A szóprobléma eldöntése

Formális nyelvek, 8. gyakorlat

Célja: A szóprobléma megoldásának bemutatása a különböző Chomsky-osztályok esetén

Fogalmak: i. típusú nyelvtanok, Chomsky-normálforma, szóprobléma, elemzés, összes levezetések gráfja, szélességi bejárás, mélységi bejárás, gyakorlati megoldhatóság, CYK algoritmus, véges determinisztikus automata (VDA), az automaták megadásának lehetőségei.

Feladatok jellege: Konkrét nyelvtanra az összes levezetések gráfjának felírása, az utak és a levezetések közötti összefüggés bemutatása, a szélességi bejárás algoritmusának elkészítése struktogramként, a CYK algoritmus egy példán keresztül, utalva a műveleti igényre, automatakészítés néhány egyszerű nyelv esetében.

2005/06 II. félév

Összes levezetések gráfja

1.feladat:

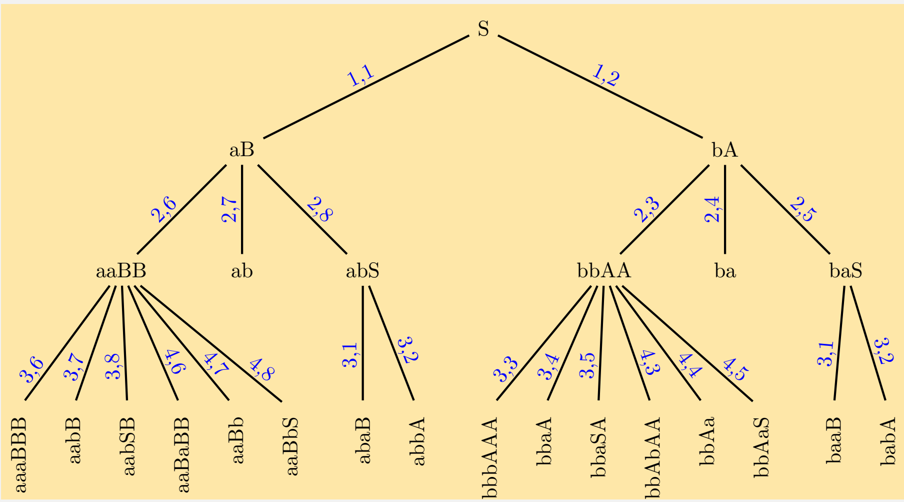
$S \to bA \mid aB$
$A \to bAA \mid a \mid aS$
$B \to aBB \mid b \mid bS$

Rajzoljuk fel G összes levezetéseinek gráfjából az S által meghatározott rész első 3 szintjét!

Összes levezetések gráfja

1. 2. 3. 4. 5. 6. 7. 8.

$S \to bA \mid aB$ $\quad A \to bAA \mid a \mid aS$ $\quad B \to aBB \mid b \mid bS$



A levezethető szavakat kiíró algoritmus

Valamely $G = \langle T, N.\mathcal{P} = \{p_1 \to q_1, \ldots, p_k \to q_k\}, S\rangle \in \mathcal{G}_0$ nyelvtan esetén a következő algoritmus szintfolytonosan bejárja az összes levezetések gráfjának $\alpha_0 \in (T \cup N)^*$-ból levezethető részét és a levezethető szavakat kiírja.

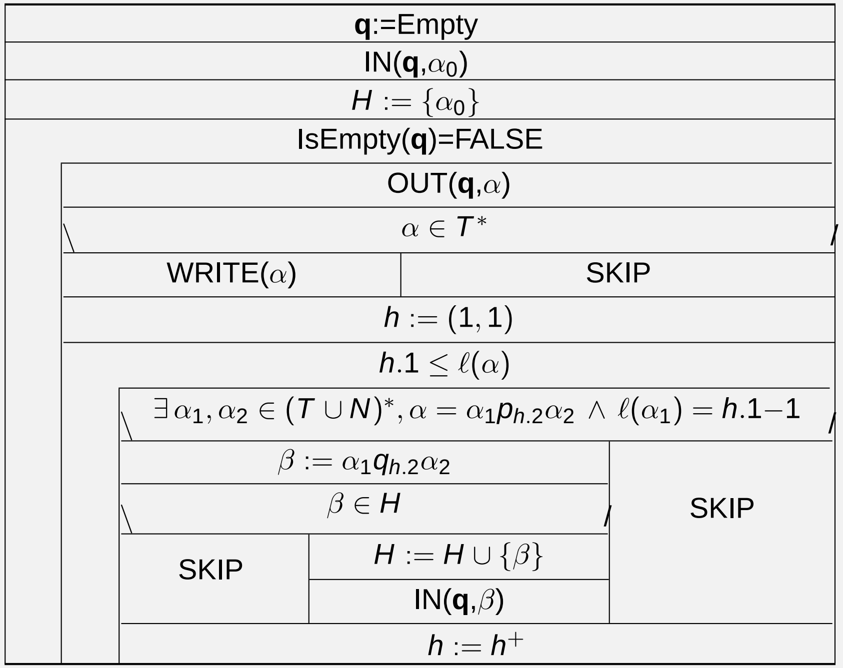
Jelölések: A $h = (i, j)$ rendezett pár első, illetve második koordinátájára úgy hivatkozunk, hogy $h.1$, illetve $h.2$.

