LABORATOR NR 3

ANALIZA SINTACTICĂ ASCENDENTĂ ŞI DE TIP KASAMI

Analiza sintactică ascendentă

Teoremă. Fie $G=(N,\Sigma,P,S)$ o gramatică independentă de context cu producțiile numerotate de la 1 la p. Definim traducătorul pushdown determinist $\mathcal{P}=(\{q\};\Sigma;N\cup\Sigma\cup\{\$\};\{1,2,...,p\};\delta;q;\$;\phi)$ unde:

1. $\delta(q,\varepsilon,\alpha)=\{(q,A,i)\mid \quad i:(A\to\alpha)\in P\}\ \forall A\in N$
2. $\delta(q,a,\varepsilon)=\{(q,a,\varepsilon)\}\quad \forall a\in\Sigma$
3. $\delta(q,\varepsilon,\$S)=\{(q,\varepsilon,\varepsilon)\}$

Atunci $\tau_\varepsilon(\mathcal{P})=\{(w,\pi)\mid S\overset{\tilde{\pi}}{\underset{G}{\Rightarrow}}w\}$.

Aplicație. Fie gramatica expresiilor aritmetice simple:

1. $E\to E+T$
2. $E\to T$
3. $T\to T*F$
4. $T\to F$
5. $F\to(E)$
6. $F\to a$

Se construieşte traducătorul pushdown nedeterminist:

$\mathcal{P}=(\{q\};\{+,*,(,),a\};\{E,T,F,+,*,(,),a\};\{1,2,...,6\};\delta;q;E;\phi)$

1. $\delta(q,\varepsilon,E+T)=\{(q,E,1)\}$
2. $\delta(q,\varepsilon,T)=\{(q,E,2)\}$
3. $\delta(q,\varepsilon,T*F)=\{(q,T,3)\}$
4. $\delta(q,\varepsilon,F)=\{(q,T,4)\}$
5. $\delta(q,\varepsilon,(E))=\{(q,F,5)\}$

6. $\delta(q,\varepsilon,a)=\{(q,F,6)\}$

7. $\delta(q,a,\varepsilon)=\{(q,a,\varepsilon)\}\ \forall a\in\Sigma$

8. $\delta(q,\varepsilon,\$S)=\{(q,\varepsilon,\varepsilon)\}$

Pentru secvența $w=a+a$ se obține traducerea 146246 astfel:

$(q,a+a,\$,\varepsilon)\vdash(q,+a,\$a,\varepsilon)\vdash(q,+a,\$F,6)\vdash(q,+a,\$T,64)\vdash(q,+a,\$E,642)$
$\vdash(q,a,\$E+,642)\vdash(q,\varepsilon,\$E+a,642)\vdash(q,\varepsilon,\$E+F,6426)\vdash(q,\varepsilon,\$E+T,64264)\vdash(q,\varepsilon,\$E,642641)$
$\vdash(q,\varepsilon,\varepsilon,642641)$

$\tilde{\pi}=146246$ - analiza sintactică la dreapta

Algoritmul de analiză sintactică ascendentă cu reveniri

Intrare: G o gramatică independentă de context, fără cicluri, fără ε-producții și un cuvânt $w=a_1a_2...a_n$, $n\geq 0$. Presupunem că producțiile sunt numerotate de la 1 până la p.

Ieşire: analiza sintactică la dreapta a lui *w* dacă aceasta există, sau "eroare" în caz contrar.

Metoda: Algoritmul utilizează configurații de forma (s,i,α,β) unde:

- s este starea algoritmului şi $s\in\{q,b,t\}$, q-mers normal, b-mers înapoi, t-terminare
- $i\in\{1,2,...,n+1\}$ este poziția curentă pe banda de intrare. Al n+1-lea simbol de intrare este \$ folosit pe post de delimitator la dreapta
- $\alpha\in(N\cup\Sigma)^*\$$ reprezintă conținutul benzii pushdown L_1, bandă cu vârful spre dreapta, şi reprezintă porțiunea în care a fost redusă secvența de intrare
- $\beta\in(\{1,2,...,p,s\})^*$ reprezintă conținutul unei benzi pushdown L_2, cu vârful spre stânga. Această bandă ține evidența reducerilor efectuate pentru a se obține conținutul lui L_1.

