Wortproblem für kontextfreie Grammatiken

G kontextfreie Grammatik. $w \in \Sigma^*$ $w \in L(G)$? Wortproblem ist primitiv rekursiv entscheidbar. (schlechte obere Schranke!)

Kellerautomat der $L(G)$ akzeptiert   Ist dieser effizient?

Problem:

- keine Endeutigkeit (mehrere Strukturbäume)
- Kellerautomat ist nicht-deterministisch.
- Falls deterministischer Kellerautomat möglich, so effizienter.

Beachte Beispiele:

- Boolesche Formeln über Signatur (PL-Formeln)
- Terme über Signatur
- Formeln über Signatur
- While Programme über Signatur

Mehrere Regeln mit gleicher linken Seite!

Verallgemeinerung der deterministischen Kellerautomaten

Mit Vorausschau $n \in \mathbb{N}$, falls in Abhängigkeit vom Kellerinhalt und den n-nächsten Eingabezeichen eindeutig die Möglichkeit besteht, die einzig richtige, als nächstes anzuwendende Produktion zu finden.

1-Vorausschau $\{a^n b^n : n \geq 1\}$

Schlagwort LR(k)-LL(k) Analyse.

7.45 Definition Normalformen für kontext-freie Grammatiken

Sei G eine kontext-freie Grammatik, G ist in

- **Chomsky-Normalform**: Produktionen der Form

$$A \rightarrow BC \text{ oder } A \rightarrow a \qquad A, B, C \in N, a \in T$$

- **Greibach-Normalform**: Produktionen der Form

$$A \rightarrow a\alpha \qquad A \in N, a \in T, \alpha \in N^*$$

7.46 Satz

Zu jeder kontextfreien Grammatik G mit $\varepsilon \notin L(G)$ gibt es eine kontextfreie Grammatik G' in Chomsky-Normalform, mit $L(G) = L(G')$.

Die Transformation $G \rightsquigarrow G'$ ist effektiv.

Beweisidee

Beweisidee: $G = (N, T, \Pi, Z)$ kontextfrei.

1. Schritt: **ε-frei**: Da $\varepsilon \notin L(G)$, gibt es eine äquivalente Grammatik G', die keine Produktionen der Form $A \longrightarrow \varepsilon$ enthält.

2. Schritt: **Normierte Terminierung**: Zu $G = (N, T, \Pi, Z)$ gibt es eine äquivalente Grammatik $G' = (\tilde{N}, T, \Pi', Z)$, die nur Produktionen $A \longrightarrow a$ mit $a \in T$ und $A \longrightarrow \alpha$ mit $\alpha \in \tilde{N}^*$ enthält.
Sei $\tilde{N} = N \cup \{A_a : a \in T\}$. Π' entsteht aus Π indem jedes $a \in T$ in Π durch A_a ersetzt wird, vereinigt mit $\{A_a \longrightarrow a : a \in T\}$ Platzhalter.

3. Schritt: **Keine Kettenproduktionen**: Zu einer Grammatik $G = (N, T, \Pi, Z)$ gibt es eine äquivalente Grammatik $G' = (N, T, \Pi', Z)$, die keine Produktionen der Form $A \longrightarrow B$ mit $A, B \in N$ enthält (siehe NEA).
Sei $M = \{(A, B) \in N^2 : A \underset{G}{\vdash} B\}$
(lässt sich berechnen: Entferne Zyklen $(A, B), (B, A) \in M$. Beginne mit (A, A).)
$\Pi' = \Pi \backslash \{A \longrightarrow B : A, B \in N\}$ vereinigt mit Produktionen $A \longrightarrow r', |r'| > 1$, die aus Produktionen $A \longrightarrow r \in \Pi$ durch Ersetzen mancher B in r durch C mit $(B, C) \in M$ entstehen, vereinigt mit $A \longrightarrow a$ für alle $(A, A_a) \in M$.

Beweisidee (Forts.)

4. Schritt: Chomsky-Normalform: Produktionen der Form $A \to B_1 \ldots B_n$, $n > 2$ ersetzen: Dazu $A \to B_1 H_1, H_1 \to B_2 H_2, \cdots, H_{n-3} \to B_{n-2} H_{n-2}, H_{n-2} \to B_{n-1} B_n$ mit neuen Nichterminalsymbolen $H_1, \ldots, H_{n-2}$.

