sprachäquivalenzproblem für CFG ist nicht berechenbar.

Beweis.

Gegeben $G_1 = (N, T, P, S)$. Konstruiere G_2 mit $L(G_2) = T^*$.

$L(G_1) = L(G_2)$ gdw. $L(G_1) = T^*$.

Wäre das Sprachäquivalenzproblem berechenbar, dann wäre auch das Universalitätsproblem berechenbar. □

Das Inklusionsproblem für CFG

Definition

Das Inklusionsproblem für CFG:

- Eingabe: Zwei CFGs G_1 und G_2
- Frage: Ist $L(G_1) \subseteq L(G_2)$?

Theorem

Das Inklusionsproblem für CFG ist nicht berechenbar.

Beweis.

Gegeben $G_1 = (N, T, P, S)$. Konstruiere G_2 mit $L(G_2) = T^*$.

$L(G_2) \subseteq L(G_1)$ gdw. $L(G_1) = T^*$.

Wäre das Inklusionsproblem berechenbar, dann wäre auch das Universalitätsproblem berechenbar. □

Entfernen von ϵ-Produktionen

Theorem

Es sei $G = (N, T, P, S)$ eine CFG mit $\epsilon \notin L(G)$.

Dann gibt es eine CFG G' ohne ϵ-Produktionen, die dieselbe Sprache wie G erzeugt.

Beweis.

Idee:

1. Finde alle nullierbaren Symbole $Z \subseteq N$.
2. Für jede Produktion $A \to \alpha B \gamma$ mit $B \in Z$, erzeuge zusätzlich eine neue Produktion $A \to \alpha\gamma$ (wiederhole dies rekursiv).
3. Streiche alle Produktionen der Form $A \to \epsilon$.

□

Beispiel

$$S \to AaS \mid b$$
$$A \to Sb \mid aAA \mid \epsilon$$
$$B \to AA \mid AB \mid BAa \mid b$$

Welche Symbole sind nullierbar?

A und B

$$S \to AaS \mid aS \mid b$$
$$A \to Sb \mid aAA \mid aA \mid a$$
$$B \to AA \mid AB \mid A \mid B \mid BAa \mid Ba \mid Aa \mid a \mid b$$

Definition

Eine Grammatik $G = (N, T, P, S)$ ist in *Chomsky-Normalform*, wenn sie nur Produktionen folgender Form enthält:

$$A \to a \text{ und } A \to BC,$$

wobei $a \in T$ und $A, B, C \in N$.

Beispiel:

$S \to R_a A \mid R_b B$
$A \to S R_a \mid R_a R_a \mid R_b R_a$
$B \to S R_b \mid R_a R_b \mid R_b R_b$
$R_a \to a$
$R_b \to b$

Transformation in CNF

1. ϵ-Produktionen entfernen
2. Neues Nonterminal R_a für jedes $a \in T$
3. Ersetze $A \to aBbC$ durch $A \to R_a B R_b C$
4. Neue Regel $R_a \to a$ für jedes $a \in T$
5. Ersetze $A \to B_1 B_2 B_3 \cdots B_k$ durch $A \to B_1 B_{23\ldots k}$ (neues Symbol)
   $B_{23\ldots k} \to B_2 B_{34\ldots k}$
   $B_{34\ldots k} \to B_3 B_{45\ldots k}$
   $\ldots$
   $B_{k-1,k} \to B_{k-1} B_k$

Jetzt fast in CNF. Aber es gibt noch „Kettenregeln“ $A \to B$.

Beispiel

$S \to aSa \mid bSb \mid a \mid b$

$S \to R_aSR_a \mid R_bSR_b \mid R_a \mid R_b$
$R_a \to a$
$R_b \to b$

$S \to R_aA$
$S \to R_bB$
$A \to SR_a$
$B \to SR_b$
$R_a \to a$
$S \to R_a$
$R_b \to b$
$S \to R_b$

Kettenregeln

So können wir Kettenregeln eliminieren:

Falls die Regel $A \to B$ existiert, dann füge für jede Produktion $C \to \alpha A \beta$ eine Produktion $C \to \alpha B \beta$ hinzu.

Füge zu $S \to A$ die Produktionen $S \to \beta$ mit $A \to \beta \in P$ hinzu.

Dann streiche alle Kettenregeln.

Theorem

Es gibt einen Algorithmus, der zu einer Grammatik G eine Grammatik G' in Chomsky-Normalform berechnet, wobei $L(G') = L(G) \setminus \{\epsilon\}$.

