

# تجزیه پایین به بالا

## ۱ - ۵ مقدمه

در روش تجزیه پایین به بالا مراحل تجزیه از ترمهای پایانی درون جمله داده شده آغاز، و به سر ترم گرامر خاتمه می یابد. تجزیه گرهای پایین به بالا گرامرهای گسترده تری نسبت به تجزیه گرهای بالا به پایین را می پوشانند. در این فصل انواع روشهای اتوماتیک تجزیه پایین به بالا مطرح خواهد شد. چگونگی ترسیم جداول تجزیه، به روشهایی موسوم به، LALR، SLR، LR مطرح می گردد. نکته قابل توجه استفاده از گرامرهای مبهم است که موضوع آخر این فصل را به خود اختصاص می دهد. روش LR(K) برای تجزیه پایین به بالا K ترم بعدی را در جمله مورد کامپایل در هر مرحله از تجزیه استفاده می نماید. چون عمل خواندن لغات از داخل جمله ورودی مورد کامپایل از چپ به راست انجام می شود حرف L که مخفف Left to Right می باشد، در نام این روش ظاهر شده است. حرف R مخفف Rightmost derivation یا استنتاج راست است. کلمه LALR مخفف Lookahead LR است. روش LALR(k) نیز برای دسته ای محدودتر از گرامرها نسبت به روش LR(k) مورد استفاده قرار می گیرد. کلمه SLR مخفف Simple LR است. در این فصل روشهای LR(1)، LALR(1)، SLR(1) مورد بررسی قرار خواهد گرفت.

## ۲ - ۵ اصول تجزیه پایین به بالا

اصولا" در روش تجزیه پایین به بالا پس از اینکه تحلیلگر نحوی لغات را دریافت نمود دو عمل را ممکن است انجام دهد. تحلیلگر یا لغت را مستقیما" به داخل یک پشته به نام پشته تجزیه انتقال می دهد و یا اینکه بر اساس لغت دریافت شده، ترمهای بالای پشته که با سمت راست یک قاعده از قواعد گرامر مطابقت دارند را از بالای پشته خارج نموده، با ترم میانی سمت راست قاعده جایگزین می کند. با جایگزینی ترمهای بالای پشته با ترم میانی سمت چپ قاعده مربوطه در واقع چند ترم بالای پشته به یک ترم کاهش یافته اند. لذا، این عمل را در اصطلاح عمل کاهش یا Reduce گویند. عمل انتقال لغات دریافتی از تحلیلگر لغوی به داخل پشته تجزیه را در اصطلاح عمل انتقال یا Shift گویند. عمل کاهش یا Reduce بر اساس یک قاعده از گرامر انجام می شود. برای نمونه ترمهای بالای پشته تجزیه که عینا مشابه ترمهای سمت راست قاعده شماره مثلا" n هستند، به ترم سمت چپ قاعده که یک

ترم میانی یا سرترم گرامر است کاهش داده می شود. این عمل را بطور خلاصه با دستور  $Rn$  مشخص می کنند. دستورالعمل  $Rn$  مشخص می کند که عمل Reduce بر اساس قاعده شماره  $n$  انجام می شود. اصولاً در روش تجزیه پایین به بالا عمل خواندن لغات مثل قبل از چپ به راست جمله داده شده انجام می شود. ترمها بر اساس قواعد زبان، دسته بندی وبه يك ترم میانی در سمت راست قواعد کاهش می یابند و یا در اصطلاح خارجی Reduce داده می شوند. يك جمله یا برنامه در داخل يك فایل متن یا در اصطلاح Text قرار می گیرد، در انتهای هر فایل متن علامت خاتمه فایل ظاهر می شود، هر جمله داده شده نهایتاً در صورت صحت از لحاظ فرم گرامری به سرترم گرامر کاهش داده می شود. بنابراین پس از سرترم گرامر همواره انتظار مشاهده علامت خاتمه فایل می رود.

علامت خاتمه فایل در اینجا با کرکتر \$ مشخص می شود. جهت کسب اطمینان از ظهور علامت خاتمه فایل پس از سرترم گرامر چنانچه برای نمونه سرترم گرامر  $S$  باشد، قاعده

$$S' \rightarrow S\$$$

را به ابتدای گرامر افزوده، در اصطلاح گرامر را به فرم توسعه یافته یا Extended تبدیل می نمایند. بنابر این به ابتدای گرامر عبارت چهار عمل اصلی با سرترم  $E$ ، قاعده:

$$E' \rightarrow E\$$$

افزوده شده، در اصطلاح گرامر عبارات به فرم توسعه یافته تبدیل می شود.

- 0  $E' \rightarrow E\$$
- 1  $E \rightarrow E+T$
- 2  $E \rightarrow E-T$
- 3  $E \rightarrow T$
- 4  $T \rightarrow T * F$
- 5  $T \rightarrow T / F$
- 6  $T \rightarrow F$
- 7  $F \rightarrow id$
- 8  $F \rightarrow no$
- 9  $F \rightarrow (E)$

بنابر گرامر فوق، صحت عبارت  $(a-b) * c/d$  سنجیده می شود. برای این منظور از يك پشته تجزیه، جهت حفظ فرمهای جمله ای استفاده می شود.

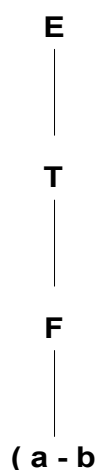
عملیات	ورودی	پشته تجزیه
S,S	$(a-b)*c/d$ \$	(a
R7,R6,R3	$-b)*c/d$ \$	(E
S,S	$-b)*c/d$ \$	(E-b
R7	$) *c/d$ \$	(E-F
R6	$) *c/d$ \$	(E-T
R2	$) *c/d$ \$	(E
S	$) *c/d$ \$	(E)
R9	$) *c/d$ \$	F
R6	$) *c/d$ \$	T
S,S	$/d$ \$	T*c
R7	$/d$ \$	T*F
R4	$/d$ \$	T
S,S	\$	T/d
R7	\$	T/F
R5	\$	T
R3,S,R0	\$	E

شکل ۱ و ۴ - مراحل تجزیه جمله  $(a-b)*c/d$

در جدول فوق ، نکته قابل توجه انتخاب عمل کاهش و یا انتقال در هر مرحله از تجزیه پایین به بالا است. تصمیم گیری در مورد انتخاب عمل کاهش (Reduce) و یا انجام عمل انتقال (Shift)، به موقعیت کنونی پشته و فرم جمله موجود در آن و بالاخره چگونگی ترم موجود بر سر ورودی که در اصطلاح ترم پیش بینی یا Look ahead نامیده می شود، وابسته است. برای نمونه در شروع عمل تجزیه ابتدا ' ( ' و سپس ' a ' از ورودی خوانده می شود. چون ترم بعدی موجود بر سر ورودی اکنون علامت منها می باشد و طبق گرامر و بنا بر قاعده :

2  $E \rightarrow E-T$

حتما قبل از منها يك E باید وجود داشته باشد ، تصمیم به کاهش a بر اساس قاعده شماره هفت و یا بطور مختصر تصمیم به انجام دستورالعمل R7 و سپس R6 و نهایتا " R3 گرفته شد. لذا، تا این مرحله درخت تجزیه بصورت زیر ایجاد می شود



در مرحله بعدی با انجام عمل S یا در واقع عمل انتقال لغت ' - ' از سر ورودی خوانده می شود. اکنون طبق گرامر پس از علامت ' - ' باید در ورودی T ظاهر شود. لذا، ترم بعدی یعنی 'b' از ورودی به داخل پشته تجزیه انتقال یا در اصطلاح Shift داده می شود.

حالا، بر سر ورودی ترم ' ( ' قرار گرفته است. قبل از ' ( ' بر طبق قاعده:

9 F > (E)

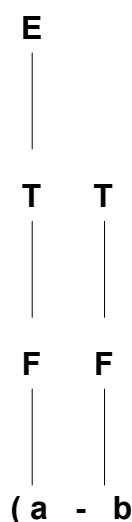
باید E ظاهر شود. تا این مرحله طبق جدول تجزیه بر سر پشته فرم جمله ای

'(E-b'

قرار دارد و ترم پیش بینی ' ( ' است. لذا، جهت رسیدن به E قبل از ' ( ' عمل R7 و سپس عمل R6 انجام می شود. بنابراین فرم جمله ای موجود در پشته تجزیه به صورت

'(E-T'

تبدیل می شود. تا این مرحله از تجزیه پایین به بالا درخت تجزیه بصورت زیر است. فرمهای جمله ای ایجاد شده نیز در داخل پشته تجزیه در بالا مشخص می باشند.



حالا ترم پیش بینی ' ( ' و بر سر پشته تجزیه ترم T موجود است. اما بلا فاصله T با E جایگزین نشده، دستورالعمل R3 اجرا نمی شود. باید توجه نمود که همواره تنها ترم پیش بینی عامل تصمیم گیرنده نیست بلکه، چگونگی فرم جمله ای در بالای پشته تجزیه نیز در تصمیم گیری و انتخاب نوع عمل کاهش و یا انتخاب عمل انتقال موثر است. لذا، در این مرحله عمل R2 انتخاب شده و بر سر پشته تجزیه E-T به E بر طبق قاعده شماره ۲ کاهش داده می شود. تا این مرحله از تجزیه پایین به بالا درخت تجزیه به صورت زیر است:



### ۳-۵ طرح روشی برای ایجاد جدول تجزیه LR(1)

در این بخش چگونگی ایجاد الگوریتم تجزیه پایین به بالا استدلال می شود. الگوریتم حاصل بطور خلاصه در انتها ارائه شده است. الگوریتم حاصل تجزیه LR(1) نامیده می شود. این الگوریتم مبتنی بر تحلیل روش تجزیه که در بالا توضیح داده شد ایجاد میشود.

#### ۱-۳-۵ تولید الگوریتم

در بخش قبل مشاهده کردید که در حالت کلی برای انجام عمل تجزیه نیاز به استفاده از يك پشته تجزیه می باشد. دو عمل صورت می گیرد. یکی انتقال و دیگری کاهش. عمل انتقال و کاهش بر اساس چگونگی ترم پیش بینی بر سر ورودی و وضعیت کنونی پشته تجزیه مشخص می شد. در بعضی از تجزیه گره های LR نیاز به بیش از يك ترم پیش بینی برای عمل تجزیه است، برای نمونه اگر حداکثر K ترم پیش بینی نیاز باشد، تجزیه گر را LR(K) می نامند.