Falls $\varepsilon \in L(G)$, so ist $(N \cup \{Z'\}, T, \Pi \cup \{Z' \to Z, Z' \to \varepsilon\}, Z')$, wobei (N, T, Π, Z) in Chomsky-Normalform.

7.47 Beispiel Sei $G = (\{Z, A, B\}, \{a, b\}, \Pi, Z)$

$$\begin{array}{lll} \Pi: & Z \to bA & Z \to aB \\ & A \to a & B \to b \\ & A \to aZ & B \to bZ \\ & A \to bAA & B \to aBB \end{array}$$

1. Schritt: **ε-frei**: ok.

2. Schritt:

$$\begin{array}{ll} Z \to A_b A \mid A_a B & \\ A \to a,\ B \to b,\ A_a \to a,\ A_b \to b & \\ A \to A_a Z & B \to A_b Z \\ A \to A_b A A & B \to A_a B B \end{array}$$

3. Schritt: Keine Kettenproduktionen: ok.

Wortproblemalgorithmen für k.f. Sprachen

4. Schritt: Letzte Zeile oben:

$$A \to A_b C_1, C_1 \to AA$$
$$B \to A_a D_1, D_1 \to BB$$

Auswirkungen auf Strukturbaum? (binär)

$\rightsquigarrow$ Pumping Lemma Konstante: $2^{|N|} + 1$.

$x \in L(G) \rightsquigarrow x$ ist in höchstens $2|x| + 1$ Schritten in G ableitbar (exponentieller Aufwand für Entscheidung $x \in L(G)$).

7.5 Algorithmus von Cocke-Kasami-Younger

7.48 Satz

Sei G in Chomsky-Normalform. Dann gibt es einen Algorithmus der das Wortproblem für G mit Laufzeit $O(n^3)$ entscheidet.

$w \in L(G) \qquad |w| = n$ Laufzeit $O(n^3)$

Beweis: Sei $w = a_1 \dots a_n$,
$L_{ij}(w) = \{A \in N : A \underset{G}{\vdash} a_i \dots a_j\} \qquad (i \leq j)$

Es gilt $w \in L(G)$ gdw $Z \in L_{1n}(w)$.

Wie berechnet man aus w die L_{ij}. **Dynamisches Programmieren**.

Induktiv über $j - i$ Berechnung von L_{ij}:

- $j - i = 0 : L_{jj} = \{A : A \to a_j \in \Pi\}$
  (da Chomsky-Normalform)

Algorithmus von Cocke-Kasami-Younger

- $j - i > 0$: Berechne L_{ij} aus L_{ik-1} und L_{kj} für ein k mit $i < k \leq j$, wobei $A \in L_{ij}$, falls $A \rightarrow BC \in \Pi$, $B \in L_{ik-1}$, $C \in L_{kj}$.

In jedem Schritt müssen maximal $2n$ Mengen betrachtet werden und es gibt weniger als n^2 Mengen L_{ij}, daher kann die Laufzeit durch cn^3 beschränkt werden, wobei c eine Konstante ist, die von der Grammatik G abhängt.

Verwaltung mithilfe einer Erkennungs-Matrix

$j-i$ $\quad i$	1	2	$\ldots$	n
0	$\{\}$	$\{\}$		$\{\}$
1	$\{\}$	$\ldots$	$\{\}$	
2	$\vdots$			
$\vdots$	$\vdots$			
$n-1$	$\{\}$			

Beispiel:

$Z \rightarrow CB \mid FA \mid FB$
$A \rightarrow CZ \mid FD \mid a$
$B \rightarrow FZ \mid CE \mid b$
$D \rightarrow AA, E \rightarrow BB, C \rightarrow a, F \rightarrow b$

$w = aababb$, $|w| = 6$

Teilw. Länge	$j-i$	$i=1$	2	3	4	5	6	
1	0	A, C	A, C	B, F	A, C	B, F	B, F	(n Einträge)
2	1	D	Z	Z	Z	E, Z		($n-1$ Eintr.) Kosten 1
3	2	A	A	B	A, B			($n-1$ Eintr.) Kosten 2
4	3	D	Z	Z, E				$\vdots$
5	4	A	A, B					$\vdots$
6	5	D, Z						1-Eintrag Kosten $n-1$

$$n + \sum_{i=2}^{n} (n-i+1)(i-1) = \frac{n^3+5}{6}$$

Auf Mehrband TM mit Zeit n^3 realisierbar. Siehe z. B. Hopcroft/Ullman Automaten + formale Sprachen.
Viele Verbesserungen: Mit Einschränkungen oft $O(n)$ möglich! (Vorausschau 1 Det.).