$(q,1,\$,\varepsilon)$ - configurația inițială

Trecerea de la o configurație la alta se realizează conform următorilor paşi.

(*P1*): încercare de reducere

$$(q,i,\alpha\beta,\gamma) \vdash (q,i,\alpha A, j\gamma)$$

unde $(A \to \beta) \in P$ este producția cu numărul *j*. După aplicare se rămâne la acest pas dacă este posibil; în caz contrar se merge la pasul 2.

(*P2*) trecere

$$(q,i,\alpha,\gamma) \vdash (q,i+1,\alpha a, s\gamma)$$

Se efectuează când $i \neq n+1$ şi constă în mutarea simbolului terminal din secvență în vârful benzii L_1, şi simbolul s se trece pe banda L_2

(*P3*) obținerea unei configurații de acceptare

$$(q,n+1,\$S,\gamma) \vdash (t,n+1,\$S,\gamma)$$

Se aplică homomorfismul *h* lui *α*:

$h(s)=\varepsilon$, $h(j)=j$, $j \in \{1,2,...,p\}$. Dacă P3 nu s-a executat se merge la P4.

(*P4)* Începutul mersului înapoi:

$$(q,n+1,\alpha,\gamma) \vdash (b,n+1,\alpha,\gamma)$$

cu condiția $\alpha \neq \$S$. Se merge la P5.

(*P5*): mersul înapoi

a) $$(b,i,\alpha A, j\gamma) \vdash (q,i,\alpha' B, k\gamma)$$

unde $(A \to \beta) \in P$ este producția cu numărul *j*.; următoarea producție, In ordinea stabilită la început, al cărei membru drept este un sufix al lui *αβ* este $(B \to \beta') \in P$ şi are numărul k; în acest caz $\alpha\beta = \alpha'\beta'$. După efectuarea acestui pas se merge la P1.

b) $$(b,n+1,\alpha A, j\gamma) \vdash (b,n+1,\alpha\beta,\gamma)$$

unde $(A \to \beta) \in P$ este producția cu numărul j şi nu există nicio altă alternativă de reducere pentru *αβ*. Apoi se merge la P5.

$$\text{c)} \quad (b, i, \alpha A, j\gamma) \vdash (q, i+1, \alpha\beta a_i, s\gamma)$$

cu $i \neq n+1$, $(A \to \beta) \in P$ este producția cu numărul *j* şi nu există nicio altă alternativă de reducere pentru *αβ*. Apoi se merge la P1.

$$\text{d)} \quad (b, i, \alpha a, s\gamma) \vdash (b, i-1, \alpha, \gamma)$$

Se renunță la trecerea lui *a* pe banda L_1;se rămâne la P5.

Aplicație. Fie gramatica expresiilor aritmetice simple:

1. $E \to E + T$
2. $E \to T$
3. $T \to T * F$
4. $T \to F$
5. $F \to (E)$
6. $F \to a$

Să se determine analiza sintactică la dreapta pentru secvența $w = a * a$

$(q,1,\$,\varepsilon) \vdash (q,2,\$a,s) \vdash (q,2,\$F,6s) \vdash (q,2,\$T,46s) \vdash (q,2,\$E,246s) \vdash (q,3,\$E*,s246s) \vdash (q,4,\$E*a,ss246s)$
$\vdash (q,4,\$E*F,46ss246s) \vdash (q,4,\$E*E,246ss246s) \vdash (b,4,\$E*E,246ss246s) \vdash (b,4,\$E*T,46ss246s) \vdash$
$\vdash (b,4,\$E*F,6ss246s) \vdash (b,3,\$E*,s246s) \vdash (b,2,\$E,246s) \vdash (q,3,\$T*,s46s) \vdash (q,4,\$T*a,ss46s) \vdash$
$\vdash (q,4,\$T*F,6ss46s) \vdash (q,4,\$T,36ss46s) \vdash (q,4,\$E,236ss46s) \vdash (t,4,\$E,236ss46s)$