اکنون با يك استدلال ساده نشان داده خواهد شد که چگونه می توان بصورت اتوماتیک بر اساس جدول تجزیه که در مورد آن بحث خواهد شد عمل تجزیه پایین به بالا را انجام داد. برای نمونه گرامر زیر را در نظر بگیرید:

- 0)  $S' \longrightarrow S\$$
- 1)  $S \longrightarrow aBb$
- 2)  $S \longrightarrow AB$
- 3)  $A \longrightarrow bA$
- 4)  $A \longrightarrow b$
- 5)  $B \longrightarrow Bb$
- 6)  $B \longrightarrow a$

بر طبق گرامر فوق نهایتاً در بالای درخت تجزیه باید سر ترم  $S'$  ظاهر شود. بنابر این آخرین مرحله در تجزیه پایین به بالا کاهش  $S\$$  به  $S'$  است. اما قبل از اینکه بتوان این عمل را انجام داد در ورودی ابتدا باید ترم  $S$  ظاهر شود. بنابر این می توان گفت در تجزیه پایین به بالا، در آغاز کار تحلیلگر نحوی در انتظار کاهش ورودی یا به عبارت دیگر برنامه یا جمله مورد کامپایل به  $S$  است. می توان این وضعیت انتظار را بصورت زیر برای تحلیلگر مشخص نمود

$S' \longrightarrow .S\$$

در این وضعیت که در انتظار S باید بود. اما برای تشکیل S باید طبق گرامر یا عمل R1 یا عمل R2

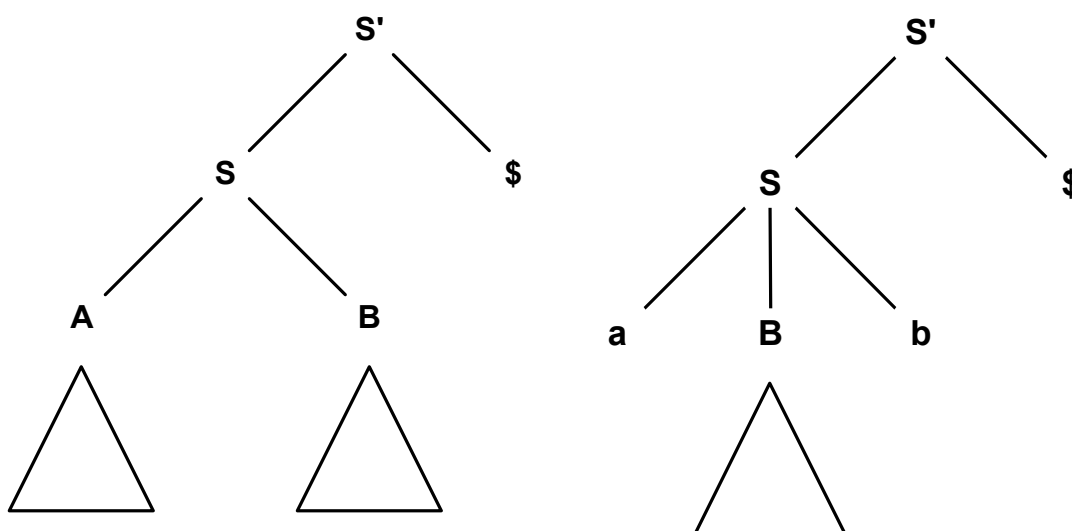
تجزیه شود. یعنی طبق گرامر

$$1) S \longrightarrow aBb$$

$$2) S \longrightarrow AB$$

رشته aBb یا رشته AB در ورودی ظاهر شود. به عبارت دیگر قبل از خاتمه عمل تجزیه باید یکی از

درختهای زیر ایجاد شوند.



پس از مشاهده سر ترم S طبق وضعیت انتظار  $S' \longrightarrow .S\$$  می رود که علامت خاتمه فایل \$

در ورودی ظاهر شود. پس از مشاهده \$ است که می توان عمل R0 را انجام داد. بنابر این تا این

مرحله می توان گفت که وضعیت در شروع عمل تجزیه بصورت زیر می تواند باشد.

$$S' \longrightarrow .S\$$$

$$S \longrightarrow .aBb, \$$$

$$S \longrightarrow .AB, \$$$

در بالا علامت \$ توسط ویرکول از دو گسترش متفاوت S جدا شده است. به این ترتیب مشخص شده

که پس از کاهش سمت راست این دو قاعده به S، انتظار می رود که در ورودی \$ ظاهر شود. به

عبارت دیگر انتظار می رود که پس از تشخیص S ترم پیش بینی \$ باشد. این حالت  $S' \longrightarrow .S\$$

را با وضعیت روبرو مشخص نمود .

بنابر این تا کنون مشخص شده است که در حالت شروع و قبل از آن هیچ لغتی از جمله مورد کامپایل در ورودی ظاهر نشده است، تحلیلگر نحوی انتظار  $S'$  را دارد، لذا جهت تشکیل  $S'$  در آغاز انتظار کاهش ورودی به  $S$  را دارد. پس از مشاهده سر ترم  $S$  انتظار می رود که علامت خاتمه فایل (  $\$$  ) در ورودی ظاهر شود .

برای مشاهده  $S$  در آغاز کار انتظار مشاهده  $a$  و یا تشکیل ترم میانی  $A$  می رود . طبق وضعیت :

**$S \longrightarrow .aBb, \$$**

انتظار می رود ابتدا  $a$  و سپس  $B$ ، و پس از آن  $b$  در ورودی ظاهر شوند تا بتوان عمل  $R1$  را انجام داد و  $S$  را ایجاد کرد. پیش بینی  $R1$  در صورتی مؤثر است که در سر ورودی  $\$$  وجود داشته باشد. به همین ترتیب طبق وضعیت  **$S \longrightarrow .AB, \$$**  باید ابتدا  $A$  ایجاد شود پس از آن  $B$ ، تا بتوان عمل  $R2$  را انجام داد . بنابر این باید ابتدا در ورودی  $A$  تشخیص داده شود و پس از تشخیص  $A$  باید حتماً در سر ورودی ترم متعلق به مجموعه  $First(B)$  ظاهر شود تا بتوان مطمئن شد که می توان  $B$  را پس از  $A$  دید .

برای ایجاد  $A$  نیز طبق گرامر باید با عمل  $R3$  و یا  $R4$  انجام شود . و بعد از انجام عمل حالا بر طبق انتظاری که در بالا طبق وضعیت  **$S \longrightarrow .AB, \$$**  می رفت، تحلیلگر نحوی باید یک ترم متعلق به  $First(B)$  را در ورودی مشاهده کند تا بتواند به کار خود ادامه دهد (  $B$  را در ورودی تشخیص دهد). بنابر این جهت رسیدن به  $A$  یکی از دو وضعیت زیر در آغاز کار و قبل از دریافت هر گونه لغتی از تحلیلگر لغوی پیش بینی می شود.

**$A \longrightarrow .bA, First(B)$**

**$A \longrightarrow .b, First(B)$**

بنا بر این در حالت شروع و قبل از دریافت اولین لغت از تحلیلگر لغوی وضعیت های مورد انتظار تحلیلگر

نحوی بصورت زیر می باشد.

<b><math>I0: S' \longrightarrow .S\\$</math></b>
<b><math>S \longrightarrow .aBb, \\$</math></b>
<b><math>S \longrightarrow .AB, \\$</math></b>
<b><math>A \longrightarrow .bA, First(B)</math></b>
<b><math>A \longrightarrow .b, First(B)</math></b>

حال فرض کنید که اولین ترم پایانی یا اولین لغت از تحلیلگر لغوی دریافت شود. چه اتفاقی می تواند رخ دهد؟ طبق انتظار تحلیلگر (که در بالا مشخص شده است) این لغت باید b یا a باشد. طبق وضعیت

$S \longrightarrow .aBb, \$$

انتظار مشاهده a و طبق وضعیتهای:

$A \longrightarrow .bA, First(B)$

$A \longrightarrow .b, First(B)$

در آغاز انتظار مشاهده b می رود. اگر در ورودی b ظاهر شود حالت جدید زیر به وجود می آید.

$I1: A \longrightarrow b.A, First(B)$ $A \longrightarrow b., First(B)$
---

حالا اگر در ورودی First(A) ظاهر شود باید به سراغ گسترش A در ادامه وضعیت  $b \bullet A$  رفت، و اگر First(B) در ورودی ظاهر شود باید طبق قاعده  $b \bullet$  عمل کاهش b به A را انجام داد.

بنابر وضعیت  $A \longrightarrow b.A, First(B)$  برای مشاهده A در این حالت، انتظار دیدن A دومی هم در سمت راست وضعیت می رود. اما برای مشاهده A در ورودی طبق قواعد ۳ و ۴ گرامر باید bA یا b در ورودی ظاهر شوند. بعد از این حالت باید در ورودی First(B) ظاهر شود. بنابر این حالت I1 به صورت زیر تبدیل می شود.

$I1: A \longrightarrow b.A, First(B)$ $A \longrightarrow b., First(B)$
$A \longrightarrow b.A, First(B)$ $A \longrightarrow b., First(B)$

در حالت I1 طبق وضعیت  $A \longrightarrow b.A, First(B)$  انتظار مشاهده A پس از b در سمت راست وضعیت می رود، همچنین پس از A انتظار مشاهده First(B) می رود. بنابر این، برای تشکیل در ورودی دو وضعیت :

$A \longrightarrow b.A, First(B)$

$A \longrightarrow b., First(B)$



به حالت 11 افزوده شده است. به این ترتیب در این حالت طبق یکی از دو وضعیت فوق A در ورودی تشخیص داده خواهد شد (وضعیتی که در انتظار تشکیل A در ورودی است) و از این حالت تغییر حالت داده و حالت جدید I5 تشکیل می شود .

**I5: A  $\longrightarrow$  bA. ,First(B)**

حال می توان عمل R3 را انجام داد.

