

Limbaje formale, automate și compilatoare

Curs 12

Gramatici LR(1)

► Definiție

- Fie $G = (V, T, S, P)$ o gramatică redusă. Un articol LR(1) pentru gramatica G este o pereche $(A \rightarrow \alpha \bullet \beta, a)$, unde $A \rightarrow \alpha \beta$ este un articol LR(0), iar $a \in \text{FOLLOW}(A)$ ($\#$ în loc de ε).

► Definiție

- Articolul $(A \rightarrow \beta_1 \bullet \beta_2, a)$ este valid pentru prefixul viabil $\alpha \beta_1$ dacă are loc derivarea
 - $S \xRightarrow{dr} \alpha A u \Rightarrow \alpha \beta_1 \beta_2 u$
 - iar $a = 1 : u$ ($a = \#$ dacă $u = \varepsilon$).

► Teorema

- O gramatică $G = (V, T, S, P)$ este gramatică LR(1) dacă și numai dacă oricare ar fi prefixul viabil φ , nu există două articole distincte, valide pentru φ , de forma $(A \rightarrow \alpha \bullet, a)$, $(B \rightarrow \beta \bullet \gamma, b)$ unde $a \in \text{FIRST}(\gamma b)$.

Gramatici LR(1)

- ▶ Nu există conflict deplasare/reducere. Un astfel de conflict înseamnă două articole $(A \rightarrow \alpha \bullet, a)$ și $(B \rightarrow \beta \bullet a \beta', b)$ valide pentru același prefix.
- ▶ Nu există conflict reducere/reducere. Un astfel de conflict înseamnă două articole complete $(A \rightarrow \alpha \bullet, a)$ și $(B \rightarrow \beta \bullet, a)$ valide pentru același prefix
- ▶ Pentru a verifica dacă o gramatică este LR(1) se construiește automatul LR(1) în mod asemănător ca la LR(0):
 - Automatul are ca stări mulțimi de articole LR(1)
 - Tranzițiile se fac cu simboluri ce apar după punct
 - Închiderea unei mulțimi de articole se bazează pe faptul că dacă articolul $(B \rightarrow \beta \bullet A \beta', b)$ este valid pentru un prefix viabil ϕ atunci toate articolele de forma $(A \rightarrow \bullet \alpha, a)$ unde $a \in \text{FIRST}(\alpha)$ sunt valide pentru același prefix.

Procedura de închidere LR(1)

```
▶ flag= true;
▶ while( flag) {
  ◦ flag= false;
  ◦ for ( (A→  $\alpha \bullet B \beta$ , a) ∈ I) {
    • for B →  $\gamma$  ∈ P)
      • for( b ∈ FIRST( $\beta a$ )) {
        • if( (B →  $\bullet \gamma$  , b) ∉ I) {
          ▪ I = I ∪ {(B →  $\bullet \gamma$  , b)};
          ▪ flag = true;
        • }//endif
      • }//endforb
    • }//endforB
  ◦ }//endforA
▶ }//endwhile
▶ return I;
```

Automatul LR(1)

```
▶ t0 = închidere((S' → • S, #)); T = {t0}; marcat(t0) = false;
▶ while(∃ t ∈ T && !marcat(t)) { // marcat(t) = false
  ◦ for( X ∈ Σ ) {
    ◦ t' = Φ;
    • for( (A → α • Xβ , a) ∈ t )
      • t' = t' ∪ { (B → αX • β , a) | (B → α • Xβ , a) ∈ t };
      • if( t' ≠ Φ ) {
        • t' = închidere( t' );
        • if( t' ∈ T ) {
          ▪ T = T ∪ { t' };
          ▪ marcat( t' ) = false;
        } // endif
        • g(t, X) = t';
      • } // endif
    ◦ } // endfor
    ◦ marcat( t ) = true;
  ▶ } // endwhile
```

Automatul LR(1)

