



مسئله ۱.

گرامر $LL(1)$ با قاعده‌های زیر را در نظر بگیرید.

$$\begin{aligned} S &\rightarrow LB \\ B &\rightarrow ; S ; L \mid L \\ L &\rightarrow (EJ \mid EJ \\ J &\rightarrow) \\ E &\rightarrow a \mid L \end{aligned}$$

الف) برای غیرپایانه‌های این گرامر، مجموعه‌های $first$ و $follow$ آنها را بدست آورید.
ب) جدول پارس $LL(1)$ این گرامر را بدست آورید. (همچنین خانه‌هایی که شامل SYNC می‌شوند را مشخص کنید).
ج) با استفاده از جدول بخش قبل، رشته $(a) = (a, (= a))$ را پارس کنید. (اگر در حین پارس به خطایی برخوردید، آن را نیز مشخص کنید)

حل:

(الف)

$$\begin{aligned} First(S) &= \{ ' ; , ' , ' \} \\ First(B) &= \{ ' ; , ' , ' = ' \} \\ First(L) &= \{ ' (, ' , ' = ' \} \\ First(J) &= \{ ') ' \} \\ First(E) &= \{ ' a , ' (, ' , ' \} \end{aligned}$$

$$\begin{aligned} Follow(S) &= \{ ' \$, ' , ' \} \\ Follow(B) &= \{ ' \$, ' , ' \} \\ Follow(L) &= \{ ' \$, ' , ' , ' = ' , ') ' \} \\ Follow(J) &= \{ ' \$, ' , ' , ' = ' , ') ' \} \\ Follow(E) &= \{ ') ' \} \end{aligned}$$

(ب)

\$)	a	,	(=	;	
synch			LB	LB		synch	S
synch					=L	;S;L	B
synch	synch		,EJ	(EJ			L
synch)				synch	synch	J
	synch	a	L	L			E

(ج)

Action	Input	Stack
LB	$(a) = (a, (= a))\$$	\$
$(EJ$	$(a) = (a, (= a))\$$	LB\$
terminal	$(a) = (a, (= a))\$$	(EJB\$
a	$a) = (a, (= a))\$$	EJB\$
terminal	$a) = (a, (= a))\$$	aJB\$
($) = (a, (= a))\$$	JB\$
terminal	$) = (a, (= a))\$$)B\$
$= L$	$= (a, (= a))\$$	B\$
terminal	$= (a, (= a))\$$	=L\$
$(EJ$	$(a, (= a))\$$	L\$
terminal	$(a, (= a))\$$	(EJ\$
a	$a, (= a))\$$	EJ\$
terminal	$a, (= a))\$$	aJ\$
<i>error, skip', '</i>	$, (= a))\$$	J\$
<i>error, skip')'</i>	$(= a))\$$	J\$
<i>error, MissingTerm</i>	$= a))\$$	J\$
REJECT	$a))\$$	\$

با توجه به اینکه در panic mode روشی برای هندل کردن زمانی که Stack فقط شامل علامت دلار باشد وجود ندارد، رشته رد می‌شود.

در حالت دیگر می‌توان با فرض اینکه ادامه رشته ورودی با دیدن علامت دلار در پشت، skip می‌شود، می‌توان در نظر گرفت که ورودی قبول شده و رشته $a = (a)$ شناسایی می‌شود.

مسئله ۲.

برای هر کدام از زبان‌های زیر یک گرامر مستقل از متن بنویسید.

- $L = \{x \in \{0, 1\}^* \mid x \text{ ends with } 1 \text{ and does not contain the substring } 00\}$
- $L = \{a_i b_i c_i \mid i \geq 1\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains at least three } 1\text{s}\}$
- $L = \{ab^n acab^n a \mid n \geq 0\}$

حل:

(الف)

$$\begin{aligned} S &\rightarrow \bullet A | A \backslash \\ A &\rightarrow AB \mid \varepsilon \\ B &\rightarrow \backslash \bullet | \backslash | \varepsilon \end{aligned}$$

(ب)

می‌توان به کمک لم پمپاژ اثبات کرد که زبان مورد نظر مستقل از متن نیست در نتیجه نمی‌توان برای آن یک گرامر مستقل از متن پیشنهاد داد.