در حالت I1 طبق **A  $\longrightarrow$  b.A ,First(B)** وضعیت ، مشخص شده که در ورودی b ظاهر شده است. اکنون، در صورتیکه ترم پیش بینی عنصری متعلق به First(B) باشد، می توان ادعا کرد که در ورودی A مشاهده شده است. بنابر این می توان به حالت I0 برگشت و اعلام کرد که طبق انتظار A در ورودی مشاهده شد.

اکنون ، در حالت I0 وضعیت **S  $\longrightarrow$  A.B , \$** ، که در انتظار مشاهده A در ورودی بوده به وضعیت جدید:

**S  $\longrightarrow$  A.B , \$**

تبدیل می شود . این وضعیت هسته حالت جدید تری بنام I2 بصورت زیر می شود .

<b>I2: S <math>\longrightarrow</math> A.B , \$</b>
<b>B <math>\longrightarrow</math> .Bb , \$, b</b>
<b>B <math>\longrightarrow</math> .a , \$, b</b>

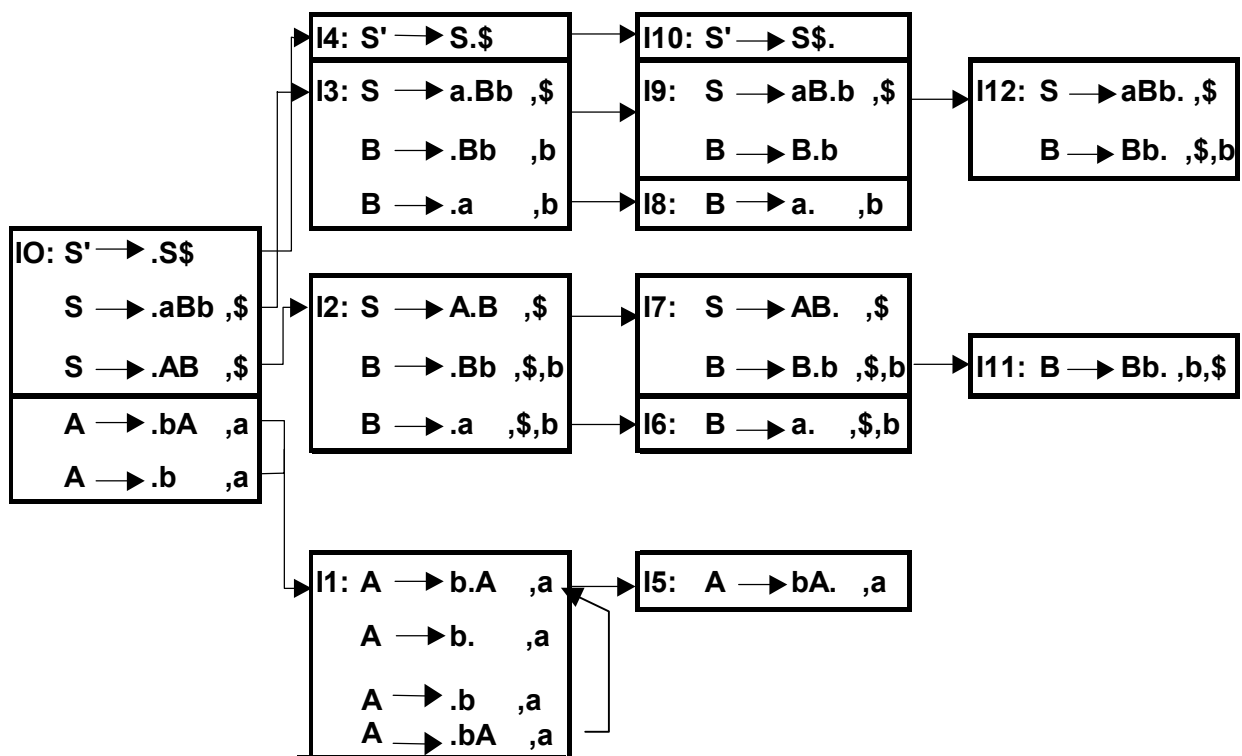
در هسته حالت I2 طبق **S  $\longrightarrow$  A.B , \$** وضعیت انتظار مشاهده B می رود. پس از B نیز همانند S انتظار مشاهده \$. بنابر این باید گسترشهای مختلف b با ترم پیش بینی \$ را ایجاد کرد. پس دو وضعیت زیر به این حالت افزوده می شود :

**B  $\longrightarrow$  .Bb , \$**  
**B  $\longrightarrow$  .a , \$**

حال با توجه به  $B \rightarrow .Bb, \$$  وضعیت انتظار مشاهده B در سمت راست ترم می رود، و پس از B انتظار دیدن b در ورودی. برای مشاهده B طبق گرامر باید Bb یا a در ورودی ظاهر شوند. بنابراین باید دو وضعیت  $B \rightarrow .Bb, b$  و  $B \rightarrow .a, b$  به این حالت افزوده شوند. اما چون از قبل این دو وضعیت با ترم پیش بینی \$ وجود دارند، و می توان ترم پیش بینی b را فقط به آن افزود. بنابراین این دو وضعیت زیر به حالت I2 افزوده شده است.

$B \rightarrow .Bb, \$, b$   
 $B \rightarrow .a, \$, b$

اگر به این ترتیب ادامه داده شود، گراف تجزیه LR(1) بصورت زیر ایجاد می شود:



شکل ۵-۱ گراف تجزیه LR(1)

## ۵-۳-۲ الگوریتم تولید گراف تجزیه LR(1)

بطور خلاصه برای تولید گراف حالات که يك نمونه از آن در شکل ۵،۱ نمایش داده شد، باید مراحل زیر طی شوند:

۱- عمل تولید گراف از ایجاد حالت شروع IO آغاز می شود. چنانچه سر ترم گرامر S باشد، در هسته یا در اصطلاح Kernel این حالت وضعیت

**IO: S'  $\rightarrow$  .S\$** .

می شود ، علامت • قبل از S مشخص می کند که در آغاز انتظار مشاهده S می رود .

۲- برای هر وضعیت

**A  $\rightarrow$  a.B $\beta$  ,  $\delta$**

از يك حالت که در آن a و یا هر رشته ای از ترم ها و یا اینکه تهی می تواند باشد، باید کلیه گسترشهای ترم میانی B بصورت زیر وضعیتهای:

**B  $\rightarrow$  •  $\eta$  , FIRST( $\beta$ )**

را به آن حالت افزود. ترم پیش بینی برای این گسترشها ( FIRST( $\beta$ ) ) است و چنانچه  $\beta$  وجود نداشته باشد ، آنگاه ترم پیش بینی  $\delta$  است. برای نمونه یکی از وضعیت های حالت IO در گراف شکل ۵،۱ وضعیت **S  $\rightarrow$  .AB , \$** است. چون پس از • ترم میانی A قرار دارد، گسترشهای مختلف A با ترم پیش بینی First(B) که مساوی a است، در نظر گرفته شده است.

**A  $\rightarrow$  .bA , a**

**A  $\rightarrow$  .b , a**

۳- ترم پیش بینی برای وضعیت های خود بازگشتی با فرم کلی :

**A  $\rightarrow$  • A  $\alpha$  ,  $\delta$**

**A  $\rightarrow$  •  $\beta$  ,  $\delta$**

بصورت:

**A  $\rightarrow$  • A  $\alpha$  ,  $\delta$ , First( $\alpha$ )**

**A  $\rightarrow$  •  $\beta$  ,  $\delta$ , First( $\alpha$ )**

است . برای نمونه در حالت I2 از گراف شکل ۵،۱ برای وضعیت  $S \rightarrow A.B$  , \$  
طبق معمول دو گسترش ترم میانی B با ترم پیش بینی S ایجاد می شود:

$B \rightarrow .Bb$  , \$  
 $B \rightarrow .a$  , \$

اکنون پس از • ترم میانی B وجود دارد لذا، باید وضعیتهای:

$B \rightarrow .Bb$  , b  
 $B \rightarrow .a$  , b

را به حالت 12 افزود ، بنابر این با ترکیب این دو وضعیت با وضعیت های بالا دو وضعیت زیر حاصل می شود :

$B \rightarrow .Bb$  , \$, b  
 $B \rightarrow .a$  , \$, b

### ۳-۳-۵ خلاصه عملیات در گراف تجزیه

اگر به گراف شکل ۵،۱ توجه نمایید. سه نوع عملیات در آن مستتر است.

۱- عمل کاهش یا Reduce : عمل کاهش در بالا توضیح داده شد. برای نمونه در حالت I11 طبق وضعیت چنانچه بر سر  $B \rightarrow Bb$  , b, \$ ورودی یکی از دو ترم پایانی \$ یا b ظاهر شود، می توان بر طبق قاعده عمل کاهش را انجام داد یا  $B \rightarrow Bb$  به عبارت دیگر می توان طبق قاعده شماره ۵ عمل R5 را انجام داد. در حالت کلی عمل Rn به مفهوم کاهش بر اساس قاعده شماره n است .

۲- عمل انتقال shift: عمل دیگری که در گراف مستتر است، عمل انتقال است. برای نمونه در

حالت I0 با مشاهده b در ورودی، b به داخل پشته تجزیه انتقال داده شده، و سپس به وضعیت

I1 تغییر حالت داده می شود. این عمل را با نماد S1 نمایش می دهند. در حالت کلی عمل Sn

به مفهوم انتقال و سپس گذر به حالت n است.

۳- عمل GoTO: عمل سوم عمل رفتن مستقیم یا در اصطلاح Goto از يك حالت به يك حالت دیگر است. برای نمونه در حالت IO با تشکیل A در سر پشته بلافاصله گذری به حالت ۲ انجام می شود (عمل Goto2 انجام می شود). در حالت کلی با تشکیل يك ترم میانی مورد انتظار بر روی پشته می توان Go to را به حالت جدیدتر انجام داد.

