

Laborator 4

Implementare Automat Push Down pt. gramatici LL(k)

1. Notiuni teoretice:

Numele gramaticii LR(k) provine de la prescurtarea:

Left - semnifică citirea sirului de analizat de la stânga la dreapta (left to right)

Right - se urmărește derivarea extremă dreapta

k - numărul de caractere citite deodată pentru a decide ce reducere facem

Definiție: O gramatică de tipul 2 este din clasa LR(k) dacă din derivările:

$$S \Rightarrow u x p_1 p_2 \rightarrow u \alpha p_1 p_2 \quad \text{cu} \quad x, x' \in V_N \\ \alpha, u, u' \in V_T \cup V_N$$

$$S \xrightarrow{*} u' x' p' \rightarrow u \alpha p_1 p_2 \quad p_1, p_2, p_2' \in V_T$$

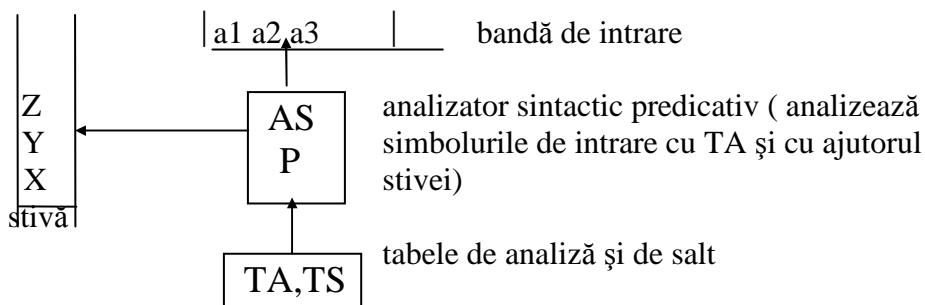
și $|p_1|=k$ lungimea lui p_1 este k (p₁ se numește sir de anticipare) rezultă:
 $x = x'; u = u'; p = p_1 p_2;$

Cu alte cuvinte dacă două forme propoziționale coincid k simboluri după conductorul $u\alpha$, atunci și formele propoziționale precedente coincid până la k simboluri după ultimul neterminal, deci derivarea $x \rightarrow \alpha$ este unică ca și istoria derivărilor anterioare $u=u'$. Conductorul $u\alpha$ include și caractere dinaintea simbolurilor în care se va deriva x.

Pentru recunoașterea părții drepte α a unei reduceri $x \rightarrow \alpha$ este necesară și o parte din contextul reducerii: istoria u și k caractere după α în sirul de intrare=p₁.

O definiție echivalentă ar fi: o gramatică este de tip LR(k) dacă pentru localizarea conductorului trebuie să parcurgem cel mult k simboluri după conductor.

Schema generală a analizatorului bazat pe gramatici LR(K) este un automat push down APD:



Exemplu: Gramatica LR(1) pentru expresii aritmetice

Expresiile aritmetice simple pot fi generate și verificate pe baza următoarei gramatici de tip LR(1)
 $G=(N,T,S,P)$ cu:

$N=\{E,T,F\}$ E semnifica expresie, T termen F factor

$T=\{a,+,-,*,/,(,)\}$ a este identificator de variabile sau constantă numerică

$S=\{E\}$ E este simbol initial

$P=\{1. E \rightarrow E+T,$

2. $E \rightarrow T,$

3. $T \rightarrow T*F,$

4. $T \rightarrow F,$

5. $F \rightarrow (E),$

6. $F \rightarrow id,$

51. $F \rightarrow -(E) \}$

un exemplu de expresie generata de aceata gramatica este : $a+a^*a$

Analiza sintactică este dirijată de două tabele: *tabela de acțiuni* și *tabela de salt*.

În tabela de acțiuni sunt specificate acțiunile pe care urmează să le execute APD extins (se numește *extins* pentru că are și tabela de salt), în funcție de starea în care se află și următorul simbol din banda de intrare. Acestea sunt în genere la fel cu cele de la gramatici de precedență: d_i =deplasare în banda de intrare și salt la starea i ; r_i =reducere cu producția i ;

În tabela de salt este specificată următoarea stare a automatului în funcție de starea automatului și neterminalul la care se face o reducere.