Beispiel

$$S \to R_a A$$
$$S \to R_b B$$
$$A \to S R_a$$
$$B \to S R_b$$
$$R_a \to a$$
$$S \to R_a$$
$$R_b \to b$$
$$S \to R_b$$

$$S \to R_a A \mid R_b B \mid a \mid b$$
$$A \to S R_a \mid R_a R_a \mid R_b R_a$$
$$B \to S R_b \mid R_a R_b \mid R_b R_b$$
$$R_a \to a$$
$$R_b \to b$$

Linksrekursion

Definition

Eine Regel $A \to A\alpha$ ist *linksrekursiv*.

Linksrekursion läßt sich eliminieren:
Für $A \in N$ seien

- $A \to A\alpha_1 \mid \cdots \mid A\alpha_k$ die linksrekursiven und
- $A \to \beta_1 \mid \cdots \mid \beta_l$ die anderen Regeln.

Ersetze $A \to A\alpha_1 \mid \cdots \mid A\alpha_k$ durch

1. $A \to \beta_1 Z \mid \cdots \mid \beta_l Z$ und
2. $Z \to \alpha_1 Z \mid \cdots \mid \alpha_k Z \mid \alpha_1 \mid \cdots \mid \alpha_k$,

wobei Z ein neues Symbol ist.

Beispiel

$S \to SS \mid AS \mid SB$

$S \to AS \mid ASZ$
$Z \to S \mid B \mid SZ \mid BZ$

Greibach-Normalform

Definition

Eine CFG ist in *Greibach-Normalform* (GNF) falls jede Regel von der Form $A \to aBCD\cdots$ ist, d.h. $A \to \beta$ mit $\beta \in TN^*$.

Theorem

Es sei G ein CFG mit $\epsilon \notin L(G)$. Dann gibt es eine CFG G' in GNF mit $L(G') = L(G)$.

Starte mit G in CNF ohne unnütze und nullierbare Symbole.

O.b.d.A. sei $N = \{A_1, \cdots, A_n\}$.

1. Schritt:

Ändere G so, daß es keine Regeln $A_i \to A_j\alpha$ mit $j \leq i$ gibt.
Nehmen wir an, dies gilt schon für $i = 1, \cdots, k$.

Sei l die kleinste Zahl, für die es eine Regel
$A_{k+1} \to A_l\alpha$ gibt.

Es seien $A_l \to \beta_1 \mid \cdots \mid \beta_m$ alle Regeln für A_l.
Ersetze $A_{k+1} \to A_l\alpha$ durch $A_{k+1} \to \beta_1\alpha \mid \cdots \mid \beta_m\alpha$.

Wiederhole dies, bis es keine Regel
$A_{k+1} \to A_l\alpha$ mit $l < k + 1$ mehr gibt.

Jetzt kann es noch Regeln $A_{k+1} \to A_{k+1}\alpha$ geben, die wir mit Hilfe eines neuen Symbols eliminieren (das größer ist).
Alle Regeln haben jetzt die Form $A_i \to A_j\alpha$ mit $j > i$ oder $A_i \to a\alpha$.
Wir ersetzen $A_i \to A_j\alpha$ durch $A_i \to \beta_1\alpha| \cdots |\beta_k\alpha$ falls $A_j \to \beta_1| \cdots |\beta_k$ bis jede rechte Seite mit einem Terminalsymbol beginnt.

Beispiel

$S \to AB \mid a$
$A \to AA \mid b$
$B \to AB \mid a$

$S \to AB \mid a$
$A \to b \mid bZ_1$
$B \to AB \mid a$
$Z_1 \to A \mid AZ_1$

$S \to AB \mid a$
$A \to b \mid bZ_1$
$B \to bB \mid bZ_1B \mid a$
$Z_1 \to A \mid AZ_1$

$S \to AB \mid a$
$A \to b \mid bZ_1$
$B \to bB \mid bZ_1B \mid a$
$Z_1 \to b \mid bZ_1 \mid bZ_1 \mid bZ_1Z_1$

$S \to bB \mid bZ_1B \mid a$
$A \to b \mid bZ_1$
$B \to bB \mid bZ_1B \mid a$
$Z_1 \to b \mid bZ_1 \mid bZ_1Z_1$