#### ۴-۳-۵ ایجاد جدول تجزیه

جدول تجزیه در واقع نمایش ماتریس گراف تجزیه است. عملیات انتقال و Goto موجب تغییر حالت و یا گذر

بین گراف می شود. برای نمونه جدول تجزیه شکل ۱-۵ را می توان به جدول زیر تبدیل نمود.

	a	b	\$	S	A	B
0	S3	S1	-	4	2	
1	-	S1	-	-	5	-
2	S6	-	-	-	-	7
3	S8	-	-	-	-	9
4	-	-	S10	-	-	
5	R3	-	-	-	-	-
6	-	R6	R6	-	-	-
7	-	S11	R2	-	-	-
8	-	R6	-	-	-	-
9	-	S12	-	-	-	-
10	-		ACCEPT		-	-
11	-	R5	R5	-	-	-
12	-	R5	R1	-	-	-

شکل ۳-۵ جدول تجزیه LR(1)

اکنون با استفاده از جدول فوق می توان به راحتی عمل تجزیه بالا به پایین را انجام داد، برای نمونه عبارت aabbb را توسط جدول فوق بصورت زیر می توان مورد تجزیه پایین به بالا قرار داد.

پشته تجزیه	ورودی	دستورالعمل
0	<u>a</u> abbb\$	S3
0a3	<u>a</u> bbb\$	S8
0a3a8	<u>b</u> bb\$	R6
0a3B	<u>b</u> bb\$	Goto 9
0a3B9	<u>b</u> bb\$	S12
0a3Bb12	<u>b</u> b\$	R5
0a3B	<u>b</u> b\$	Goto 9
0a3B9	<u>b</u> b\$	S12
0a3Bb12	<u>b</u> \$	R5
0a3B	<u>b</u> \$	Goto 9
0a3B9	<u>b</u> \$	S12
0a3B9b12	\$	R1
0s	\$	Goto 4
0s4	\$	S10
0s4\$10		پذیرش
\$		

شکل ۴-۵ مراحل تجزیه LR(1)

مثال : برای مشخص شدن چگونگی عمل تجزیه LR(1) به مثال زیر توجه نمایید:

$S \longrightarrow CC$   
 $C \longrightarrow aC$   
 $C \longrightarrow d$

گرامر ساده فوق را در نظر بگیرید، هدف ایجاد جدول تجزیه LR(1) برای این گرامر است. همانگونه که قبلاً گفته شد، ابتدا باید گرامر را به فرم توسع یافته تبدیل نمود و پس از آن قواعد را شماره گذاری کرد. به این ترتیب گرامر فوق به صورت زیر تبدیل می شود:

0)  $S' \longrightarrow S\$$   
 1)  $S \longrightarrow CC$   
 2)  $C \longrightarrow ac$   
 3)  $C \longrightarrow d$

اکنون می توان در مورد حالات ممکن تجزیه شروع به استدلال نمود. به این ترتیب که در حالت شروع ،  
 \_\_\_\_\_ در وضعیت  $S \rightarrow S\$$  قرار داریم. در اینجا علامت '.' پس از  
 فلش، چگونگی وضعیت را مشخص می کند. هدف تشخیص S و در نهایت مشاهده علامت \$ می  
 باشد.

پس باید در حالت شروع، در ورودی يك S را مشاهده کرد ( انتظار دیدن S می رود )، و پس از آن  
 انتظار دیدن علامت \$ می رود. اما طبق گرامر برای مشاهده S باید در ورودی CC دیده شود، لذا  
 وضعیت در حالت شروع به حالت زیر تبدیل می شود:

$S' \rightarrow S\$$   
 $S \rightarrow CC$

طبق وضعیت ،  $S \rightarrow S\$$  پس از S انتظار دیدن \$ می رود. پس، پیش بینی می شود که  
 بعد از C نیز در ورودی علامت \$ مشاهده شود. حال در این وضعیت انتظار می رود که در اولین مرحله  
 در ورودی يك C و بعد از این C يك C دومی نیز مشاهده شود:

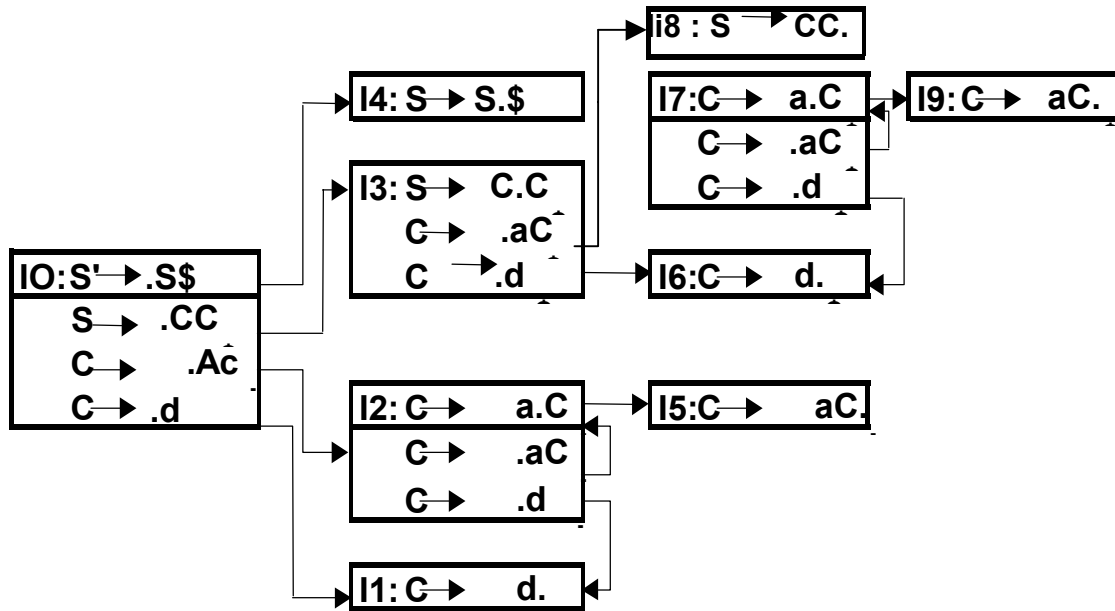
$S' \rightarrow S\$$   
 $S \rightarrow CC, \$$   
 $C \rightarrow aC, First(C)$   
 $C \rightarrow d, First(C)$

همانگونه که مشاهده می شود، طبق وضعیت  $S \rightarrow CC, \$$  انتظار مشاهده يك C در ورودی  
 را داریم. لذا، باید C در ورودی تشخیص داده شود. برای تشخیص C باید طبق گرامر یا aC و یا ترم  
 پایانی d مشاهده شوند. در حالت شروع اگر در ورودی d ظاهر شود حالت جدید I1 تولید می شود.  
 حالت I1 تنها شامل وضعیت:

$C \rightarrow d, First(C)$

می باشد. لذا، پس از مشاهده d در ورودی می توان، به حالت I0 بازگشت و اعلام کرد که در  
 وضعیت \_\_\_\_\_  $S \rightarrow CC$  انتظار مشاهده يك C می رفت و حالا در ورودی C تشخیص

داده شده است. به این ترتیب، حالت جدید I3 با وضعیت در هسته آن ظاهر می شود. پس اینجا تلاش برای دیدن C دوم در ورودی و مشاهده \$ بعد از C دوم آغاز می شود. بطور کلی گراف تجزیه بر طبق استدلال فوق بصورت زیر ایجاد می شود:



شکل ۵-۵ گراف تجزیه LR(1)

که جدول تجزیه برای گراف فوق بصورت زیر می باشد:

	Action			Go To	
	a	d	\$	S	C
0	S2	S1	-	4	3
1	R3	R3	-	-	-
2	S2	S1	-	-	5
3	S7	S6	-	-	8
4	Accept			-	-
5	R2	R2	-	-	-
6	-	-	R3	-	-
7	S7	S6	-	-	9
8	-	-	R1	-	-
9	-	-	R2	-	-

شکل ۵-۶ جدول تجزیه LR(1)



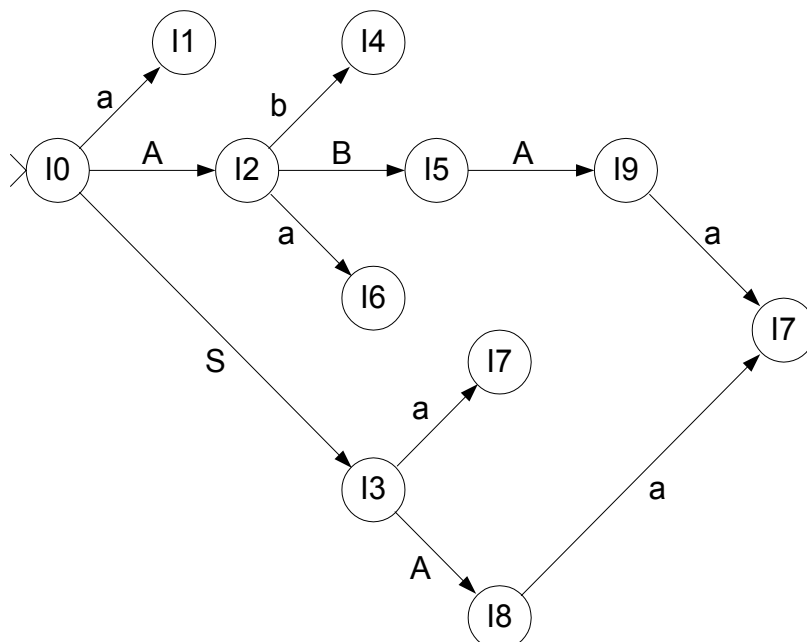
مثال : جدول تجزیه برای LR(1) گرامر زیر ایجاد کنید:

$S' \rightarrow S\$$   
 $S \rightarrow SA$   
 $S \rightarrow AB$   
 $B \rightarrow BA$   
 $B \rightarrow b$   
 $A \rightarrow Aa$   
 $A \rightarrow a$

همانگونه که مشاهده می شود گرامر فوق در فرم توسعه یافته می باشد، پس می توان گراف تجزیه را بلافاصله ایجاد کرد. چون تعداد حالات ممکن نسبتاً زیاد می باشد، هر يك از حالات را بصورت جداگانه مشخص نموده سپس با استفاده از يك گراف ارتباط بین حالات را ترسیم می کنیم.