► Teorema

- Automatul M construit în algoritmul 2 este determinist și $L(M)$ coincide cu mulțimea prefixelor viabile ale lui G . Mai mult, pentru orice prefix viabil γ , $g(t_0, \gamma)$ reprezintă mulțimea articolelor LR(1) valide pentru γ .
- Automatul LR(1) poate fi folosit pentru a verifica dacă G este LR(1)
 - Conflict reducere/reducere: Dacă în T există o stare ce conține articole de forma $(A \rightarrow \alpha\bullet, a)$, $(B \rightarrow \beta\bullet, a)$ atunci gramatica nu este LR(1);
 - Conflict deplasare/reducere: Dacă în T există o stare ce conține articole de forma $(A \rightarrow \alpha\bullet, a)$ și $(B \rightarrow \beta_1\bullet a\beta_2, b)$, atunci G nu este LR(1).
 - O gramatică este LR(1) dacă orice stare $t \in T$ este liberă de conflicte

Exemplu

► $S \rightarrow L=R|R, L \rightarrow *R|a, R \rightarrow L$

0
 $(S' \rightarrow \bullet S, \#)$
 $(S \rightarrow \bullet L=R, \#)$
 $(S \rightarrow \bullet R, \#)$
 $(L \rightarrow \bullet *R, \{=, \#\})$
 $(L \rightarrow \bullet a, \{=, \#\})$
 $(R \rightarrow \bullet L, \#)$

6
 $(S \rightarrow L=\bullet R, \#)$
 $(R \rightarrow \bullet L, \#)$
 $(L \rightarrow \bullet *R, \#)$
 $(L \rightarrow \bullet a, \#)$

1
 $(S' \rightarrow S\bullet, \#)$

2
 $(S \rightarrow L\bullet=R, \#)$
 $(R \rightarrow L\bullet, \#)$

7
 $(L \rightarrow *R\bullet, \{=, \#\})$

8
 $(R \rightarrow L\bullet, \{=, \#\})$

12
 $(L \rightarrow a\bullet, \#)$

3
 $(S \rightarrow R\bullet, \#)$

5
 $(L \rightarrow a\bullet, \{=, \#\})$

9
 $(S \rightarrow L=R\bullet, \#)$

10
 $(R \rightarrow L\bullet, \#)$

13
 $(L \rightarrow *R\bullet, \#)$

4
 $(L \rightarrow *\bullet R, \{=, \#\})$
 $(R \rightarrow \bullet L, \{=, \#\})$
 $(L \rightarrow \bullet *R, \{=, \#\})$
 $(L \rightarrow \bullet a, \{=, \#\})$

11
 $(L \rightarrow *\bullet R, \#)$
 $(R \rightarrow \bullet L, \#)$
 $(L \rightarrow \bullet *R, \#)$
 $(L \rightarrow \bullet a, \#)$

Tabela de tranziție

g	a	=	*	S	L	R
0	5		4	1	2	3
1						
2		6				
3						
4	5		4		8	7
5						
6	12		11		10	9
7						
8						
9						
10						
11	12		11		10	13
12						
13						

Construcția tablei de analiză LR(1)

- ▶ `for` ($t \in T$)
 - `for` ($a \in T$) `ACTIUNE` (t, a) = "eroare";
 - `for` ($A \in V$) `GOTO` (t, A) = "eroare";
- ▶ `for` ($t \in T$) {
 - `for` ($(A \rightarrow \alpha \bullet a \beta, L) \in t$)
 - `ACTIUNE` (t, a) = "D $g(t, a)$ "; //deplasare in $g(t, a)$
 - `for` ($(B \rightarrow \gamma \bullet, L) \in t$) { // acceptare sau reducere
 - `for` ($c \in L$) {
 - `if` ($B == 'S'$) `ACTIUNE` (t, a) = "acceptare";
 - `else` `ACTIUNE` (t, c) = "R $B \rightarrow \gamma$ "; //reducere cu $B \rightarrow \gamma$
 - } //endfor
 - } // endfor
 - `for` ($A \in N$) `GOTO` (t, A) = $g(t, A)$;
- ▶ } //endfor

Exemplu

- 0: $S' \rightarrow S$, 1: $S \rightarrow L=R$, 2: $S \rightarrow R$, 3: $L \rightarrow *R$, 4: $L \rightarrow a$, 5: $R \rightarrow L$

STARE	ACȚIUNE				GOTO		
	a	=	*	#	S	L	R
0	D5		D4		1	2	3
1				acceptare			
2		D6		R5			
3				R2			
4	D5		D4			8	7
5		R4		R4			
6	D12		D11			10	9
7		R3		R3			
8		R5		R5			
9				R1			
10				R5			
11	D12		D11			10	13
12				R4			
13				R3			

Exemplu

- ▶ Fie cuvintele
 - $***a$
 - $a = **a$
 - $*a = **a$
- ▶ Analiza LR(1)?