7.6 Unentscheidbare Probleme für kontextfreie Grammatiken

Unentscheidbare Probleme für allgemeine Grammatiken

- Wortproblem
- $L(G) = \varnothing$
- $L(G) = \Sigma^*$
- $L(G)$ endlich
- $L(G_1) = L(G_2)$
- $\varepsilon \in L(G)$?

Für rechts-lineare-Grammatiken alle entscheidbar.

Für kontexfreie Grammatiken? Wortproblem, $L(G)$ endlich ?, $L(G) = \varnothing?, \varepsilon \in L(G)?$ entscheidbar.

7.49 Satz

Sind G_1, G_2 kontextfreie Grammatiken.
Es ist unentscheidbar, ob die zugehörigen Sprachen disjunkt sind.

Folgendes Problem ist nicht rekursiv entscheidbar:

Eingabe: kontextfreie Grammatiken G_1, G_2.

Frage: $L(G_1) \cap L(G_2) \neq \varnothing$?

Unentscheidbare Probleme für kontextfreie Grammatiken (Forts.)

Beweis: Reduktion des PCP auf dieses Problem.

Sei $\mathcal{L} = (u_1 \sim v_1, \ldots, u_k \sim v_k)$, $u_i, v_i \in \Gamma^+$, $k \geq 1$.

Sei $J = \{1, \ldots, k\}$, $J \cap \Gamma = \varnothing$.

Definiere Grammatiken $G_j = (N_j, T, \Pi_j, Z_j)$, $j = 1, 2$.

$T = \Gamma \cup J$, $N_j = \{Z_j\}$

$\Pi_1 = \{Z_1 \to u_i Z_1 i \mid u_i i : i = 1, \ldots, k\}$ $2k$-Regeln

$\Pi_2 = \{Z_2 \to v_i Z_2 i \mid v_i i : i = 1, \ldots, k\}$ $2k$-Regeln

$$
\begin{aligned}
L(G_1) \cap L(G_2) \neq \varnothing \quad & \text{gdw } \exists x \in \Sigma^* \;\; x \in L(G_1) \cap L(G_2) \\
& \text{gdw } \exists t_1, t_2 \in J^* \\
& \qquad x = U(t_1) t_1^{mi} = V(t_2) t_2^{mi} \\
& \text{gdw } \exists t \in J^+ \;\; x = U(t) t^{mi} = V(t) t^{mi} \\
& \text{gdw } \exists t \in J^+ \;\; U(t) = V(t) \\
& \text{gdw } PCP(\mathcal{L})
\end{aligned}
$$

Beachte:

Die Konstruktion liefert „einfache“ k.f. Grammatiken G_1 und G_2:
Sie sind **linear** ($A \to uBv$ Regeln) und **eindeutig**: nur eine Linksableitung möglich!

Folgerungen

- Es gibt kein effektives Verfahren, um für zwei kontextfreie Grammatiken G_1, G_2 eine kontextfreie Grammatik G zu bestimmen mit $L(G) = L(G_1) \cap L(G_2)$. (Begründung: $L(G) \neq \varnothing$ ist für kontextfreie Grammatiken entscheidbar).
- Man kann jedoch eine kontextsensitive Grammatik berechnen mit $L(G) = L(G_1) \cap L(G_2)$, d. h. $L(G) \neq \varnothing$ ist nicht entscheidbar für kontextsensitive Grammatiken.

7.50 Satz Das Mehrdeutigkeitsproblem für kontextfreie Grammatiken ist unentscheidbar.

Eingabe: G kontextfreie Grammatik.

Frage: Ist G mehrdeutig?