<b>I0: <math>S' \rightarrow .S\\$</math></b> $S \rightarrow .Sa, \$, a$ $S \rightarrow .AB, \$, a$ $A \rightarrow .Aa, b, a$ $A \rightarrow .a, b, a$	<b>I3: <math>S' \rightarrow S. \\$</math></b> $S \rightarrow S.A, \$, a$ $A \rightarrow .Aa, \$, a$ $A \rightarrow .a, \$, a$	<b>I7: <math>A \rightarrow a. , \\$, a</math></b>
<b>I1: <math>A \rightarrow a. , b, a</math></b>	<b>I4: <math>B \rightarrow b. , \\$, a</math></b>	<b>I8: <math>S \rightarrow SA. , \\$, a</math></b> $A \rightarrow A.a, \$, a$
<b>I2: <math>S \rightarrow A.B, \\$, a</math></b> $A \rightarrow A.a, b, a$ $B \rightarrow .BA, \$, a$ $B \rightarrow .b, \$, a$	<b>I5: <math>S \rightarrow AB. , \\$, a</math></b> $B \rightarrow B.A, \$, a$ $A \rightarrow .Aa, \$, a$ $A \rightarrow .a, \$, a$	<b>I9: <math>B \rightarrow BA. , \\$, a</math></b> $A \rightarrow A.a, \$, a$
	<b>I6: <math>A \rightarrow Aa. , a, b</math></b>	<b>I10: <math>A \rightarrow Aa. , \\$, a</math></b>

گراف تجزیه بصورت زیر است:



جدول تجزیه LR(1) و گرامر در زیر ارایه شده است.

	a	b	\$	S	A	B
O	S1	-	-	13	12	-
1	R6	R6	-	-	-	-
2	S6	S4	-	-	-	5
3	S7	-	پذیرش	-	8	-
4	R4	-	R4	-	-	-
5	R2 , S7	-	R2	-	9	-
6	R5	R5	-	-	-	-
7	R6	-	R6	-	-	-
8	R1,S10	-	R1	-	-	-
9	R3,S10	-	R3	-	-	-
10	R5	-	R5	-	-	-

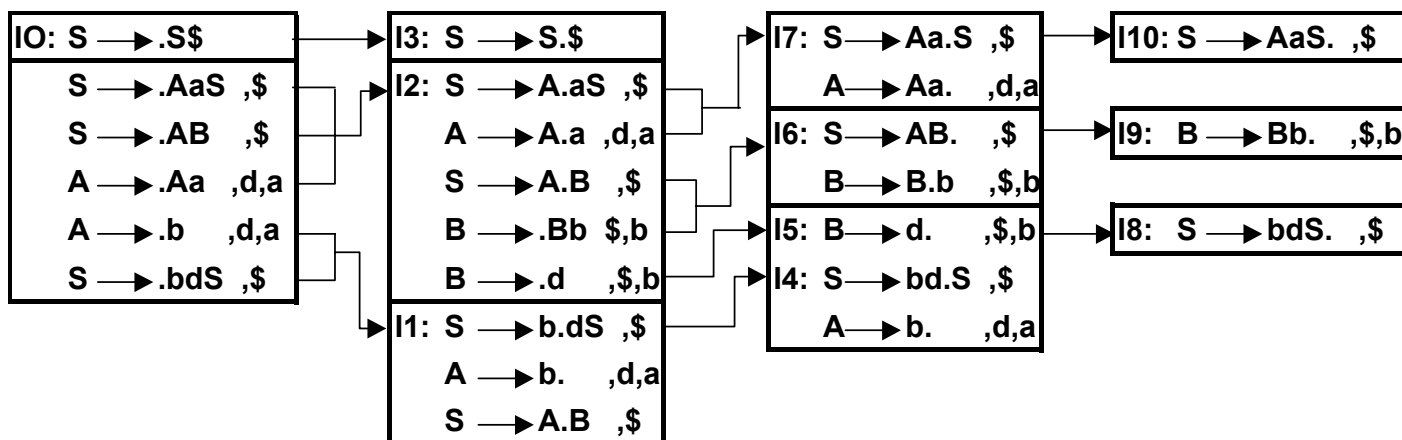
همانگونه که مشاهده می شود، در سه حالت I5, I8 و I9 در داخل جدول تجزیه مشخص نیست که با مشاهده a آیا باید عمل کاهش و یا عمل انتقال را انجام داد. در اصطلاح، در این سه حالت اختلال در کاهش و انتقال وجود دارد، لذا، گرامر LR(1) نیست.

اختلال در کاهش و انتقال را در اصطلاح Shift Reduce Conflict نیز می گویند. این مشکل موجب می شود که با در دست داشتن يك ترم پیش بینی نتوان تصمیم گرفت عمل Shift باید انجام شود یا عمل Reduce، ممکن است Reduce Reduce Conflict نیز در برخی موارد ایجاد شود.

مثال : جدول تجزیه LR(1) برای گرامر زیر ایجاد کنید:

$S' \longrightarrow S\$$   
 $S \longrightarrow AaS$   
 $S \longrightarrow bdS$   
 $S \longrightarrow AB$   
 $A \longrightarrow Aa$   
 $A \longrightarrow b$   
 $B \longrightarrow Bb$   
 $B \longrightarrow d$

گراف تجزیه LR(1) برای گرامر ذکر شده بصورت زیر است:



گرامر فوق LR(1) نیست. زیرا در حالت I1 با مشاهده d در ورودی مشخص نیست که آیا عمل R5 و یا عمل S4 باید انجام شود. عبارت دیگر نمی توان مشخص کرد که آیا عمل انتقال باید انجام شود و یا عمل کاهش، بنابراین، در این حالت اختلال در کاهش و انتقال و یا در اصطلاح Shift Reduce Conflict وجود دارد.

## ۴-۵ مشکل گرامرهای LR(1)

اگر توجه نموده باشید در گرامرهای LR(1) جدول تجزیه نسبت به تعداد قواعد آن بسیار بزرگ می باشد. برای نمونه برای گرامر کوچک:

$S \rightarrow CC$   
 $C \rightarrow aC$   
 $C \rightarrow d$

جدول تجزیه دارای ۱۰ ردیف و ۵ ستون بود. به این ترتیب جدول تجزیه برای يك گرامر با فقط ۳ قاعده تعداد ۵۰ خانه داشت. برای گرامرهای واقعی با حدود ۱۰۰ قاعده، جدول تجزیه حداقل ۱۰۰۰۰ خانه از حافظه را اشغال می کند. جهت رفع این مشکل سعی شده که گرامر را محدود کنند و گرامرها با امکانات کمتر از گرامرهای LR(1) جهت بیان قواعد استفاده شود و یا اینکه گرامرها بصورت مبهم استفاده شوند. برای مثال گرامر عبارات بصورت:

$E \rightarrow E + T \mid E - T$   
 $T \rightarrow T * F \mid T / F$   
 $F \rightarrow id \mid no \mid '( )'$

را می توان بصورت مبهم و با يك ترم میانی بصورت زیر تعریف کرد:

$$E \longrightarrow E + E \mid E - E \mid E * E \mid E / E \mid (' E ') \mid id \mid no$$

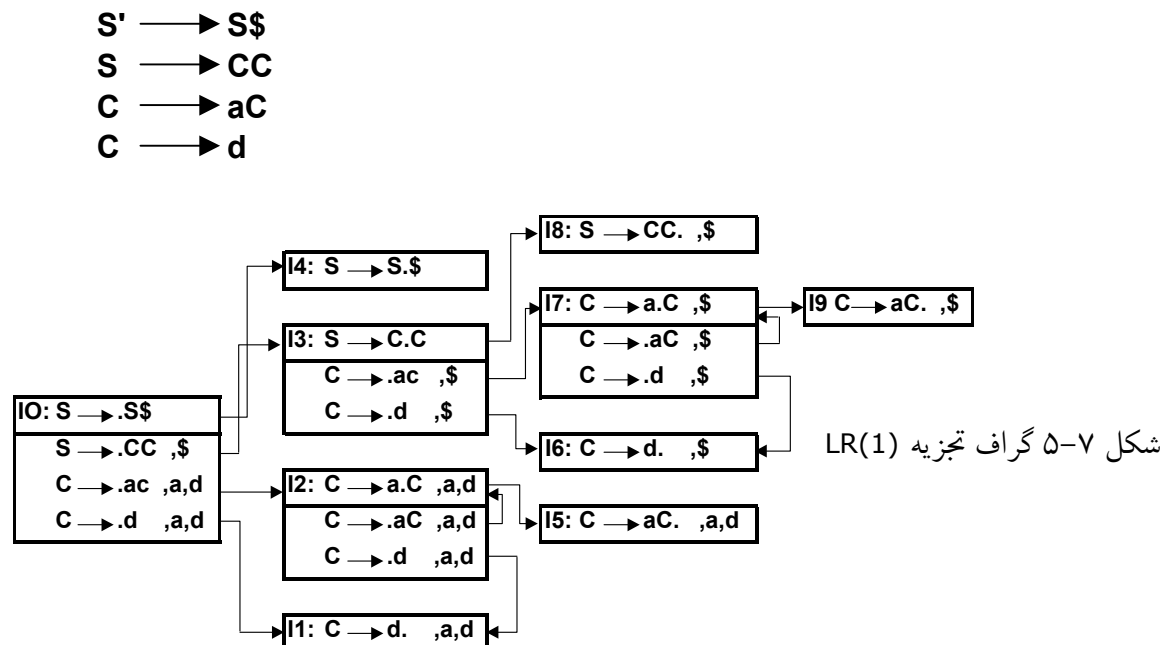
همانطور که مشاهده می کنید، در این گرامر بجای ۳ ترم میانی E, T, F فقط يك ترم میانی استفاده شده، و بجای ۹ قاعده برای فرم کلی عبارات ۷ قاعده استفاده شده است، که جدول را بسیار کوچکتر می نماید، البته ابهام در گرامر مشکلات قابل حلی بوجود خواهد آورد که در مورد آنها کاملاً بحث خواهد شد.

روش دیگر برای کوچکتر نمودن جدول تجزیه پایین به بالا، استفاده از گرامرهای محدودتری به نام LALR و SLR می باشد.

این نوع گرامرها در ادامه بحث توضیح داده خواهد شد.