Pentru gramatica de mai sus sunt valabile urmatoarele tabele:

Nr stare	Tabel de acțiuni : TA						Tabel de salt : TS		
	id	+	*	()	\$	E	T	F
0	d_5			d_4			1	2	3
1		d_6				a_{cc}			
2		r_2	d_7		r_2	r_2			
3		r_4	r_4		r_4	r_4			
4	d_5			d_4			8	2	3
5		r_6	r_6		r_6	r_6			
6	d_5			d_4				9	3
7	d_5			d_4					10
8		d_6			d_{11}				
9		r_1	d_7		r_1	r_1			
10		r_3	r_3		r_3	r_3			
11		r_5	r_5		r_5	r_5			

Fig.3.2.3 Tabelele de acțiuni și de salt pentru gramatica G

Stările de pe prima coloană se află pe stivă iar terminalele de pe prima linie din TA reprezintă următorul caracter pe banda de intrare (p_1 din definiție). Spațiul gol din matrice reprezintă eroare. Detailând funcționarea APD care recunoaște gramatica LR(1) prin compararea vârfului stivei cu banda de intrare rezultă următoarele acțiuni:

- d_i = deplasare peste un terminal în sirul de intrare, depunere în stivă a terminalului din sirul de intrare și a stării i din tabelul de acțiuni, salt la starea i ;

d_6 : deplasare și salt la 6;

- r_i = reducere cu producția i , se scot $n*2$ simboluri din stivă (n =lungime membru drept al producției i) și se înlocuiesc cu membrul stâng al producției i (= un neterminal) și apoi se face o comparare a vârfului anterior al stivei(cel dinaintea simbolurilor scoase) și a neterminalului stâng(depus) și se depune în stivă starea rezultată din tabela de salt TS.

Se scot $n*2$ simboluri din stivă pentru că pe lângă simbolurile din membrul drept al producției = n simboluri în stivă se mai află și stări depuse în urma acțiunilor de deplasare și reducere.

Stiva se inițializează cu \$0 (0 este prima stare a automatului) iar cuvântul de intrare se termină cu \$. Evoluția automatului pentru parcurgerea cuvântului de intrare **id+id*id** este prezentată în continuare. Caracterele subliniate din stivă reprezintă simbolurile șterse la o reducere.

Evolutia automatului pentru cuvântul de intrare $\text{id}+\text{id}*\text{id}$ este:

	Stivă	cuvânt de intrare	acțiune rezultată
1	\$ 0	$\text{id}+\text{id}*\text{id}$ \$	d_5
2	\$ 0 <u>id5</u>	$+\text{id}*\text{id}$ \$	$r_6 \Rightarrow F+TS(0,F)$
3	\$ 0 <u>F3</u>	$+\text{id}*\text{id}$ \$	$r_4 \Rightarrow T+TS(0,T)$
4	\$ 0 <u>T2</u>	$+\text{id}*\text{id}$ \$	$r_2 \Rightarrow E+TS(0,E)$
5	\$ 0 E1	$+\text{id}*\text{id}$ \$	d_6
6	\$ 0 E1+6	$\text{id}*\text{id}$ \$	d_5
7	\$ 0 E1+6 <u>id5</u>	$*\text{id}$ \$	$r_6 \Rightarrow F+TS(6,F)$
8	\$ 0 E1+6 <u>F3</u>	$*\text{id}$ \$	$r_4 \Rightarrow T+TS(6,T)$
9	\$ 0 E1+6 <u>T9</u>	$*\text{id}$ \$	d_7
10	\$ 0 E1+6 <u>T9*7</u>	id \$	d_5
11	\$ 0 E1+6 <u>T9*7 id5</u>	\$	$r_6 \Rightarrow F+TS(7,F)$
12	\$ 0 E1+6 <u>T9*7 F10</u>	\$	$r_3 \Rightarrow T+TS(6,T)$
13	\$ 0 <u>E1+6 T9</u>	\$	$r_1 \Rightarrow E+TS(0,E)$
14	\$ 0 E 1	\$	acceptare

2. Cerinte:

Se va implementa printr-un program Automatul Push Down pentru gramatici LR(1)
El va citi un sir de intrare și va decide dacă este corect(generat de gramatică) sau eronat.

2.1 Indicatii de programare:

- Tabele de analiză și salt sunt date de intrare, trebuie stabilite structuri de date pentru memorarea lor
- Stiva se va implementa să poată memora perechi de simbol-stare (șir de caractere-numeric). Se vor implementa actiuni de ‘pop’ și ‘push’ pentru perechi simbol stare
- se vor folosi structurile de date din laboratorul 3 (generare de siruri pornind de la o gramatica) pentru memorarea productiilor gramaticii
- Programul trebuie să implementeze actiunile automatului push-down: deplasare, reducere