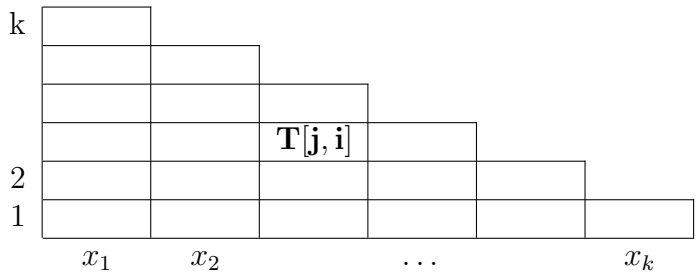
Cocke-Younger-Kasami algoritmus, a levezetési fák nyomon követésével

Az algoritmus egy CNF formára alakított G nyelvtannal indul és egy adott x szóhoz meghatározza, hogy x levezethető-e a nyelvtanból. Azt is megkaphatjuk, hogy ha $x \in L(G)$, akkor a levezetése egyértelmű-e és meg tudjuk határozni x összes levezetési fáját is.

Egy T táblázatot hozunk létre, ennek oszlopai a szó $x_1, x_2, \ldots, x_k \in \Sigma$ betűinek felelnek meg, sorai pedig a $j = 1, 2, \ldots, k$ indexeknek. Igazából a táblázatnak csak az átlóját, és az az alatti részt használjuk, a $T[j, i]$-be azok a változók kerülnek, melyekből az $x_i x_{i+1} \ldots x_{i+j-1}$ szórészlet levezethető. (Úgy is mondhatjuk, hogy azok a nemterminálisok kerülnek a $T[j, i]$ cellába, melyekből az x_i-vel kezdődő, j hosszú részszó levezethető.)



A táblázat kitöltése soronként lentről felfelé történik:

- $j = 1$: az A változó bekerül a $T[1, i]$ cellába, ha $A \to x_i$ szabály G-ben
- $j > 1$: az A változó bekerül a $T[j, i]$ cellába, ha $A \to BC$ szabály G-ben, $B \in T[\ell, i]$ és $C \in T[j - \ell, i + \ell]$, valamilyen $1 \leq \ell \leq j - 1$-re. Ez azt jelenti, hogy az A változó úgy generálja az $x_i x_{i+1} \ldots x_{i+j-1}$ részszót, hogy az $A \to BC$ szabállyal olyan B és C változókat kapunk, hogy B a j hosszú $x_i x_{i+1} \ldots x_{i+j-1}$ részszó első ℓ karakterét, C pedig a maradék $j - \ell$ karaktert állítja elő.

Világos, hogy ezzel a kitöltési eljárással a $T[j, i]$ pontosan azokat a változókat tartalmazza, amelyekből a megfelelő részszó levezethető. El tudjuk tehát dönteni, hogy az

input x szó S-ből levezethető-e vagy sem, mert amennyiben a kezdőváltozó megjelenik a $T[k,1]$ (bal felső) cellában, akkor $x \in L(G)$, különben $x \notin L(G)$.

Ahhoz, hogy a levezetési fákat is meg tudjuk keresni, a táblázat kitöltése közben (a második sortól kezdve) további dolgokat kell tárolnunk. A nyelvtan $A \to BC$ alakú szabályait megszámozzuk és amikor egy A változó egy cellába bekerül, akkor két indexet fog kapni: az első azt mutatja, hogy hányas szabály alapján került ide (hányas volt az az $A \to BC$ szabály, ami miatt az A változó az adott cellába került), a második index pedig azt fogja mutatni, hogy az első index által jelzett szabály jobb oldalán elöl álló B változót hányadik sorban találjuk az adott oszlopon belül (ebből már egyértelmű, hogy C-t hol kell keresnünk).

A levezetési fák ezen információk segítségével meghatározhatók (lásd a példát a fejezet végén), így azt is el tudjuk majd dönteni, hogy x-et egyetlen levezetési fa segítségével lehet-e megkapni vagy sem.

Az algoritmus lépésszáma egy k hosszú szó esetén:

- cellánként: $\leq k$ esetet kell ellenőrizni
- a táblázat mérete: k^2
- összesen: $O(k^3)$

Megjegyezzük, hogy vannak a gyakorlatban ennél jobban használható eljárások is, amelyekre a fordítóprogramok is épülnek.

