

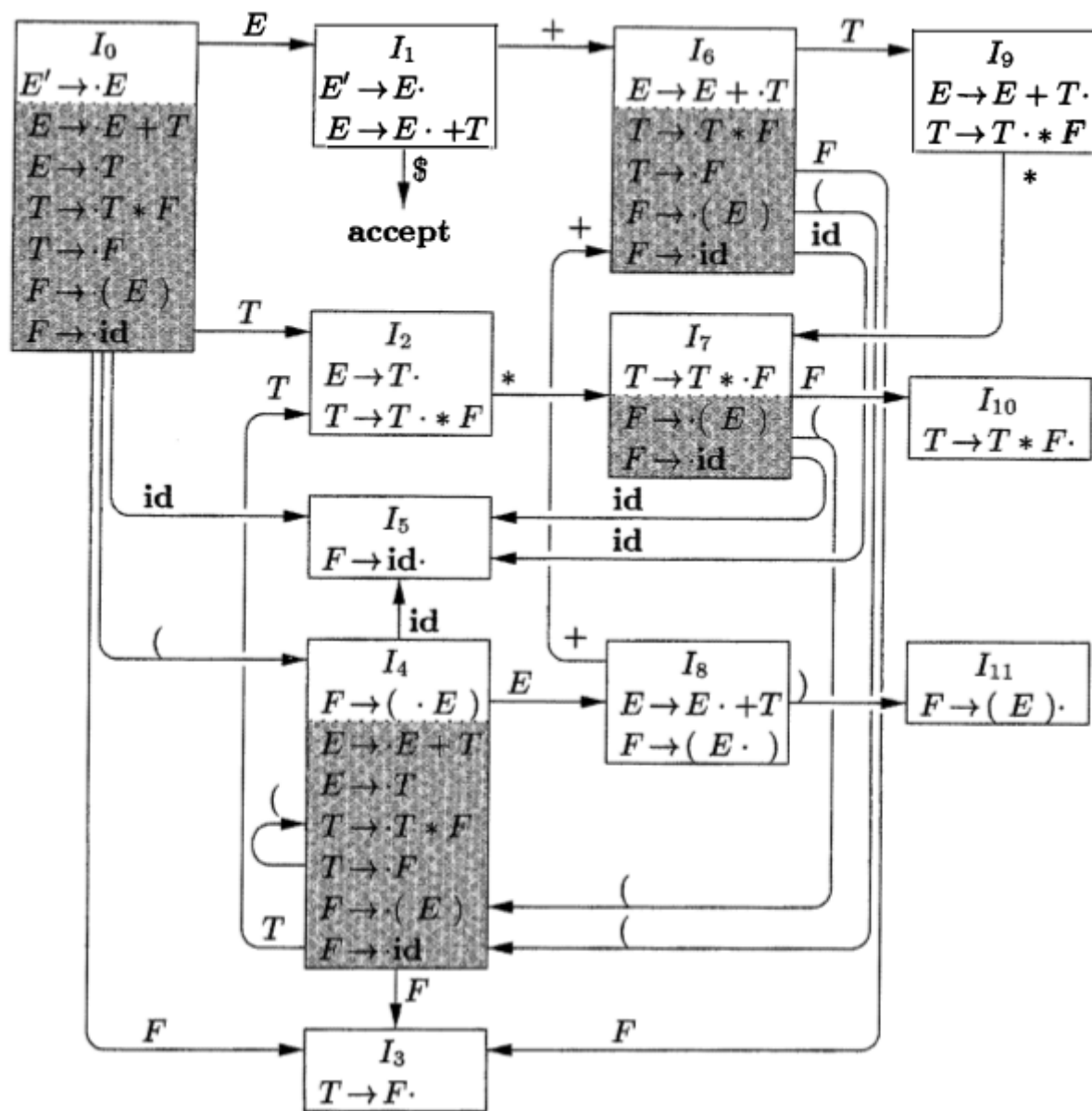
به نام خدا

SLR(1)