$h(\gamma) = h$ (236ss46s)=23646 – analiza sintactică la dreapta

Algoritmul Cocke-Younger Kasami

Acest algoritm face trecerea de la analiza globală a lui $w = a_1 a_2 ... a_n$ la analize locale. Fiecărei pereche (i, j) ce corespunde porțiunii $a_i ... a_{i+j-1}$ îi ataşăm o mulțime de simboluri neterminale $T(i, j)$ astfel încât $A \in T(i, j)$ dacă şi numai dacă $A \overset{*}{\Rightarrow} a_i ... a_{i+j-1}$. Dacă $S \in T(1, n)$ *atunci* $w \in L(G)$. Se construieşte tabelul T cu regulile:

$T[i,1] = \{A \mid A \in N, (A \to a_i) \in P\}, i \in \{1, ..., n\}$

$T[i,j]=\{A \mid A\in N,(A\to BC)\in P, B\in T[i,k], C\in T[i+k,j-k], k\in\{1,...,j-1\}\}, 2\le j\le n-i+1$

După ce a fost construit tabelul, analiza sintactică se obține cu următorul algoritm:

Intrare: G în formă normală Chomsky, cu producțiile numerotate 1,2,...,p , tabelul T şi $w=a_1a_2...a_n$.

Ieşire: analiza sintacțică la stânga a lui w sau „eroare”.

Metoda: Se foloseşte rutina gen(i,j,A) pentru a genera un arbore de derivare corespunzător derivației $A\overset{*}{\Rightarrow}a_i...a_{i+j-1}$. Rutina este definită astfel:

1) Dacă j=1 şi a m-a producție din P este $A\to a_i$ atunci se emite numărul m.
2) Dacă j>1 şi k este cel mai mic indice, $1\le k<j$, astfel încât pentru $B\in T[i,k], C\in T[i+k,j-k]$, $(A\to BC)\in P$ este producția cu numărul m, atunci se emite m şi se execută gen(i,k,B) şi apoi gen(i+k,j-k,C).

   Dacă $S\in T(1,n)$ se execută gen(1,n,S) iar dacă $S\notin T(1,n)$ se emite mesajul de eroare.

Aplicație.

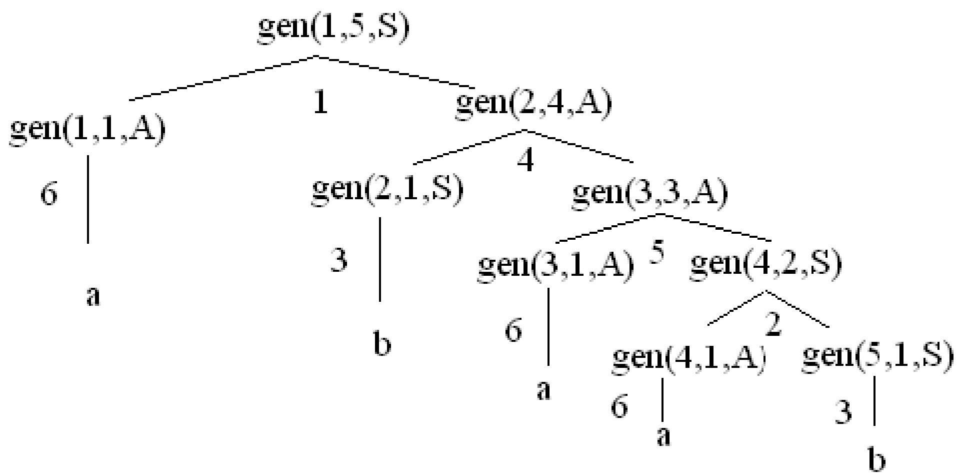
Se consideră gramatica cu producțiile :

1.$S\to AA$
2.$S\to AS$
3.$S\to b$
4.$A\to SA$
5.$A\to AS$
6.$A\to a$

Pentru secvența $w=abaab$ se obține următorul tabel:

	1 a	2 b	3 a	4 a	5 b
1	A	S	A	A	S
2	A,S	A	S	A,S	
3	A,S	S	A,S		

4	A,S	A,S	
5	A,S		



Analiza sintactică la stânga este 164356263.

Teme propuse:

1. Implementați algoritmul de analiză sintactică ascendentă care utilizează un traducător pushdown.
2. Implementați algoritmul de analiză sintactică ascendentă cu reveniri.
3. Implementați algoritmul de analiză sintactică de tip Kasami.