Ha $h = (h.1, h.2)$ akkor legyen h^+ a következő:

$$h^+ = \begin{cases} (h.1, h.2+1) & h.2 < k \\ (h.1+1, 1) & h.2 = k \\ \text{nem definiált} & \text{egyébként} \end{cases}$$

Az algoritmus használja a sor adatszerkezetet.

IN(**q**,x)	beteszi az x elemet a **q** sor végére,
OUT(**q**,x)	kiveszi az első elemet a **q** sor elejéről, x-nek ezt az értéket adja és törli a sorból.
Empty	az üres sor,
IsEmpty(**q**)	TRUE/FALSE attól függően, hogy a **q** sor üres-e.



A szóprobléma eldöntése

2.feladat:

Hogyan kell módosítani az előbbi algoritmust, ha egy adott $u(\in T^*)$ szóról szeretnénk eldönteni, hogy

a. benne van-e $L(G)$-ben?

b. benne van-e $L(G)$-ben, és ha igen mi egy levezetése?

c. Mit kell módosítani az előbbi algoritmuson hosszúságot nem csökkentő nyelvtan esetén, ha azt akarjuk, hogy mindig termináljon?

Szóprobléma eldöntése

0. típusú nyelvtanok

Megoldás:

a. IF $\alpha \in T^*$ THEN WRITE(α) helyett:
IF $\alpha = u$ THEN RETURN(Igen)

b. Csinálunk egy $A[i,j]$ tömböt is, $1 \leq j \leq 4$. A tömb egy sorának típusa $(T \cup N)^* \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ legyen. Amikor **q**-ba bekerül egy β elem, melyet az $\alpha = A[m,1]$-ból vezettünk le, akkor a legkisebb olyan h indexre melyre A még nem definiált legyen $A[h,1] := \beta$, $A[h,2]$ ahanyadik betűtől, $A[h,3]$ pedig ahanyadik szabály helyettesítésével kapjuk α-ból β-t. Végül legyen $A[h,4] := m$.

Az algoritmusunkat továbbá úgy módosítjuk, hogy ha $\alpha = u$, akkor előállítjuk a levezetést is, fordított sorrendben. Az A tömb tartalmazza azt az információt, mellyel megkapható a levezetés.

Szóprobléma eldöntése

1. típusú nyelvtanok

c. Az első $f(t) := \sum_{i=0}^{t} |(T \cup N)|^i$ szintet bejárva, az eredeti algoritmus kiírja az összes legfeljebb t hosszú levezethető szót. Számolja a módosított algoritmusunk tehát azt, hogy hanyadik szinten járunk! A **q** sorba tehát először ne α_0-t, hanem $(\alpha_0, 0)$-t írjunk, illetve β helyett a $(\beta, t+1)$ párt tegyük be, ha (α, t)-t vettünk ki.

Az első WHILE ciklus addig menjen csak amíg a kivett elem második koordinátája legfeljebb $f(\ell(u))$. Ha eddig nem vezettük le u-t, akkor már nem is fogjuk, álljunk le "Nem" válasszal.

CYK algoritmus
A szóprobléma eldöntése 2. típusú nyelvtanokra

Cocke-Younger-Kasami (CYK) algoritmus:

Adott egy környezetfüggetlen $G = \langle T, N, \mathcal{P}, S \rangle$ nyelvtan Chomsky-normálformában adva.
Az algoritmus adott $u \in T^*$ esetén eldönti, hogy "$u \in L(G)$"-e.

Legyen $u = t_1 \dots t_n$, $t_i \in T$. Jelölje α_i a $P_i \in \mathcal{P}$ szabály bal-, β_i pedig a jobboldalát. ($\alpha_i \in N$, $\beta_i \in T \cup N^2$.)

A CYK algoritmus rekurzíven definiál $H_{i,j}$, $1 \le i \le j \le n$ halmazokat $(j-i)$ szerint növekvő sorrendben.

$$H_{i,i} := \{\alpha_j \mid \beta_j = t_i\}$$

$$H_{i,j} := \{\alpha_k \mid \beta_k \in \bigcup_{h=i}^{j-1} H_{i,h}H_{h+1,j}\} \quad (i < j)$$

Ha $S \in H_{1,n}$, akkor $u \in L(G)$, különben $u \notin L(G)$.

