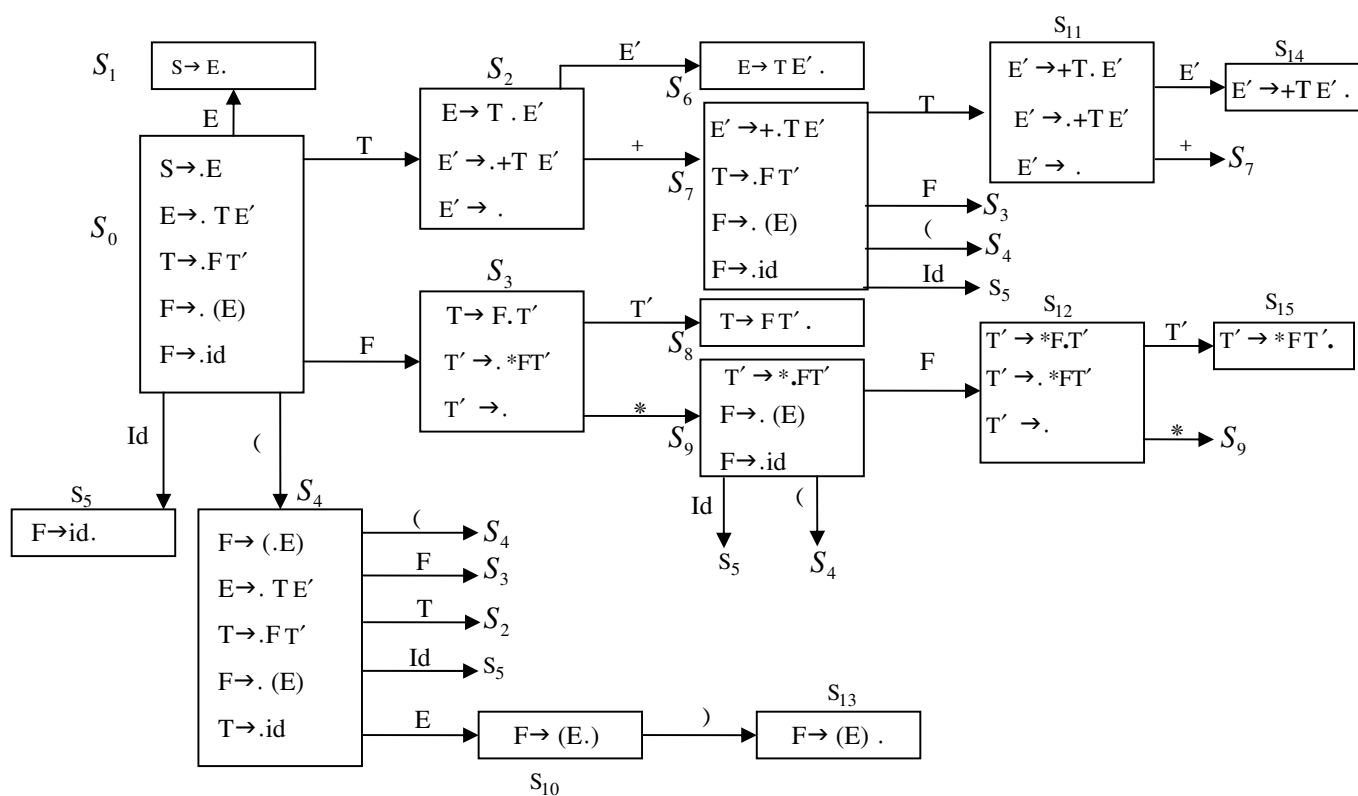


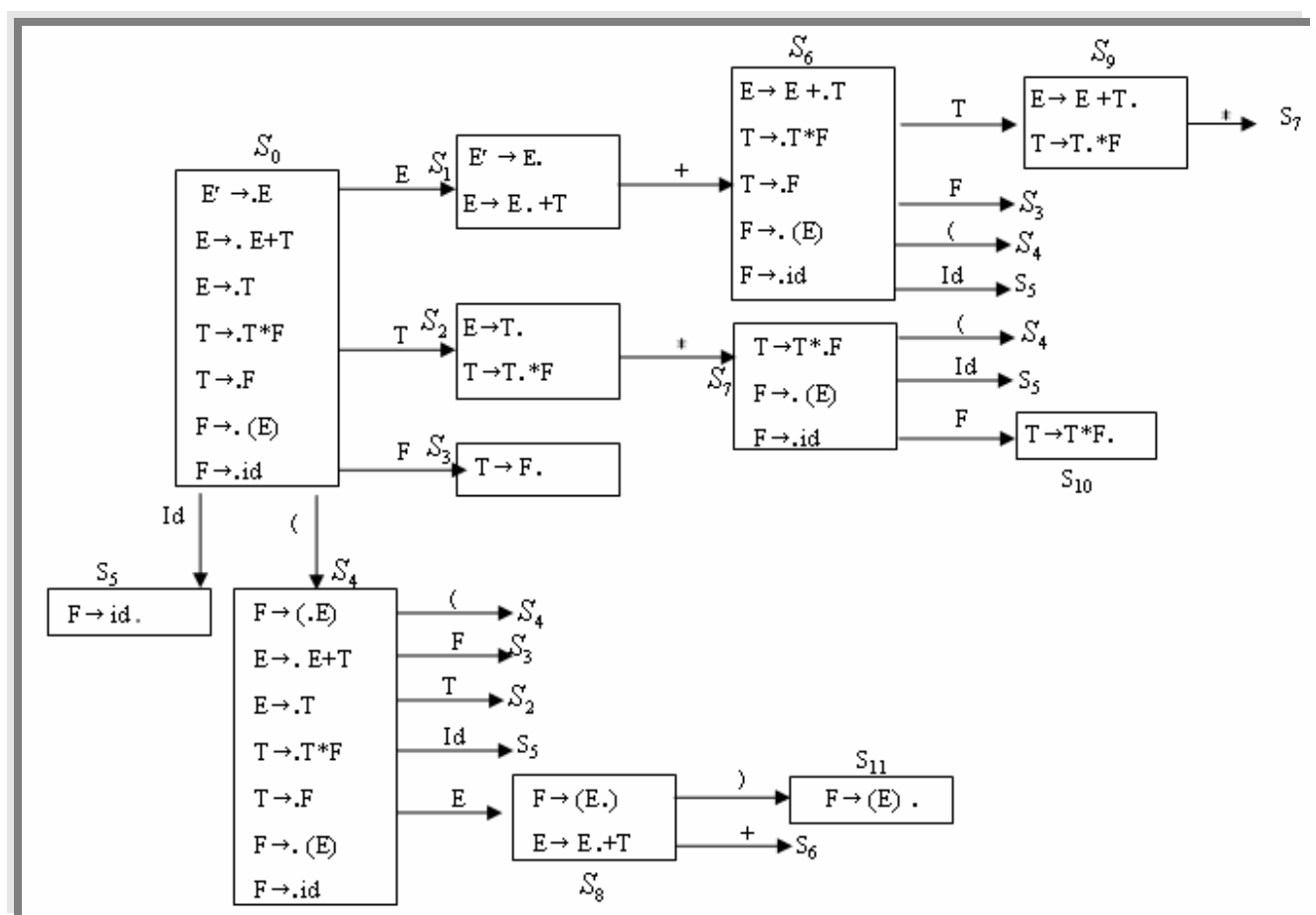
مثال. دیاگرام حالت گرامر زیر را رسم کنید.

- 1) $E \rightarrow TE'$
- 2) $E' \rightarrow +TE'$
- 3) $E' \rightarrow \varepsilon$
- 4) $T \rightarrow FT'$
- 5) $T' \rightarrow *FT'$
- 6) $T' \rightarrow \varepsilon$
- 7) $F \rightarrow (E)$
- 6) $F \rightarrow id$



مثال. دیاگرام حالت گرامر زیر را رسم کنید.

- $E' \rightarrow E$
- 1) $E \rightarrow E + T$
 - 2) $E \rightarrow T$
 - 3) $T \rightarrow T * F$
 - 4) $T \rightarrow F$
 - 5) $F \rightarrow (E)$
 - 6) $F \rightarrow id$



در جدول تجزیه SLR :

- سطرها State ها را نشان می دهند.
- ستون های قسمت اول (action)، ترمینال ها + \$ را نشان می دهند
- ستون های قسمت دوم (goto) غیرترمینال ها را نشان می دهند.

برای تکمیل قسمت action و goto به صورت زیر عمل می کنیم

- اگر a یک ترمینال باشد و در حالت S_i با دیدن a به خانه S_j برویم در $action[i, a]$ قرار می دهیم S_j
- در حالت S_i به ازای تمامی دستورات $A \rightarrow \alpha$ ، به ازای تمام اعضای مجموعه $Follow(A)$ (مثلا b) در $action[i, b]$ قرار می دهیم R_n به طوریکه n شماره دستور $A \rightarrow \alpha$ است.
- اگر در حالت S_i دستور $S' \rightarrow S$ قرار داشته باشد در $action[i, \$]$ فرمان $accept$ قرار می گیرد.
- برای تکمیل بخش goto کافی است چنانکه از حالت S_i با یک غیر ترمینال (مثلا A) به حالت S_j می رویم در $goto[i, A]$ قرار می دهیم j

مثال. جدول تجزیه دیاگرام صفحه قبل را رسم کنید.