Példa Legyen

$$\begin{aligned} S &\to AB \mid BC \\ A &\to BA \mid a \\ B &\to CC \mid b \\ C &\to AB \mid a \end{aligned}$$

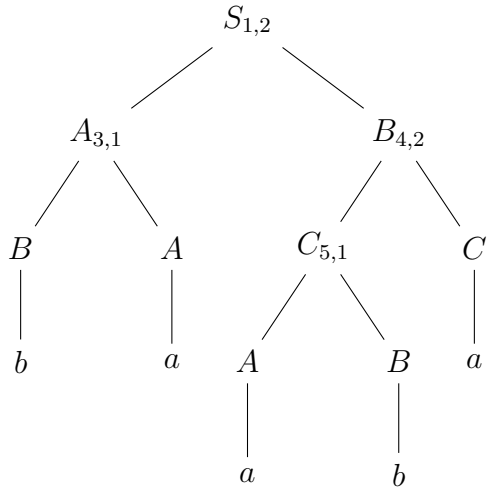
a nyelvtan és $x = baaba$ a szó.

A táblázat alsó sorának kitöltése egyszerű. Vegyük a $T[2,1]$ elemet. Ehhez egy megvizsgálandó cellapár tartozik: $T[1,1]$ és $T[1,2]$. Keressük azokat a szabályokat, amik ezen cellapárok változóit tartalmazzák a jobb oldalon (az adott sorrendben), vagyis $X \to BA$, illetve $X \to BC$ alakúak. A nyelvtan alapján $X = A$ vagy $X = S$, a cella tartalma pedig $A_{3,1}, S_{2,1}$ lesz, mert az A változó a 3. szabály alapján került a cellába és az ezen szabály jobb oldalán elöl álló B az 1. sorban található, az S változó pedig a 2. szabály miatt jutott ide, melynek jobb oldali első karaktere szintén az első sorban van. A $T[3,1]$-hez két cellapár tartozik: $T[2,1]$–$T[1,3]$ és $T[1,1]$–$T[2,2]$, azaz a $\{A,S\}\{A,C\}$ vagy $\{B\}\{B\}$ halmazokból kiolvasható AA, AC, SA, SC, BB párokhoz keresünk szabályt. Ilyen viszont nincs, ezt jelzi a kihúzás a táblázatban. A többi cella hasonló megfontolásokkal kitölthető.

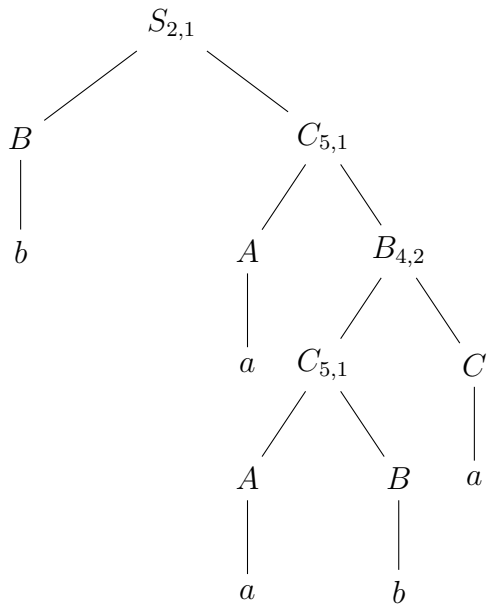
$S_{1,2}, S_{2,1}, A_{3,1}, C_{5,2}$				
–	$S_{1,1}, S_{2,3}, A_{3,2}, A_{3,3}, C_{5,1}$			
–	$B_{4,1}$	$B_{4,2}$		
$A_{3,1}, S_{2,1}$	$B_{4,1}$	$C_{5,1}, S_{1,1}$	$A_{3,1}, S_{2,1}$	
B	A, C	A, C	B	A, C
b	a	a	b	a

A táblázatból azonnal kiolvasható, hogy az x szó a megadott nyelvtanból levezethető, mert a bal felső cellában szerepel a kezdőszimbólum. Az is látszik rögtön, hogy biztosan lesz legalább két levezetési fa ($S_{1,2}$ és $S_{2,1}$ kezdéssel), vagyis a *baaba* szónak egynél több levezetési fája van.

Maguk a levezetési fák az alábbi módon kaphatók meg. Az első levezetési fa gyökere $S_{1,2}$ lesz, az indexek pedig megmutatják, hogy $S_{1,2}$ két gyereke az $S \to AB$ szabály miatt A és B lesz, ahol az A nemterminális a második sorban levő $A_{3,1}$, az ehhez tartozó B pedig ekkor csak a 3. sor, 3. oszlopában levő $B_{4,2}$ lehet. Ugyanezen logikát követve a most megjelent $A_{3,1}$ és $B_{4,2}$ (és a később előkerülő újabb nemterminálisok) kifejtésére, kapjuk a következő levezetési fát:



Az $S_{2,1}$ kezdéssel pedig a következő fát kapjuk:


$S_{2,1}$
B
$C_{5,1}$
b
A
$B_{4,2}$
a
$C_{5,1}$
C
A
B
a
a
b