## ۵-۵ گرامرهای LALR(1)

در هنگام تولید جدول تجزیه LR(1) چنانچه بتوان بدون هیچ گونه مشکلی حالات با وضعیتهای مشابه، اما ترمهای پیش بینی متفاوت را با یکدیگر ادغام نمود. گرامر را LALR(1) می نامند. مشکل به این صورت می تواند باشد که با در دست داشتن يك ترم پیش بینی یا عبارت دیگر يك ورودی بعدی نتوان تصمیم گرفت که چه عملی باید صورت بگیرد. برای نمونه نتوان تصمیم گرفت که آیا عمل کاهش باید انجام شود یا عمل انتقال. برای درك بهتر مسئله به گرامر و گراف تجزیه مربوط به آن در شکل ۵-۷ توجه کنید.



در گراف شکل ۵-۷ دو حالت I2 و I7 دارای وضعیتهای یکسان با ترمهای پیش بینی متفاوت هستند. حالات I1 و I6 و همچنین حالات I5 و I9 نیز به همین صورت می باشند. اگر این حالات را با یکدیگر ادغام نماییم. حاصل بصورت زیر خواهد بود:

الف- حالت I2 با I7 ادغام شده، حالت I2-7 ایجاد می شود.

<div> <div>I7: C → a.C , \$</div> <div>C → .aC , \$</div> <div>C → .d , \$</div> </div>	+	<div> <div>I2: C → a.C , a,d</div> <div>C → .aC , a,d</div> <div>C → .d , a,d</div> </div>	→	<div> <div>I2_7: C → a.C , a,d,\$</div> <div>C → .aC , a,d,\$</div> <div>C → .d , a,d,\$</div> </div>
---	---	--	---	---

ب- حالت I1 با حالت I6 ادغام شده، حالت I1-6 ایجاد می شود.

<div>I1: C → d. , a,d</div>	+	<div>I6: C → d. , \$</div>	→	<div>I1_6: C → d. a,d,\$</div>
-----------------------------	---	----------------------------	---	--------------------------------

ج- حالت I5 با حالت I9 ادغام شده حالت جدید I5-9 ایجاد می شود.

<div>I5: C → aC. , a,d</div>	+	<div>I9: C → aC. , \$</div>	→	<div>I9: C → aC. , \$,a,d</div>
------------------------------	---	-----------------------------	---	---------------------------------

اکنون در جدول تجزیه باید به جای حالت‌های ادغام شده، حالت جدید ادغامی را قرار دهیم.

برای نمونه در بالا حالت‌های I5 و I9 از جدول تجزیه LR(1) حذف و با I5-9 جایگزین می شود. به این ترتیب، هر دستورالعمل انتقال S5 یا S9 با دستورالعمل S5-9 جایگزین می شود. هر عمل Goto به I5-9 تبدیل می شود. جدول تجزیه قبل از جایگزینی و پس از جایگزینی حالت‌های ترکیب شونده، در

زیر مشخص شده است:

	Action			Go To	
	a	d	\$	S	C
0	S2	S1	-	4	3
1	R3	R3	-	-	-
2	S2	S1	-	-	5
3	S7	S6	-	-	8
4	Accept			-	-
5	R2	R2	-	-	-
6	-	-	R3	-	-
7	S7	S6	-	-	9
8	-	-	R1	-	-
9	-	-	R2	-	-

الف - قبل از ادغام

	Action			Go To	
	a	d	\$	S	C
O	S2_7	S1_6	-	4	3
1	R3	R3	R3	-	-
2	S2_7	S1_6	-	-	5_9
3	S2_7	-	S1_6	-	8
4	Accept			-	-
5	R2	R2	R2	-	-
6	-	-	R1	-	-

ب- پس از ادغام

قبل از ادغام باید حالت‌های کاندید را در داخل جدول مشخص کرد. هر جا که دستورالعمل Goto به یکی از حالت‌های ادغامی وجود دارد، نام حالت ادغامی به جای آن قرار می‌گیرد. همین عمل برای دستورالعمل‌های انتقال یا در اصطلاح Shift نیز باید انجام داد.

مثال : جدول تجزیه LR(1) برای گرامر زیر ایجاد کنید:

$E' \rightarrow E\$$   
 $E \rightarrow E + T$   
 $E \rightarrow T$   
 $T \rightarrow T * F$   
 $T \rightarrow F$   
 $F \rightarrow ( E )$   
 $F \rightarrow id$

حالات گراف تجزیه LR(1) برای این گرامر در زیر مشخص شده است:

I0: $E' \rightarrow .E\$$ $E \rightarrow .E+T$ ,+, $E \rightarrow .T$ ,+, $T \rightarrow .T^*F$ ,+, $T \rightarrow .F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,	I7: $F \rightarrow .(E)$ ,+, $E \rightarrow .E+T$ ,+, $E \rightarrow .T$ ,+, $T \rightarrow .T^*F$ ,+, $T \rightarrow .F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,	I13: $F \rightarrow .(E)$ ,+, $E \rightarrow .E+T$ ,+, I14: $T \rightarrow .T^*F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+, I15: $E \rightarrow .E+T$ ,+, $T \rightarrow .T^*F$ ,+, $T \rightarrow .F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,
I1: $F \rightarrow .id$ ,+, I2: $F \rightarrow .(E)$ ,+, $E \rightarrow .E+T$ ,+, $E \rightarrow .T$ ,+, $T \rightarrow .T^*F$ ,+, $T \rightarrow .F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,	I8: $T \rightarrow .F$ ,+, I9: $E \rightarrow .T$ ,+, $T \rightarrow .T^*F$ ,+, I10: $F \rightarrow .(E)$ ,+, $E \rightarrow .E+T$ ,+, I11: $T \rightarrow .T^*F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,	I16: $F \rightarrow .(E)$ ,+, I17: $T \rightarrow .T^*F$ ,+, I18: $E \rightarrow .E+T$ ,+, $T \rightarrow .T^*F$ ,+, I19: $F \rightarrow .(E)$ ,+, I20: $T \rightarrow .T^*F$ ,+, I21: $E \rightarrow .E+T$ ,+, $T \rightarrow .T^*F$ ,+,
I3: $T \rightarrow .F$ ,+, I4: $E \rightarrow .T$ ,+, $T \rightarrow .T^*F$ ,+, I5: $E' \rightarrow .E\$$ $E \rightarrow .E+T$ ,+, I6: $F \rightarrow .id$ ,+,	I12: $E \rightarrow .E+T$ ,+, $T \rightarrow .T^*F$ ,+, $T \rightarrow .F$ ,+, $F \rightarrow .(E)$ ,+, $F \rightarrow .id$ ,+,	

در زیر جدول تجزیه LR(1) و LALR(1) برای این گرامر ایجاد شده است: ( شکل ۹-۵ )

	Action						Go To		
	id	(	)	+	*	\$	F	T	E
0	S1	S2	-	-	-	-	3	4	5
1	-	-	-	R6	R6	R6	-	-	-
2	S6	S7	-	-	-	-	8	9	10
3	-	-	-	R4	R4	R4	-	-	-
4	-	-	-	R2	S11	R2	-	-	-
5	-	-	-	S12	-	Accept	-	-	-
6	-	-	R6	R6	R6	-	-	-	-
7	S6	S7	-	-	-	-	8	9	13
8	-	-	R4	R4	R4	-	-	-	-
9	-	-	R2	R2	S14	-	-	-	-
10	-	-	S16	S15	-	-	-	-	-
11	S1	S2	-	-	-	-	17	-	-
12	S1	S2	-	-	-	-	3	18	-
13	-	-	S19	S15	-	-	-	-	-
14	S6	S7	-	-	-	-	20	-	-
15	S6	S7	-	-	-	-	8	21	-
16	-	-	-	R5	R5	R5	-	-	-
17	-	-	-	R3	R3	R3	-	-	-
18	-	-	-	R1	S11	R1	-	-	-
19	-	-	R5	R5	R5	-	-	-	-
20	-	-	R3	R3	R3	-	-	-	-
21	-	-	R1	R1	S14	-	-	-	-

الف - جدول تجزیه LR(1)

	Action						Go To		
	id	(	)	+	*	\$	F	T	E
O	S1	S2	-	-	-	-	3	4	5
1	-	-	R6	R6	R6	R6	-	-	-
2	S1	S2	-	-	-	-	3	4	10
3	-	-	R4	R4	R4	R4	-	-	-
4	-	-	R2	R2	S11	R2	-	-	-
5	-	-	-	S12	-	Accept	-	-	-
10	-	-	S16	S12	-	-	-	-	-
11	S1	S2	-	-	-	-	17	-	-
12	S1	S2	-	-	-	-	3	18	-
16	-	-	R5	R5	R5	R5	-	-	-
17	-	-	R3	R3	R3	R3	-	-	-
18	-	-	R1	R1	S11	R1	-	-	-

ب- جدول تجزیه LALR(1) حاصل از ترکیب حالات :

(1, 6) - (2, 7) - (3, 8) - (4, 9) - (10, 13) -

(14, 11) - (15, 12) - (16, 19) - (17, 20) - (18, 21) -

برای ادغام حالات، ابتدا باید حالاتی را در نظر گرفت که بدون ایجاد مسئله اختلال در کاهش و انتقال، بسادگی قابل ادغام هستند. پس، حالات I1 و I6 و I3 و I8 و I16 و I19 و I17 و I20 بلافاصله قابل ادغام هستند، با جایگزینی حالات معادل مشاهده می شود که حالات I11 و I14 نیز قابل ادغام هستند، با ادغام این دو حالت می توان نتیجه گرفت که دو حالت I18 و I21 نیز قابل ادغام هستند.