Beweis:

PCP auf Mehrdeutigkeitsproblem reduzieren: Seien G_1 und G_2 die kontextfreien Grammatiken wie oben zu PCP $\mathcal{L}$ konstruiert.

$$G_{\mathcal{L}} := (\{Z, Z_1, Z_2\}, \Gamma \cup J, \Pi_1 \cup \Pi_2 \cup \{Z \to Z_1, Z \to Z_2\}, Z)$$

$\mathcal{L} \rightsquigarrow G_{\mathcal{L}}$ effektiv. $L(G_{\mathcal{L}}) = L(G_1) \cup L(G_2)$

G_1, G_2 sind eindeutig.

$G_{\mathcal{L}}$ ist mehrdeutig $\text{gdw } L(G_1) \cap L(G_2) \neq \varnothing$
$\text{gdw } PCP(\mathcal{L})$

Weitere unentscheidbare Probleme für kontextfreie Grammatiken

7.51 Satz

Folgende Probleme für kontextfreie Grammatiken sind nicht entscheidbar.

1)	$P_1(G)$	gdw $L(G) = \Sigma^*$ (G über $\Sigma = T$)
2)	$P_2(G)$	gdw $L(G)$ ist rechts-linear
3)	$P_3(G)$	gdw $\neg L(G)$ ist kontextfrei (rechts-linear, unendlich)
4)	$P_4(G)$	gdw $L(G_1) = L(G_2)$ (beide über T)
5)	$P_5(G)$	gdw $L(G_1) \subseteq L(G_2)$
6)	$P_6(G_1, G_2)$	gdw $L(G_1) \cap L(G_2)$ ist kontextfrei
7)	$P_7(G_1, G_2)$	gdw $L(G_1) \cap L(G_2)$ unendlich
8)	$P_8(G_1, G_2)$	gdw $L(G_1) \cap L(G_2)$ RL-Sprache

$\mathcal{L}_{\text{endl}} \subsetneq \mathcal{L}_3 \subsetneq \mathcal{L}_{\text{det-kf}} \subsetneq \mathcal{L}_2 \subsetneq \mathcal{L}_1 \subsetneq \mathcal{L}_{\text{prim-rek}} \subsetneq \mathcal{L}_{\text{entsch.}} \subsetneq \mathcal{L}_0 = \mathcal{L}_{\text{rek-aufzb.}}$

Kontextsensitive Grammatiken und Sprachen

Erinnerung: Die Sprache $\{a^n b^n c^n : n \in \mathbb{N}\}$ ist eine Typ-1 Sprache: Grammatik $(\{Z, A, B, H, C\}, \{a, b, c\}, \Pi, Z)$ mit Produktionen $\Pi = \{Z \to \varepsilon \mid Ac, A \to ab \mid aACB, CB \to CH, CH \to BH, BH \to BC, B \to b, Cc \to cc\}$.
Sie ist nicht kontexfrei.

Das Wortproblem für k.s. Grammatiken ist entscheidbar. Man muss nur Ableitungen bis zur Länge $(|N| + |T| + 1)^{|x|} + 1$ durchsuchen (ansonsten enthält die Ableitung zwei identische Wörter mit Länge $\leq |x|$).

Ein **linear beschränkter Automat (LBA)** ist eine (nichtdeterministische) Turing-Maschine, deren Lese-/Schreibkopf den Bereich, auf dem beim Start die Eingabe steht, nicht verlassen darf.
• Die Typ-1-Sprachen sind genau die Sprachen, die sich mit einem LBA akzeptieren lassen.

Solche Automaten lassen sich auch durch Produktionen Charakterisieren. Sie haben die Gestalt:
$qa \to q'a' \qquad q, q' \in Q, a, a' \in N \cup T$
$qa \to aq' \qquad q, q' \in Q, a, a' \in N \cup T$
$bqa \to q'ba \qquad q, q' \in Q, a, a' \in N \cup T$
letztere für alle $b \in N \cup T$ falls keine andere diese linke Seite hat.

Kontextsensitive Grammatiken und Sprachen (Fort.)

Es gibt weitere Charakterisierungen der k.s. Sprachen durch spezielle Grammatiken. Eine Grammatik $G = (N, T, \Pi, Z)$ heißt **erweiternd**, falls Π nur Produktionen der Form $l \longrightarrow r$ mit $l \neq \varepsilon$ und $|r| \geq |l|$ enthält. Es gilt: Zu jeder erweiternden Grammatik gibt eine äquivalente k.s. Grammatik.