CYK algoritmus

3.feladat:

Elemezzük CYK algoritmussal az *aabbcc* szót az alábbi G nyelvtan esetén:

$S \to AB \mid BC$
$A \to XA \mid a$
$X \to a$
$C \to YC \mid c$
$Y \to c$
$B \to UV \mid VW$
$U \to XX$
$W \to YY$
$V \to ZZ$
$Z \to b$

CYK algoritmus
CYK tábla

Megoldás:

$\{S\}$					
$\{S\}$	$\{S\}$				
$\{B\}$	$\emptyset$	$\{B\}$			
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$		
$\{A,U\}$	$\emptyset$	$\{V\}$	$\emptyset$	$\{C,W\}$	
$\{A,X\}$	$\{A,X\}$	$\{Z\}$	$\{Z\}$	$\{Y,C\}$	$\{Y,C\}$
a	a	b	b	c	c

Tehát $aabbcc \in L(G)$.

Véges determinisztikus automaták
A szóprobléma eldöntése 3. típusú nyelvtanokra

Véges determinisztikus automata (VDA) alatt a következő 5-öst értjük:
$\mathcal{A} = \langle A, T, \delta, a_0, F \rangle$, ahol

A	az állapotok (véges) halmaza
T	egy ábécé, a bemenő ábécé
$\delta : A \times T \to A$	az állapotátmeneti függvény
$a_0 \in A$	kezdőállapot
$F \subseteq A$	a végállapotok halmaza.

A VDA egy ütemben kiolvassa a központi egység állapotát, az input szó aktuális szimbólumát, ennek függvényében új állapotba kerül és az input szó következő betűjére áll az olvasófej (azaz, jobbra lép).

Konfiguráció (amitől a VDA további működése függ): $[a, v]$, $a \in A$, $v \in T^*$, a az aktuális állapot, v az input szó még olvasatlan része.

Az $u \in T^*$ input szóhoz tartozó kezdõkonfiguáció: $[a_0, u]$.

Véges determinisztikus automaták
A szóprobléma eldöntése 3. típusú nyelvtanokra

Közvetlen konfiguációátmenet: $[a,u] \underset{\mathcal{A}}{\dashv} [b,v]$, ha $u = tv$, és $b = \delta(a,t)$.

Közvetett konfigurációáatmenet: a közvetlen konfigurációátmenet tranzitív, reflexív lezártja, jelölése: $\overset{*}{\underset{\mathcal{A}}{\dashv}}$.

$[a,v]$ elfogadó konfiguráció: ha $a \in F$ és $v = \varepsilon$

$\mathcal{A}$ elfogad egy u szót ha az u-t végig beolvasó konfigurációátmenet elfogadó konfigurációval terminál.

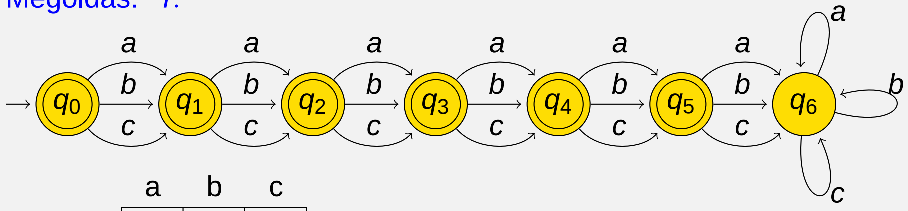
VDA-k és 3. típusú nyelvtanok kölcsönösen megfeleltethetők egymásnak úgy, hogy az elfogadott nyelv illetve a generált nyelv ugyanaz lesz.

Tehát VDA-k remekül használhatók 3. típusú nyelvtanok szóproblémáinak eldöntésére.

Véges determinisztikus automaták
VDA-k megadása

3. Feladat: $T = \{a,b,c\}$. Adjunk VDA-t mely a legfeljebb 5 hosszú szavakat fogadja el!

Megoldás: *I.*



II.

	a	b	c
$\leftrightarrow q_0$	q_1	q_1	q_1
$\leftarrow q_1$	q_2	q_2	q_2
$\leftarrow q_2$	q_3	q_3	q_3
$\leftarrow q_3$	q_4	q_4	q_4
$\leftarrow q_4$	q_5	q_5	q_5
$\leftarrow q_5$	q_6	q_6	q_6
q_6	q_6	q_6	q_6

Házi feladat

1. Elemezzük CYK algoritmussal az 3. feladatban adott nyelvtan esetén a következő 2 szót: *aaabcc* és *abbccc*.
2. Adjunk VDA-t mely a legalább 4 hosszú szavakat fogadja el!
3. Adjunk VDA-t mely a 7-tel osztható számokat fogadja el!