## ۵-۶ گرامرهای SLR(1)

گرامرهای Simple LR(1) یا بطور خلاصه SLR(1) روش دیگری برای تجزیه پایین به بالا می باشد، این نوع گرامرها بسیار محدودتر از گرامرهای نوع LALR(1) و در نتیجه گرامرهای LR(1) می باشند. زیرا، در این نوع از گرامرها فرض بر این است که ترمهای پیش بینی برای یک ترم میانی A وابسته به محل قرار گرفتن A در سمت راست قواعد نمی باشد، بلکه برای هر ترم میانی A، مستقل از چگونگی قرار گیری آن در سمت راست قواعد مختلف، همواره مجموعه Follow(A) شاخص ترمهای پیش بینی برای A است. بنابراین تعداد حالات یا ردیفهای جدول تجزیه برابر با جدول LALR(1) است. زیرا در اینجا همانند LALR(1) ترمهای پیش بینی عاملی برای تفکیک دو حالت از یکدیگر نیستند، یا بعبارت دیگر نمی توان دو حالت مشخص نمود که دارای وضعیتهای یکسان اما ترمهای پیش بینی متفاوت باشند. برای درک بهتر نقش ترمهای پیش بینی در گرامرها، به مثال زیر توجه فرمایید:

**Statement** —→ **Ifst | Whilest**  
**Ifst** —→ **If Condition Then Statement ElsePart**  
**Whilest** —→ **While Condition do Statement**

با دقت در این مثال می بینیم که پس از مشاهده If درون Ifst ترم میانی Condition قرار گرفته است. درون Whilest نیز پس از کلمه While ترم میانی Condition ظاهر شده است. اما در قالب جمله If بعد از Condition انتظار دیدن Then می رود و در whilest ما انتظار دیدن Do را داریم. بنابراین مشاهده می شود که وابسته به اینکه یک ترم میانی سمت راست کدام قاعده قرار گرفته ترمهای پیش بینی آن متفاوت است. اگر ترم میانی Condition در وضعیت زیر باشد:

**Ifst** —→ **If Condition Then Statement ElsePart**

آنگاه ترم پیش بینی برای گسترشهای متفاوت Condition کلمه Then می باشد. یعنی گسترشهای متفاوت Condition با ترم پیش بینی Then را پس از وضعیت فوق به حالت مربوطه افزود. در صورتیکه، اگر Condition در قالب جمله While و در وضعیت زیر قرار داشته باشد:

**Whilest** —→ **While Condition do Statement**

آنگاه ترم پیش بینی برای گسترشهای متفاوت condition کلمه Do خواهد بود. یعنی باید گسترشهای متفاوت Condition با ترم پیش بینی Do را پس از وضعیت فوق به حالت مربوطه افزود.

در گرامرهای SLR(1) گرامر آنقدر محدود است که مهم نیست ترم میانی در داخل چه قاعده ای قرار داشته باشد. در این نوع از گرامرها همواره برای يك ترم میانی ترمهای پیش بینی مساوی با مجموعه Follow آن ترم میانی ست. در مثال فوق اگر بجای Then در تعریف Ifst کلمه Do قرار داده شود،

**Ifst         $\longrightarrow$  If Conditiondo Statement ElsePart**  
**Whilest  $\longrightarrow$  WhileConditiondo Statement**

آنگاه گرامر بصورت

تبدیل می شود( گرامر فوق يك گرامر از نوع SLR(1 می باشد ).  
بعنوان مثال، برای گرامر زیر جدول تجزیه SLR(1 ایجاد می کنیم:

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

در مورد گرامرهای SLR(1 قبل از ایجاد گراف تجزیه باید اقدام به تولید مجموعه پیرو یا در اصطلاح Follow برای ترمهای میانی نمود تا برای هر ترم، ترمهای پیش بینی آن مشخص شود.  
مجموعه پیرو برای ترمهای گرامر فوق بصورت زیر می باشد:

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

اکنون با در دست داشته مجموعه پیرو برای ترمهای میانی بسادگی می توان گراف تجزیه SLR(1 را تولید نمود.

<b>I0:</b> $E' \rightarrow .E\$ \rightarrow I5$ $E \rightarrow .E+T \rightarrow I5$ $E \rightarrow .T \rightarrow I4$ $T \rightarrow .T * F \rightarrow I4$ $T \rightarrow .F \rightarrow I3$ $F \rightarrow .(E) \rightarrow I2$ $F \rightarrow .id \rightarrow I1$	<b>I5:</b> $E' \rightarrow E.\$$ $E \rightarrow E.+T \rightarrow I8$
	<b>I6:</b> $F \rightarrow (E.) \rightarrow I9$ $E \rightarrow E.+T \rightarrow I8$
	<b>I7:</b> $T \rightarrow T *.F \rightarrow I10$ $F \rightarrow .(E) \rightarrow I2$ $F \rightarrow .id \rightarrow I1$
<b>I1:</b> $F \rightarrow id.$	<b>I8:</b> $E \rightarrow E+.T \rightarrow I11$ $T \rightarrow .T * F \rightarrow I11$ $T \rightarrow .F \rightarrow I3$ $F \rightarrow .(E) \rightarrow I2$ $F \rightarrow .id \rightarrow I1$
<b>I2:</b> $F \rightarrow (.E) \rightarrow I6$ $E \rightarrow .E+T \rightarrow I6$ $E \rightarrow .T \rightarrow I4$ $T \rightarrow .T * F \rightarrow I4$ $T \rightarrow .F \rightarrow I3$ $F \rightarrow .(E) \rightarrow I2$ $F \rightarrow .id \rightarrow I1$	<b>I9:</b> $F \rightarrow (E).$
	<b>I10:</b> $T \rightarrow T * F.$
<b>I3:</b> $T \rightarrow F.$	<b>I11:</b> $E \rightarrow E+T.$ $T \rightarrow T *.F \rightarrow I7$
<b>I4:</b> $E \rightarrow T.$ $T \rightarrow T * F \rightarrow I7$	

در زیر جدول تجزیه (1)SLR برای گرامر فوق ارایه شده است. باید توجه داشته باشید که برای این دسته از گرامرها مجموعه های پیرو ترمهای پیش بینی را ایجاد می کنیم، لذا در جدول تجزیه برای کاهش id به F بر طبق قاعده شماره ۶ در زیر ستونهای مربوط به عناصر مجموعه پیرو F دستورالعمل R6 قرار داده شده است.

	Action						Go To		
	id	(	)	+	*	\$	F	T	E
<b>0</b>	<b>S1</b>	<b>S2</b>	-	-	-	-	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	-	-	<b>R6</b>	<b>R6</b>	<b>R6</b>	<b>R6</b>	-	-	-
<b>2</b>	<b>S1</b>	<b>S2</b>	-	-	-	-	<b>3</b>	<b>4</b>	<b>6</b>
<b>3</b>	-	-	<b>R4</b>	<b>R4</b>	<b>R4</b>	<b>R4</b>	-	-	-
<b>4</b>	-	-	<b>R2</b>	<b>R2</b>	<b>S7</b>	<b>R2</b>	-	-	-
<b>5</b>	-	-	-	<b>S8</b>	-	<b>Accept</b>	-	-	-
<b>6</b>	-	-	<b>S9</b>	<b>S8</b>	-	-	-	-	-
<b>7</b>	<b>S1</b>	<b>S2</b>	-	-	-	-	<b>10</b>	-	-
<b>8</b>	<b>S1</b>	<b>S2</b>	-	-	-	-	<b>3</b>	<b>11</b>	-
<b>9</b>	-	-	<b>R5</b>	<b>R5</b>	<b>R5</b>	<b>R5</b>	-	-	-
<b>10</b>	-	-	<b>R3</b>	<b>R3</b>	<b>R3</b>	<b>R3</b>	-	-	-
<b>11</b>	-	-	<b>R1</b>	<b>R1</b>	<b>S7</b>	<b>R1</b>	-	-	-

شکل ۹-۵ جدول تجزیه (1)SLR برای گرامر ساده عبارات

## ۵-۷ گرامرهای مبهم

همانگونه که قبلاً توضیح داده شد، گرامر مبهم گرامری ست که بر اساس آن بیش از يك درخت تجزیه برای جمله داده شده، و در نتیجه دو مفهوم متناقض برای آن بتوان داشت. با استفاده از گرامرهای مبهم می توان اندازه جدول تجزیه پایین به بالا را کوچکتر نمود. برای نمونه گرامر ساده عبارات را که در بخشهای قبل جدول تجزیه LR(1) و LALR(1) و SLR(1) برای آن ایجاد شد را بخاطر بیاورید. جدول تجزیه SLR(1) و LALR(1) برای این گرامر دارای یازده حالت و در مجموع ۱۰۸ خانه از حافظه را اشغال می کرد، با نوشتن گرامر عبارات بصورت مبهم زیر:

**$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid id \mid no$**

تعداد ترمهای میانی کاهش می یابد، در ضمن نشان داده خواهد شد که مبهم بودن گرامر مشکلاتی را نیز بدنبال خواهد داشت:

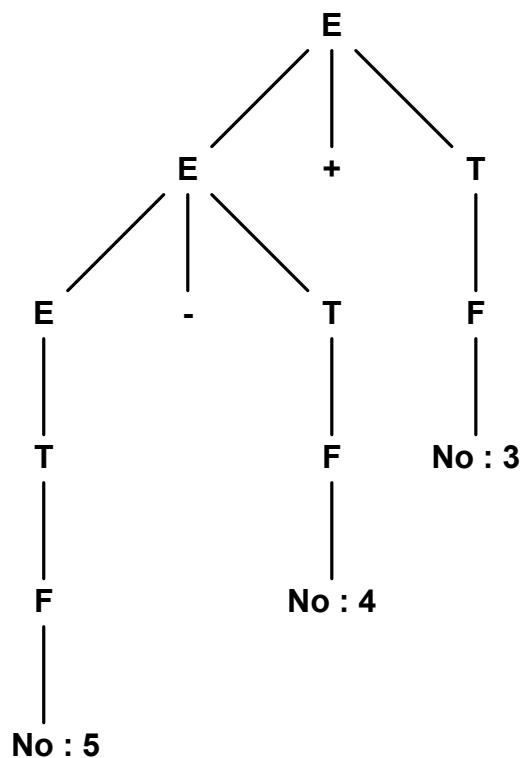
الف- در گرامرهای مبهم اولویت تمام عملگرها یکسان است (اولویت عملگر "ضرب یا تقسیم"، با "جمع و منها" یکسان است).

ب- اجتماع عملگرهای "جمع و ضرب" مشخص نیست.

قواعد مربوط به گرامر عبارات در حالت غیر مبهم بیانگر اولویت و اجتماع عملگرهاست.