مرور مطالب گذشته

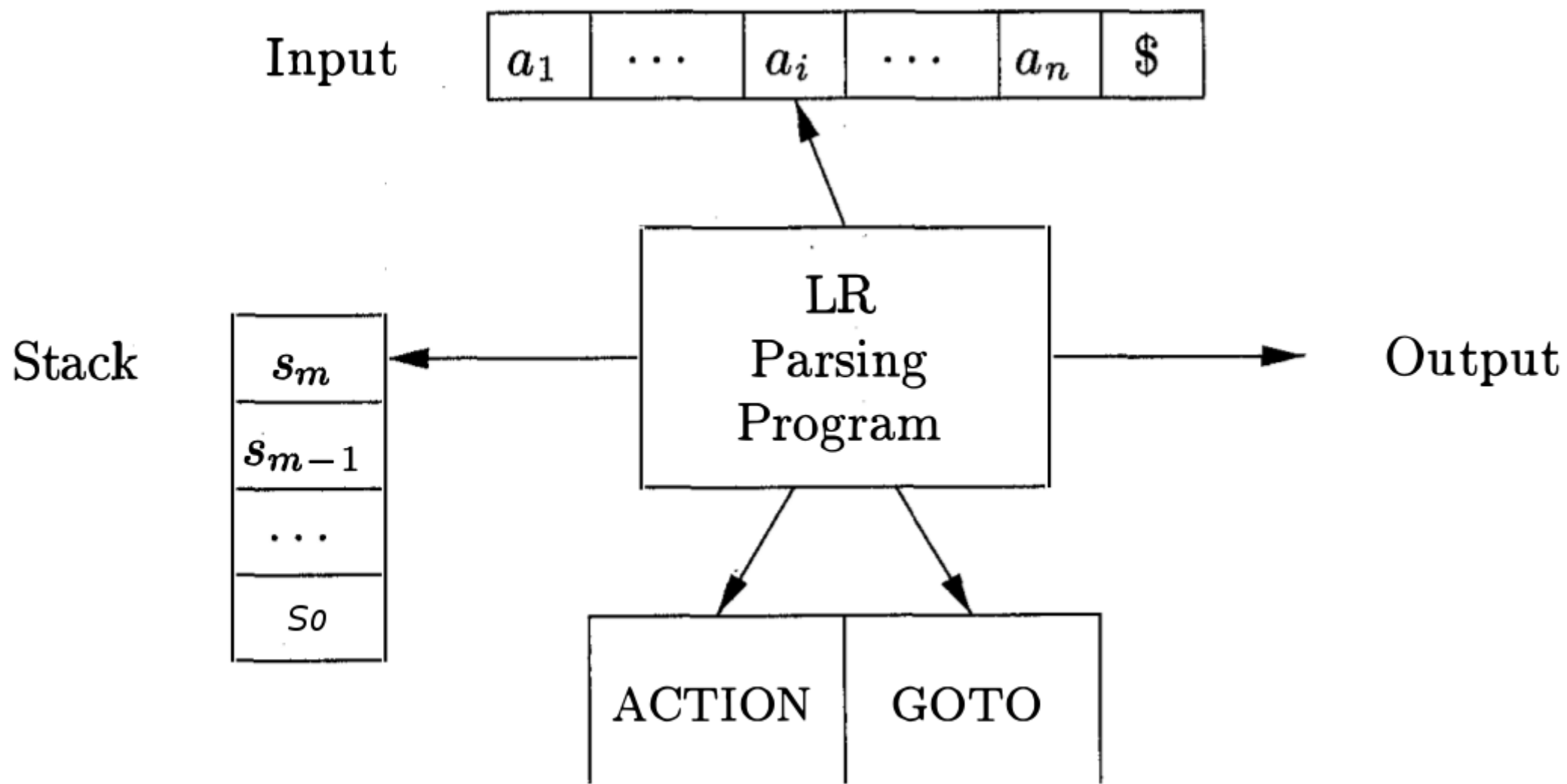
- آیتم های LR(0)
- تابع بستار
- تابع goto
- مجموعه کانونی مجموعه آیتم های LR(0)
 - $C = \text{CLOSURE}(\{[S' \rightarrow \cdot S]\});$
 - برای هر مجموعه آیتم I در C
- برای هر سمبل X (پایانه و غیر پایانه) در گرامر
 - اگر $\text{GOTO}(I, X)$ تهی نیست و در C نیست، آن را به C اضافه کن.