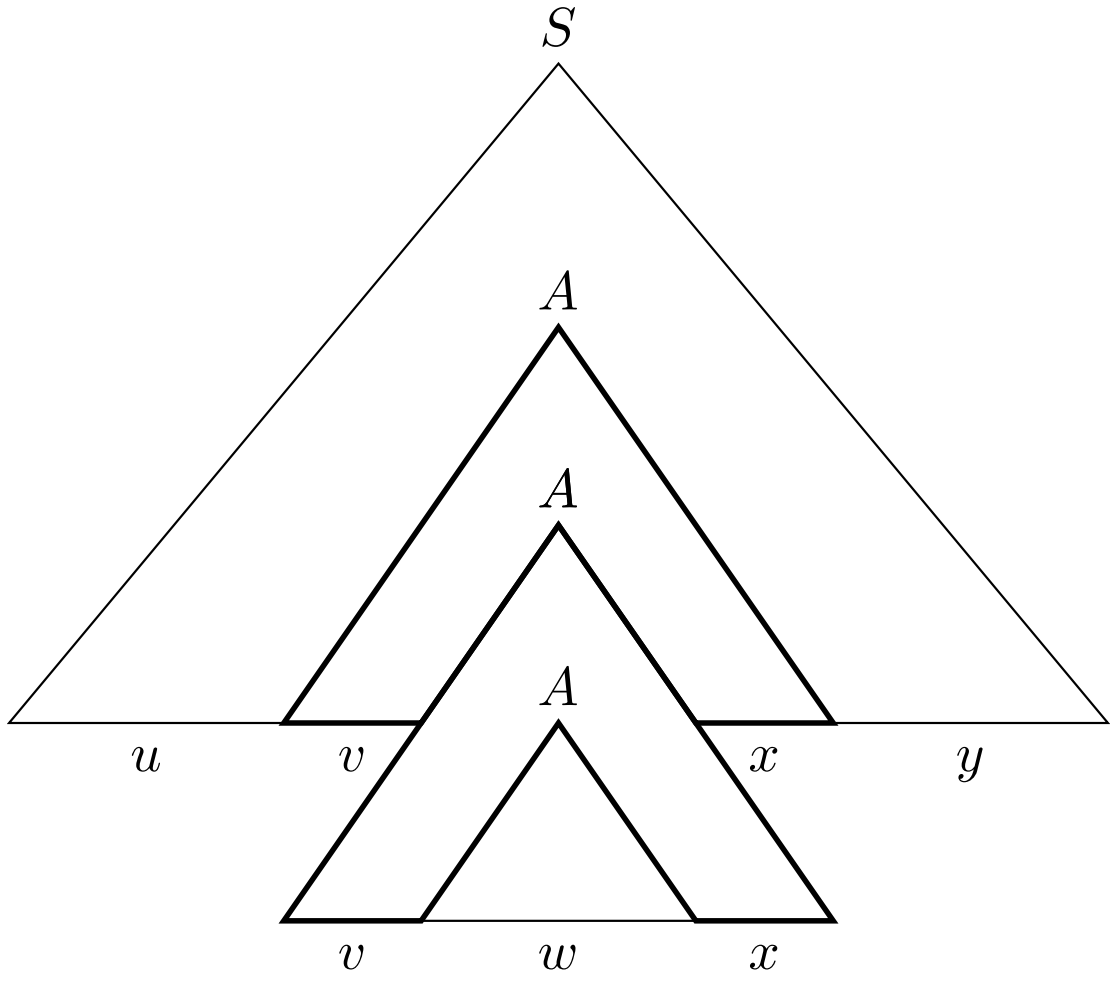
Das Pumping-Lemma für CFLs

Theorem

Für jede CFL L gibt es eine Zahl n für die gilt: Jedes Wort $z \in L$ mit $|z| > n$ hat eine Zerlegung $z = uvwxy$ mit

1. $|vwx| \leq n$
2. $|vx| > 0$
3. $uv^i wx^i y \in L$ *für jedes* $i \in \mathbf{N}_0$

Beweis.



Beispiel

$L = \{ a^n b^n c^n \mid n > 0 \}$.

Sei n die Konstante des Pumping-Lemmas.

$a^n b^n c^n = uvwxy$ mit $|vwx| \leq n$ und $|vx| > 0$.

Dann enthält vx kein a oder kein c.

Dann gilt leider $uv^2wx^2y \notin L$. Widerspruch!

Also ist L keine CFL.