- 0)  $E' \rightarrow E\$$
- 1)  $E \rightarrow E + T$
- 2)  $E \rightarrow E - T$
- 3)  $E \rightarrow T$
- 4)  $T \rightarrow T * F$
- 5)  $T \rightarrow T / F$
- 6)  $T \rightarrow F$
- 7)  $F \rightarrow (' E ')$
- 8)  $F \rightarrow id$
- 9)  $F \rightarrow no$

در گرامر فوق "جمع و منها" دارای اجتماع چپ می باشند. برای نمونه درخت تجزیه برای عبارت - 5 + 4 را در نظر بگیرید:



همانگونه که مشاهده می کنید ابتدا باید عبارت 4 - 5 را محاسبه نمود تا بتوان پس از آن حاصل را با 3 جمع کرد. در واقع باید از چپ به راست عمل جمع و تفریق انجام شود. این بخاطر بیان قواعد مربوط به E بصورت خود بازگشتی چپ است. اگر قواعد مربوط به E بصورت خود بازگشتی راست بیان می شد، یعنی بصورت:

$$E \longrightarrow T + E \mid T - E \mid T$$

آنگاه عبارت 3 + 4 - 5 از راست به چپ محاسبه می شد، ابتدا مقدار ۴ + ۳ و سپس 5 - (4 + 3) محاسبه می شد. بر طبق گرامر فوق قبل از اینکه بتوان E را محاسبه کرد، باید T محاسبه شود، زیرا درون T عملیات "ضرب و تقسیم" و در درون E عملیات "جمع و منها" تعریف شده اند. لذا طبق گرامر فوق

- 0)  $E' \longrightarrow E\$$
- 1)  $E \longrightarrow E + E$
- 2)  $E \longrightarrow E * E$
- 3)  $E \longrightarrow (E)$
- 4)  $E \longrightarrow id$

عملیات "ضرب و تقسیم" بر عملیات "جمع و منها" اولویت دارند. با بیان گرامر خلاصه عبارات بصورت مبهم:

اولاً، اولویتها یکسان می شوند، ثانیاً، اجتماع عملگرها نامشخص خواهد بود. نامشخص بودن اجتماع عملگرها و یکسان بودن اولویت آنها در هنگام ایجاد جدول تجزیه SLR(1) برای گرامر مبهم عبارات مشکلاتی را ایجاد می کند.

جدول تجزیه برای گراف فوق در شکل ۱۰-۵ دارای دو نوع عملکرد برای عملگرهای "جمع و ضرب"

<b>I0:</b> $E' \rightarrow .E\$$ $E \rightarrow .E+E$ $E \rightarrow .E^*E$ $E \rightarrow .(E)$ $E \rightarrow .id$	<b>I4:</b> $F \rightarrow (E.)$ $E \rightarrow E.+E$ $E \rightarrow E.*E$	<b>I7:</b> $E \rightarrow (E).$
		<b>I8:</b> $E \rightarrow E^*E.$ $F \rightarrow E.+E$ $E \rightarrow E.*E$
<b>I1:</b> $E \rightarrow id.$	<b>I5:</b> $E \rightarrow E^*.E$ $E \rightarrow .E+E$ $E \rightarrow .E^*E$ $E \rightarrow .(E)$ $E \rightarrow .id$	<b>I9:</b> $E \rightarrow E+E.$ $F \rightarrow E.+E$ $E \rightarrow E.*E$
<b>I2:</b> $E \rightarrow (.E)$ $E \rightarrow .E+E$ $E \rightarrow .E^*E$ $E \rightarrow .(E)$ $E \rightarrow .id$		
<b>I3:</b> $E' \rightarrow E.\$$ $E \rightarrow E+.E$ $T \rightarrow E^*.E$	<b>I6:</b> $E \rightarrow E+.E$ $E \rightarrow .E+E$ $E \rightarrow .E^*E$ $E \rightarrow .(E)$ $E \rightarrow .id$	

در حالات 9 و 8 می باشد، در این دو حالت که اختلال در کاهش و انتقال وجود دارد و مشخص نیست که با مشاهده دو عملگر "جمع و ضرب" آیا عمل کاهش را باید انجام داد یا انتقال را. باید توجه داشت که در گرامرهای SLR(1) ترمهای پیش بینی را مجموعه های پیرو مشخص می کنند. در گرامر فوق برای ترم میانی E باید مجموعه پیرو برای آن محاسبه شود.

	Action						Go To
	id	(	)	+	*	\$	E
O	S1	S2	-	-	-	-	3
1	-	-	R4	R4	R4	R4	-
2	S1	S2	-	-	-	-	5
3	-	-	-	S6	S5	Accept	-
4	-	-	S7	S6	S5	-	-
5	S1	S2	-	-	-	-	8
6	S1	S2	-	-	-	-	9
7	-	-	R3	R3	R3	R3	-
8	-	-	R2	<del>S6,R2</del>	<del>S5,R2</del>	R2	-
9	-	-	R1	<del>S6,R1</del>	<del>S5,R1</del>	R1	-

شکل ۵-۱۰ جدول تجزیه SLR(1) برای گرامر مبهم عبارات

در حالت 8 با توجه به وضعیت  $E \rightarrow E * E$  میتوان گفت که قبلاً يك علامت \* در ورودی دیده شده است. طبق وضعیت  $E \rightarrow E + E$  انتظار مشاهده + در سر ورودی می رود. لذا، با مشاهده يك عملگر + در ورودی چون عملگر \* بر + اولویت دارد عمل کاهش R2 انجام می شود. برای نمونه عبارت  $2 * 3 + 4$  را در نظر بگیرید. مسلماً در اینجا قبل از عمل انتقال علامت + به داخل پشته تجزیه، باید عمل ضرب انجام شود، لذا، به جای S6 عمل R2 انجام می شود.

در حالت 9 با توجه به وضعیت  $E \rightarrow E + E$  می توان گفت که قبلاً يك علامت + در ورودی دیده شده است. طبق وضعیت  $E \rightarrow E * E$  انتظار مشاهده \* در سر ورودی می رود، لذا، با مشاهده يك عملگر \* در ورودی چون عملگر "ضرب" بر "جمع" اولویت دارد، عمل انتقال S5 انجام می شود. برای نمونه عبارت  $2 + 3 * 4$  را در نظر بگیرید. مسلماً در اینجا قبل از انجام عمل "جمع" باید عمل "ضرب" انجام شود، لذا، عملگر \* به داخل پشته تجزیه انتقال داده شده، بجای R1 عمل S5 انجام می شود.

در حالت 8 با توجه به وضعیت  $E \rightarrow E * E$  می توان گفت که قبلاً يك علامت \* در ورودی دیده شده است. طبق وضعیت  $E \rightarrow E * E$  انتظار مشاهده \* دیگری در سر ورودی می رود، لذا، با مشاهده يك عملگر \* در ورودی چون "ضرب" دارای اجتماع چپ است، عمل کاهش R2 انجام می شود، برای نمونه عبارت  $2 * 3 * 4$  را در نظر بگیرید. مسلماً در اینجا قبل از

انتقال علامت \* به داخل پشته تجزیه، باید عمل عمل "ضرب" انجام شود، لذا بجای S5 عمل R2 انجام می شود.

در حالت 9 با توجه به وضعیت  $E \rightarrow E + E$  می توان گفت که قبلاً يك علامت + در ورودی دیده شده است. طبق وضعیت  $E \rightarrow E . + E$  انتظار مشاهده + دیگری در سر ورودی می رود، لذا با مشاهده يك عملگر + در ورودی چون جمع دارای اجتماع چپ است، عمل کاهش R1 انجام می شود. برای نمونه عبارت  $2 + 3 + 4$  در نظر بگیرید. مسلماً در اینجا قبل از انتقال علامت + به داخل پشته تجزیه، باید عمل جمع انجام شود، لذا بجای S6 عمل R1 انجام می شود.

مثال: برای گرامر مبهم جملات شرطی If جدول تجزیه SLR(1) ایجاد کنید.

- 0)  $S' \rightarrow S\$$
- 1)  $S \rightarrow I S e S$
- 2)  $S \rightarrow I S$
- 3)  $S \rightarrow a$

در گرامر فوق S بیانگر جمله، I مخفف If و e مخفف else است.

در اولین مرحله، باید مجموعه پیرو را برای ترمهای میانی و سر ترم گرامر بدست بیاوریم، که این مجموعه در واقع ترمهای پیش بینی را برای ترم میانی مربوطه مشخص می کند.

$$\text{Follow}(S) = \{ \$, e \}$$

مجموعه حالات بصورت زیر می باشد:

IO: $S' \rightarrow .S\$$
$S \rightarrow .I S e S$
$S \rightarrow .I S$
$S \rightarrow .a$
I1: $S \rightarrow a.$
I2: $S \rightarrow I.SeS$
$S \rightarrow I.S$
$S \rightarrow I.SeS$
$S \rightarrow I.S$
$S \rightarrow .a$

I3: $S' \rightarrow S.\$$
I4: $S \rightarrow IS.eS$
$S \rightarrow IS.$
I5: $S \rightarrow ISe.S$
$S \rightarrow .ISeS$
$S \rightarrow .IS$
$S \rightarrow .a$
I6: $S \rightarrow ISeS.$



	<b>I</b>	<b>e</b>	<b>a</b>	<b>\$</b>	<b>S</b>
<b>0</b>	<b>S2</b>	<b>-</b>	<b>S1</b>	<b>-</b>	<b>3</b>
<b>1</b>		<b>R3</b>	<b>-</b>	<b>R3</b>	<b>-</b>
<b>2</b>	<b>S2</b>	<b>-</b>	<b>S1</b>	<b>-</b>	<b>4</b>
<b>3</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>Accept</b>	<b>-</b>
<b>4</b>	<b>-</b>	<b>S5,R2</b>	<b>-</b>	<b>R2</b>	<b>-</b>
<b>5</b>	<b>S2</b>	<b>-</b>	<b>S1</b>	<b>-</b>	<b>6</b>
<b>6</b>	<b>-</b>	<b>R1</b>	<b>-</b>	<b>R1</b>	<b>-</b>

شکل ۱۱-۵ جدول تجزیه SLR(1) برای گرامر مبهم جملات شرطی If

در ردیف 4 از جدول تجزیه چون Else یا e متعلق به نزدیکترین If است، S5 انتخاب شده است.