ماشین LR(0) برای گرامر گفته شده

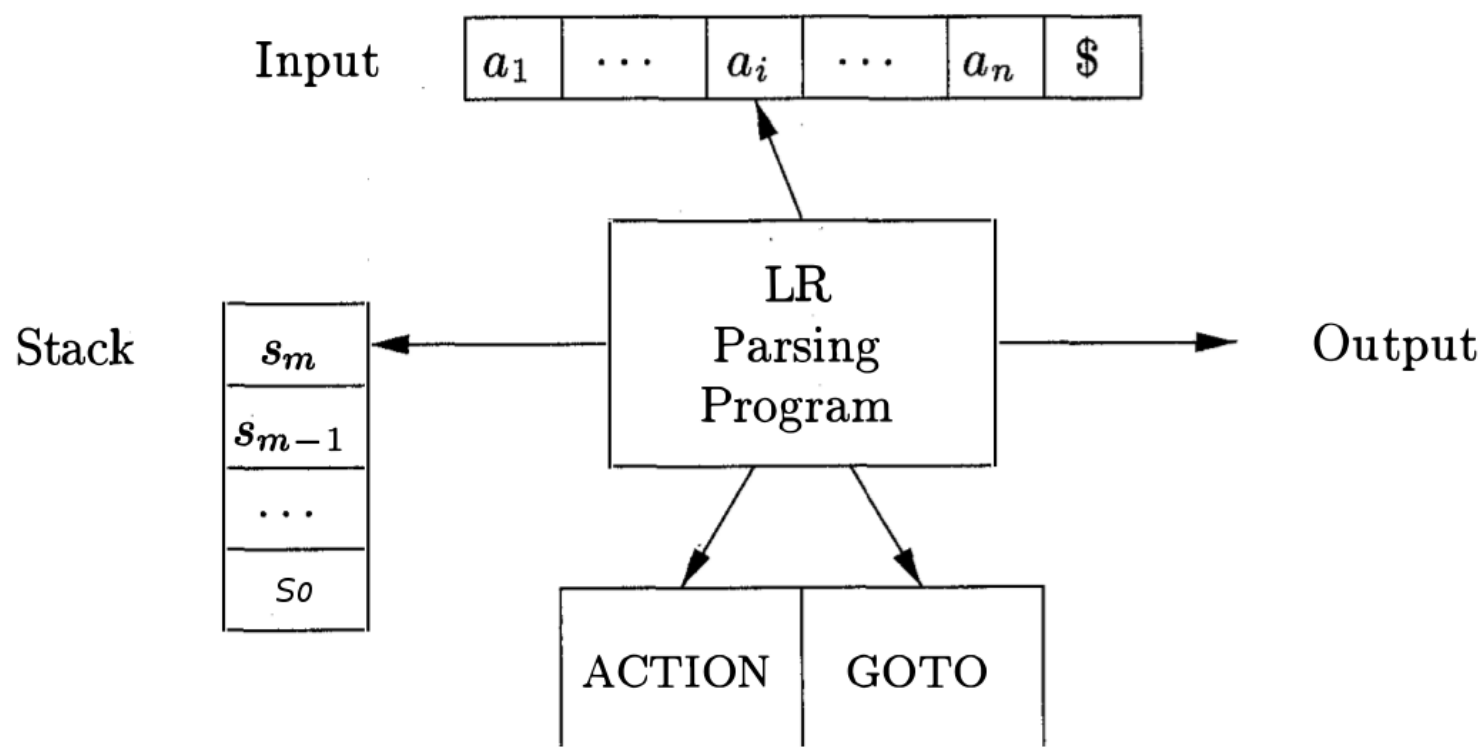


- راس (حالت): یک مجموعه آیت در مجموعه کانونی آیت های LR(0)
- یال: سمبل های گرامر
- در حین ایجاد مجموعه کانونی مجموعه آیت های LR(0)، می توانیم این ماشین را بسازیم.
- هر راس نشان دهنده یک سمبل است.
 - یالی که با آن به آن راس می رسیم.
 - پایانه
 - غیر پایانه
 - به جز راس اول