$Follow(E) = \{+,), \$\}$

$Follow(T) = \{*, +,), \$\}$

$Follow(F) = \{*, +,), \$\}$

	Action						goto		
	id	+	*	()	\$	E	T	F
S_0	S_5			S_4			1	2	3
S_1		S_6				Accept			
S_2		R_2	S_7		R_2	R_2			
S_3		R_4	R_4		R_4	R_4			
S_4	S_5			S_4			8	2	3
S_5		R_6	R_6		R_6	R_6			
S_6	S_5			S_4				9	3
S_7	S_5			S_4					10
S_8		S_6			S_{11}				
S_9		R_1	S_7		R_1	R_1			
S_{10}		R_3	R_3		R_3	R_3			
S_{11}		R_5	R_5		R_5	R_5			

جهت تجزیه یک رشته با استفاده از جدول SLR مراحل زیر را طی می کنیم.

ابتدا علامت $\$$ را به انتهای رشته ورودی اضافه می کنیم و حالت اولیه به پشته اضافه می گردد.

۱- **انتقال:** اگر $action[i, a] = S_n$ ، برنامه تجزیه کننده ابتدا نماد ورودی a و سپس n را به بالای پشته منتقل میکند.

۲- **کاهش:** اگر $action[i, a] = R_n$ باشد یک دستگیره یافت شده است که باشد کاهش یابد اگر n شماره قاعده تولید $A \rightarrow \beta$ باشد، برنامه تجزیه کننده دنباله β را از بالای پشته حذف می کند و پس از حذف β اگر حالت بالای پشته m باشد برنامه تجزیه کننده بتدا A و سپس، عدد موجود در $goto[m, A]$ را به بالای پشته اضافه می کند، نکته این است که برای حذف β از بالای پشته به تعداد $2|\beta|$ از بالای پشته حذف می گردد زیرا بین هر نماد در پشته یک شماره حالت نیز جود دارد.

۳- **پذیرش:** اگر $action[i, a] = accept$ باشد تجزیه رشته ورودی با موفقیت انجام شده است

۴- **خطا:** اگر $action[i, a] = error$ باشد تجزیه رشته ورودی با عدم موفقیت روبرو شده است

مثال. با استفاده از جدول تجزیه زیر رشته $id*id+id$ را تجزیه کنید

- $E' \rightarrow E$
 1) $E \rightarrow E + T$
 2) $E \rightarrow T$
 3) $T \rightarrow T * F$
 4) $T \rightarrow F$
 5) $F \rightarrow (E)$
 6) $F \rightarrow id$

مراحل تجزیه رشته $id*id+id$

پشته	رشته ورودی	خروجی
0	$id*id+id\$$	Shift5
0id5	$*id+id\$$	$F \rightarrow id$
0F3	$*id+id\$$	$T \rightarrow F$
0T2	$*id+id\$$	Shift7
0T2*7	$id+id\$$	Shift5
0T2*7id5	$+id\$$	$F \rightarrow id$
0T2*7F10	$+id\$$	$T \rightarrow T * F$
0T2	$+id\$$	$E \rightarrow T$
0E1	$+id\$$	Shift6
0E1+6	$id\$$	Shift5
0E1+6id5	$\$$	$F \rightarrow id$
0E1+6F3	$\$$	$T \rightarrow F$
0E1+6T9	$\$$	$E \rightarrow E + T$
0E1	$\$$	Accept

جدول تجزیه SLR

		Action						goto		
		id	+	*	()	\$	E	T	F
S_0	S_5				S_4			1	2	3
S_1			S_6				Accept			
S_2			R_2	S_7		R_2	R_2			
S_3			R_4	R_4		R_4	R_4			
S_4	S_5				S_4			8	2	3
S_5			R_6	R_6		R_6	R_6			
S_6	S_5				S_4				9	3
S_7	S_5				S_4					10
S_8			S_6			S_{11}				
S_9			R_1	S_7		R_1	R_1			
S_{10}			R_3	R_3		R_3	R_3			
S_{11}			R_5	R_5		R_5	R_5			

```

while(!EOF){
  IF(action[ $S_i, x$ ] =  $S_j$ )
    Push(x);
    Push( $S_j$ );
    x = next Token();
  else IF(action[ $S_i, x$ ] =  $R : A \rightarrow \alpha$ )
    For(i = 0 to  $2|\alpha|$ )
      pop();
      Push(A);
      Push(goto( $S_i, A$ ))
  else IF(action(s, x) = Accept) Then
    return
  else
    error();
} // end

```

با توجه به توضیحات صفحه قبل می توان الگوریتم تجزیه یک رشته را به این شکل نوشت.
 نکته: اگر گرامری SLR باشد آنگاه LALR و CLR نیز هست.