Gramatici LALR(1)

► Definiție

- Fie t o stare în automatul LR(1) pentru G . Nucleul acestei stări este mulțimea articolelor LR(0) care apar ca prime componente în articolele LR(1) din t .

► Definiție

- Două stări t_1 și t_2 ale automatului LR(1) pentru gramatica G sunt echivalente dacă au același nucleu.

Gramatici LALR(1)

- ▶ Fiecare stare a automatului LR(1) este o mulțime de articole LR(1). Pornind de la două stări t_1 și t_2 putem vorbi de starea $t_1 \cup t_2$.
 - Fie $t_1 = \{(L \rightarrow *R., \{=, \# \})\}$, $t_2 = \{(L \rightarrow *R., \#)\}$, atunci $t_1 \cup t_2 = t_1$ pentru că $t_2 \subset t_1$.
- ▶ **Definiție**
 - Fie G gramatică LR(1) și $M = (Q, \Sigma, g, t_0, Q)$ automatul LR(1) corespunzător. Spunem că gramatica G este LALR(1) (Look Ahead LR(1)) dacă oricare ar fi perechea de stări echivalente t_1, t_2 din automatul LR(1), starea $t_1 \cup t_2$ este liberă de conflicte.

Tabela de analiză LALR(1)

- ▶ **Intrare:** Gramatica $G = (N, T, S, P)$ augmentată cu $S' \rightarrow S$;
- ▶ **Ieșire:** Tabela de analiză LALR(1) (ACȚIUNE și GOTO).
- ▶ **Algoritm:**
 - **1.** Se construiește automatul LR(1), $M = (Q, \Sigma, g, t_0, Q)$
Fie $Q = \{t_0, t_1, \dots, t_n\}$. Dacă toate stările din Q sunt libere de conflict, urmează 2, altfel algoritmul se oprește deoarece gramatica nu este LR(1).
 - **2.** Se determină stările echivalente din Q și, prin reuniunea acestora, se obține o nouă mulțime de stări $Q' = \{t'_0, t'_1, \dots, t'_m\}$
 - **3.** Dacă în Q' există stări ce conțin conflicte, algoritmul se oprește deoarece gramatica G nu este LALR(1).

Tabela de analiză LALR(1)

- **4.** Se construiește automatul $M' = (Q', \Sigma, g', t'_0, Q')$, unde $\forall t' \in Q'$:
 - **5.** Dacă $t' \in Q$ atunci $g'(t', X) = g(t, X)$, $\forall X \in \Sigma$;
 - **6.** Dacă $t' = t_1 \cup t_2 \cup \dots$, $t_1, t_2, \dots \in Q$, atunci
 - **7.** $g'(t', X) = g(t_1, X) \cup g(t_2, X) \cup \dots$
- **8.** Se aplică algoritmul pentru construirea tabelii de parsare LR(1) pornind de la automatul M' . Tabela obținută se numește tabela LALR(1) pentru gramatica G .

Exemplu

- Pentru gramatica discutată anterior avem $4 \cup 11 = 4$, $5 \cup 12 = 5$, $7 \cup 13 = 7$, $8 \cup 10 = 8$

STAR E	ACȚIUNE				GOTO		
	a	=	*	#	S	L	R
0	D5		D4		1	2	3
1				acceptare			
2		D6		R5			
3				R2			
4	D5		D4			8	7
5		R4		R4			
6	D5		D4			8	9
7		R3		R3			
8		R5		R5			
9				R1			

Exemplu

- ▶ Există gramatici LR(1) care nu sunt LALR(1).
 - $S \rightarrow aAb \mid bAd \mid aBd \mid bBb$
 - $A \rightarrow e$
 - $B \rightarrow e$