تجزیه گر پایین به بالا



تجزیه گر پایین به بالا



- ورودی

- پشته

- خروجی

- برنامه تجزیه LR

- جدول تجزیه

Action -

Goto -

- برنامه تجزیه LR برای سه روش تجزیه یکسان است.

- در این سه روش جدول تجزیه فرق می کند.

ساختار جدول تجزیه

STATE	ACTION						GOTO		
	id	+	*	()	\$	<i>E</i>	<i>T</i>	<i>F</i>
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

ساختار جدول تجزیه

• Action[i,a]

- ورودی ها: i : شماره حالت، a : پایانه و یا \$
- می تواند یکی از ۴ مقدار زیر را داشته باشد:
- انتقال j که در آن j شماره حالت است. در این صورت برنامه تجزیه حالت j (که نشان دهنده ورودی است) را به پشته اضافه می کند.
- کاهش $A \rightarrow \beta$. در این صورت برنامه تجزیه β را به A کاهش می دهد.
- پذیرش: برنامه تجزیه ورودی را می پذیرد و تجزیه کاهش می یابد.
- خطا: برنامه تجزیه خطا را کشف می کند.

• Goto[i,A]

- ورودی ها: i : شماره حالت، A : غیرپایانه
- خروجی آن شماره حالت j است اگر $GOTO[I_i, A] = I_j$

وضعیت تجزیه گر LR

- وضعیت تجزیه

$$(s_0 s_1 \cdots s_m, a_i a_{i+1} \cdots a_n \$)$$

- نشان دهنده فرم جمله ای

$$X_1 X_2 \cdots X_m a_i a_{i+1} \cdots a_n$$

- هر یک از S_i ها نشان دهنده یکی از سمبل های گرامر اند.

- به جز S_0 که حالت شروع است.

- با اجرای برنامه تجزیه در هر گام یک عمل معکوس بسط سمت راست ترین را انجام می دهیم.

$$(s_0, a_1 a_2 \cdots a_n \$)$$

حالت پذیرش

$$(s_0 s_a, \$)$$



اعمال تجزیه گر

$$(s_0 s_1 \cdots s_m, a_i a_{i+1} \cdots a_n \$)$$

- در هر گام با توجه به ورودی و حالت بالای پشته عملی انجام می دهیم.
- اگر $\text{ACTION}[s_m, a_i] = \text{shift } s$: باید حالت s به پشته اضافه شود (نشان دهنده ورودی) و ورودی یکی به جلو برود تا به وضعیت زیر برسیم.

$$(s_0 s_1 \cdots s_m s, a_{i+1} \cdots a_n \$)$$

- اگر $\text{ACTION}[s_m, a_i] = \text{reduce } A \rightarrow \beta$: در این حالت باید کاهش انجام شود تا به وضعیت زیر برسیم.

$$(s_0 s_1 \cdots s_{m-r} s, a_i a_{i+1} \cdots a_n \$)$$

r is the length of β , and $s = \text{GOTO}[s_{m-r}, A]$.

- خروجی تجزیه با چاپ قاعده کاهش در این مرحله ایجاد می شود.
- اگر $\text{ACTION}[s_m, a_i] = \text{accept}$: تجزیه با پذیرش رشته پایان می پذیرد.
- اگر $\text{ACTION}[s_m, a_i] = \text{error}$: تجزیه گر خطا را کشف کرده است.

```
let  $a$  be the first symbol of  $w\$$ ;  
while(1) { /* repeat forever */  
    let  $s$  be the state on top of the stack;  
    if ( ACTION[ $s, a$ ] = shift  $t$  ) {  
        push  $t$  onto the stack;  
        let  $a$  be the next input symbol;  
    } else if ( ACTION[ $s, a$ ] = reduce  $A \rightarrow \beta$  ) {  
        pop  $|\beta|$  symbols off the stack;  
        let state  $t$  now be on top of the stack;  
        push GOTO[ $t, A$ ] onto the stack;  
        output the production  $A \rightarrow \beta$ ;  
    } else if ( ACTION[ $s, a$ ] = accept ) break; /* parsing is done */  
    else call error-recovery routine;  
}
```

مثال تجزیه

$$(1) \quad E \rightarrow E + T$$

$$(2) \quad E \rightarrow T$$

$$(3) \quad T \rightarrow T * F$$

$$(4) \quad T \rightarrow F$$

$$(5) \quad F \rightarrow (E)$$

$$(6) \quad F \rightarrow \mathbf{id}$$

1. si means shift and stack state i ,
2. rj means reduce by the production numbered j ,
3. acc means accept,
4. $blank$ means error.

مثال تجزیه

• جدول تجزیه SLR

STATE	ACTION						GOTO		
	id	+	*	()	\$	<i>E</i>	<i>T</i>	<i>F</i>
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

مثال تجزیه

	STACK	SYMBOLS	INPUT	ACTION
(1)	0		id * id + id \$	shift
(2)	0 5	id	* id + id \$	reduce by $F \rightarrow \text{id}$
(3)	0 3	F	* id + id \$	reduce by $T \rightarrow F$
(4)	0 2	T	* id + id \$	shift
(5)	0 2 7	$T *$	id + id \$	shift
(6)	0 2 7 5	$T * \text{id}$	+ id \$	reduce by $F \rightarrow \text{id}$
(7)	0 2 7 10	$T * F$	+ id \$	reduce by $T \rightarrow T * F$
(8)	0 2	T	+ id \$	reduce by $E \rightarrow T$
(9)	0 1	E	+ id \$	shift
(10)	0 1 6	$E +$	id \$	shift
(11)	0 1 6 5	$E + \text{id}$	\$	reduce by $F \rightarrow \text{id}$
(12)	0 1 6 3	$E + F$	\$	reduce by $T \rightarrow F$
(13)	0 1 6 9	$E + T$	\$	reduce by $E \rightarrow E + T$
(14)	0 1	E	\$	accept

مثال تجزیه

- معکوس استخراج سمت راست ترین

STACK	SYMBOLS	INPUT
0		id * id + id
0 5	id	* id + id
0 3	<i>F</i>	* id + id
0 2	<i>T</i>	* id + id
0 2 7	<i>T *</i>	id + id
0 2 7 5	<i>T * id</i>	+ id
0 2 7 10	<i>T * F</i>	+ id
0 2	<i>T</i>	+ id
0 1	<i>E</i>	+ id
0 1 6	<i>E +</i>	id
0 1 6 5	<i>E + id</i>	
0 1 6 3	<i>E + F</i>	
0 1 6 9	<i>E + T</i>	
0 1	<i>E</i>	

ساخت جدول تجزیه SLR

- از آیتم های $LR(0)$ و ماشین حالت $LR(0)$ کمک می گیریم.
- باید برای هر غیرپایانه مانند A ، مجموعه $FOLLOW(A)$ را بدانیم.
- ورودی: گرامر تقویت شده G'
- خروجی: جدول تجزیه شامل بخش های ACTION و GOTO

ساخت جدول تجزیه SLR

(1) مجموعه آیت‌های LR(0) را برای گرامر تقویت شده بسازید. $C = \{I_0, I_1, \dots, I_n\}$

(2) حالت i از روی آیت I_i ساخته می‌شود. بخش ACTION مربوط به حالت i به صورت زیر ساخته می‌شود:

- (a) If $[A \rightarrow \alpha \cdot a \beta]$ is in I_i and $\text{GOTO}(I_i, a) = I_j$, then set $\text{ACTION}[i, a]$ to “shift j .” Here a must be a terminal.
- (b) If $[A \rightarrow \alpha \cdot]$ is in I_i , then set $\text{ACTION}[i, a]$ to “reduce $A \rightarrow \alpha$ ” for all a in $\text{FOLLOW}(A)$; here A may not be S' .
- (c) If $[S' \rightarrow S \cdot]$ is in I_i , then set $\text{ACTION}[i, \$]$ to “accept.”

اگر با توجه به قوانین بالا برای یک وضعیت و یک ورودی دو عمل وجود داشت، می‌گوییم گرامر SLR(1) نیست.

ساخت جدول تجزیه SLR

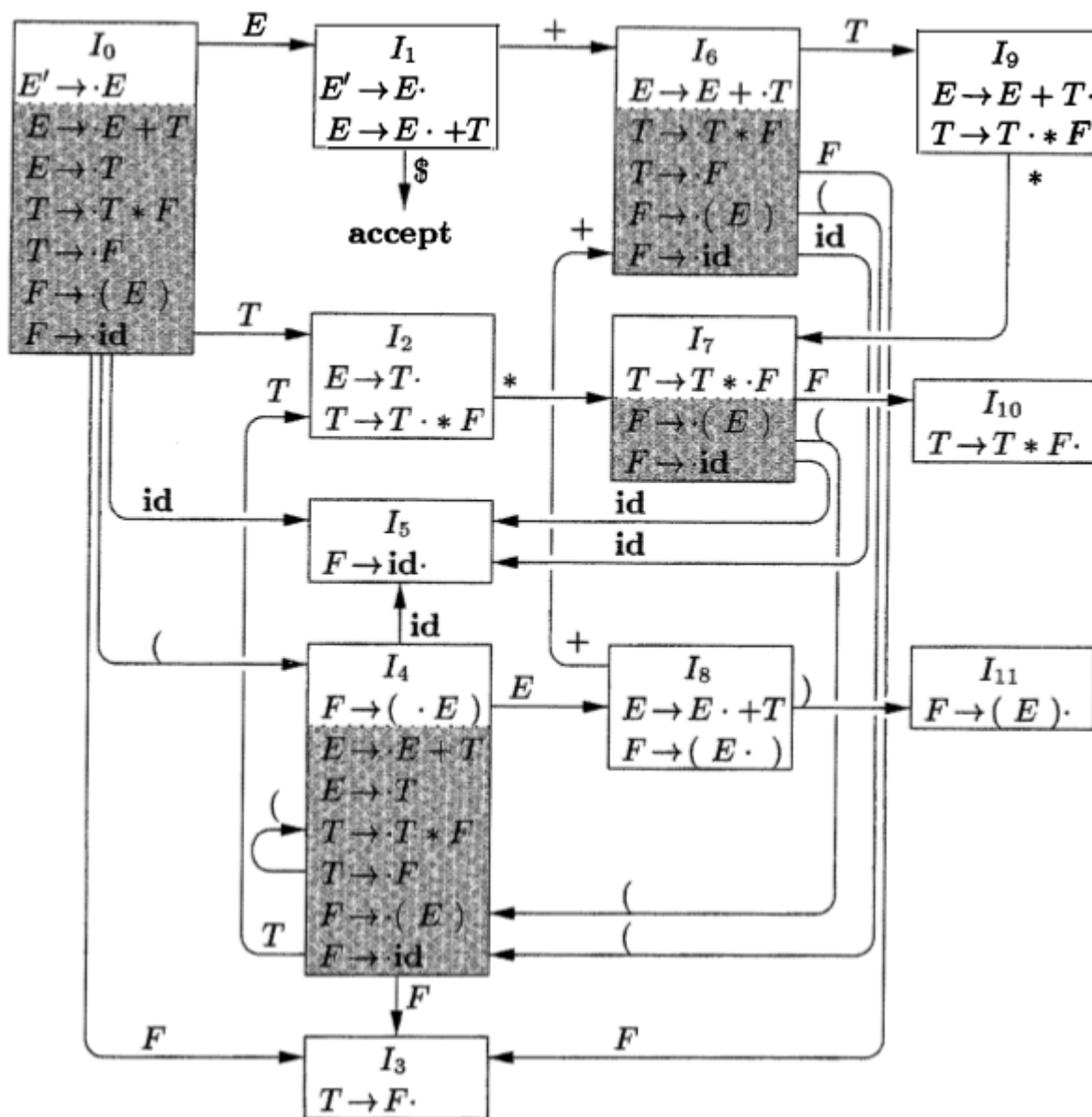
(3) برای بخش GOTO: If $\text{GOTO}(I_i, A) = I_j$, then $\text{GOTO}[i, A] = j$.

(4) خانه هایی که پس از پر کردن جدول با توجه به بخش های ۲ و ۳ خالی مانده اند، خانه های خطا هستند.

(5) حالتش شروع، حالتی است که آیتم زیر را دارد:

$$[S' \rightarrow \cdot S].$$

مثال ساخت جدول تجزیه SLR



مثال ساخت جدول تجزیه SLR

I_0 :

$E' \rightarrow \cdot E$

$E \rightarrow \cdot E + T$

$E \rightarrow \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot (E) \longrightarrow \text{ACTION}[0, (] = \text{shift } 4$

$F \rightarrow \cdot \mathbf{id} \longrightarrow \text{ACTION}[0, \mathbf{id}] = \text{shift } 5.$

I_1 :

$E' \rightarrow E \cdot \longrightarrow \text{ACTION}[1, \$] = \text{accept},$

$E \rightarrow E \cdot + T \longrightarrow \text{ACTION}[1, +] = \text{shift } 6.$

مثال ساخت جدول تجزیه SLR

I_2 :

$E \rightarrow T \cdot$ \longrightarrow $\text{ACTION}[2, \$] = \text{ACTION}[2, +] = \text{ACTION}[2,)] = \text{reduce } E \rightarrow T$

$T \rightarrow T \cdot * F$

$\text{FOLLOW}(E) = \{ \$, +,) \}$

$\text{ACTION}[2, *] = \text{shift } 7$

جدول تجزیه SLR

STATE	ACTION						GOTO		
	id	+	*	()	\$	<i>E</i>	<i>T</i>	<i>F</i>
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

مثال ساخت درخت تجزیه برای گرامری که SLR نیست

$$\begin{array}{lcl} S & \rightarrow & L = R \mid R \\ L & \rightarrow & *R \mid \text{id} \\ R & \rightarrow & L \end{array}$$

ACTION[2, =]

shift 6. reduce $R \rightarrow L$

= $\in \text{FOLLOW}(R)$

$I_0:$ $S' \rightarrow \cdot S$
 $S \rightarrow \cdot L = R$
 $S \rightarrow \cdot R$
 $L \rightarrow \cdot * R$
 $L \rightarrow \cdot \text{id}$
 $R \rightarrow \cdot L$

$I_5:$ $L \rightarrow \text{id} \cdot$

$I_6:$ $S \rightarrow L = \cdot R$
 $R \rightarrow \cdot L$
 $L \rightarrow \cdot * R$
 $L \rightarrow \cdot \text{id}$

$I_1:$ $S' \rightarrow S \cdot$

$I_7:$ $L \rightarrow *R \cdot$

$I_2:$ $S \rightarrow L \cdot = R$
 $R \rightarrow L \cdot$

$I_8:$ $R \rightarrow L \cdot$

$I_3:$ $S \rightarrow R \cdot$

$I_9:$ $S \rightarrow L = R \cdot$

$I_4:$ $L \rightarrow * \cdot R$
 $R \rightarrow \cdot L$
 $L \rightarrow \cdot * R$
 $L \rightarrow \cdot \text{id}$