(ج)

$$\begin{aligned} S &\rightarrow A \backslash A \backslash A \backslash A \\ A &\rightarrow AA \mid \bullet \mid \backslash \mid \varepsilon \end{aligned}$$

(د)

$$\begin{aligned} S &\rightarrow aAa \\ A &\rightarrow bAb \mid aca \end{aligned}$$

مسئله ۳.

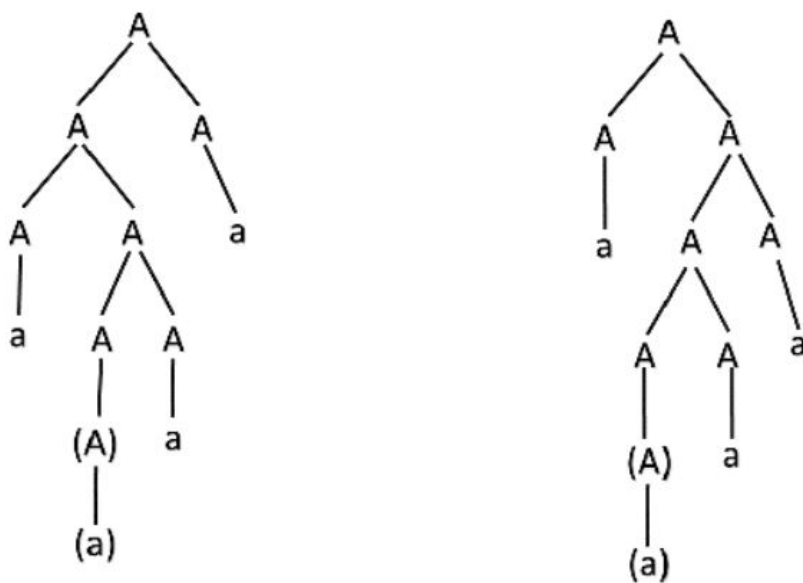
آیا گرامر زیر ابهام دارد؟ اگر دارد دو درخت پارس را رسم کنید که خروجی یکسان تولید می‌کنند.

$$A \rightarrow AA$$

$$A \rightarrow (A)$$

$$A \rightarrow a$$

حل:



شکل ۱: راه حل

مسئله‌ی ۴.

با توجه به دستورگرامر زیر:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid id \end{aligned}$$

۱. آیا دستورگرامر داده‌شده $LL(1)$ است یا خیر؟ دلیل خود را بیان کنید.
 ۲. اگر دستورگرامر $LL(1)$ باشد، جدول تجزیه را ساخته و ارائه دهید.
 ۳. اگر دستورگرامر $LL(1)$ نباشد، آن را به گونه‌ای اصلاح کنید که $LL(1)$ شود، و دستورگرامر اصلاح‌شده را همراه با جدول تجزیه ارائه دهید.
 ۴. رشته ورودی $id * id + id$ را با استفاده از دستورگرامر $LL(1)$ تجزیه کرده و گام‌های تجزیه را ارائه دهید.
- حل:

۱.

خیر.

می‌دانیم که اگر گرامر، مبهم باشد یا left recursion داشته باشد، $LL(1)$ نیست. همانطور که مشخص است در rule های $E \rightarrow E + T \mid T$ و $T \rightarrow T * F \mid F$ ، left recursion داریم، در نتیجه این گرامر $LL(1)$ نیست.

۲.

$LL(1)$ نیست.

۳.

left recursion ها را حذف می‌کنیم، داریم:

$$E \rightarrow E + T \mid T \Rightarrow \begin{cases} E \rightarrow TE' \\ E' \rightarrow +TE' \mid \varepsilon \end{cases}, \quad T \rightarrow T * F \mid F \Rightarrow \begin{cases} T \rightarrow FT' \\ T' \rightarrow *T' \mid \varepsilon \end{cases}$$

بنابراین در نهایت گرامر $LL(1)$ به صورت زیر خواهد بود:

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \varepsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \varepsilon \\ F &\rightarrow (E) \mid id \end{aligned}$$

:

$\text{first}(E) = \{ (, id \}$
 $\text{follow}(E) = \{), \$ \}$
 $\text{first}(E^*) = \{ +, \varepsilon \}$
 $\text{follow}(E') = \{), \$ \}$
 $\text{first}(T) = \{ (, id \}$
 $\text{follow}(T) = \{ +,), \$ \}$
 $\text{first}(T') = \{ *, \varepsilon \}$
 $\text{follow}(T') = \{ +,), \$ \}$
 $\text{first}(F) = \{ (, id \}$
 $f \bullet ll.w(F) = \{ *, +,), \$ \}$

\$	<i>id</i>	()	*	+	
synch	<i>TE'</i>	synch	<i>TE'</i>			<i>E</i>
ε		ε			$+TE'$	<i>E'</i>
	<i>FT'</i>		<i>FT'</i>		synch	<i>T</i>
ε		ε		$*FT'$	ε	<i>T'</i>
	id		(<i>E</i>)	synch		<i>F</i>

.۴

مطابق قسمت آخر مسئله ۱ و به کمک جدول بخش ۳ همین مسئله، رشته ورودی را پارس می‌کنیم.

Action	Input	Stack
TE'	id*id+id\$	E\$
FT'	id*id+id\$	TE'\$
id	id*id+id\$	FT'E'\$
terminal	id*id+id\$	idT'E'\$
*FT'	*id+id\$	T'E'\$
terminal	*id+id\$	*FT'E'\$
id	id+id\$	FT'E'\$
terminal	id+id\$	idT'E'\$
ϵ	+id\$	T'E'\$
+TE'	+id\$	E'\$
terminal	+id\$	+TE'\$
FT'	id\$	TE'\$
id	id\$	FT'E'\$
terminal	id\$	idT'E'\$
ϵ	\$	T'E'\$
ϵ	\$	E'\$
ACCEPT	\$	\$

همانطور که مشخص است، رشته ورودی بدون هیچ اروری پارس شده و قبول می شود.

مسئله ۵.

با توجه به دستورگرامر زیر که یک زبان برنامه‌نویسی ساده را نمایش می‌دهد:

$$\begin{aligned} S &\rightarrow \text{if } E \text{ then } S \text{ else } S \mid \text{id} := E \mid \text{while } E \text{ do } S \mid \text{begin } L \text{ end} \\ L &\rightarrow S; L \mid S \\ E &\rightarrow \text{id} \mid \text{num} \mid E + E \mid E * E \mid (E) \end{aligned}$$

۱. یک جدول تجزیه پیش‌بینی برای دستورگرامر داده‌شده بسازید.
۲. مراحل تجزیه رشته ورودی $\text{if id then id} := \text{num else id} := \text{num}$ با استفاده از جدول تجزیه پیش‌بینی را نشان دهید. حل:

جدول تجزیه پیش‌بینی:

جدول تجزیه پیش‌بینی برای دستورگرامر به شرح زیر است:

	if	then	else	:=	while	do	begin
S	۱ rule				۳ rule		۴ rule
L							
E				۲ rule			

مراحل تجزیه:

رشته ورودی $\text{if id then id} := \text{num else id} := \text{num}$ را با استفاده از جدول تجزیه پیش‌بینی تجزیه می‌دهیم:

پشته	ورودی	عملیات
\$	\$ num := id else num := id then id if	مطابقت با if
S	\$ num := id else num := id then id	استفاده از ۱ rule
S else S then S	\$ num := id else num := id	مطابقت با id
id S else S then S	\$ num := id else num :=	مطابقت با then
E := id S else S then S	\$ num := id else num	مطابقت با id
id := id S else S then S	\$ num := id else num :=	مطابقت با :=
E := E := id S else S then S	\$ num := id else num	مطابقت با num
E := num := id S else S then S	\$ num := id else	مطابقت با :=
id := num := id S else S then S	\$ num := id else	مطابقت با num
num := num := id S else S then S	\$ num := id else	مطابقت با id
id id := num := id S else S then S	\$ num :=	مطابقت با else
E := id id := num := id S else S then S	\$ num	مطابقت با :=
num := id id := num := id S else S then S	\$	مطابقت با num
num num := id id := num := id S else S then S	\$	مطابقت با \$

مسئله‌ی ۶.

به CFG زیر بر روی الفبای $\Sigma = \{=, *, id\}$ توجه کنید:

$$[۱] \quad S \rightarrow L = L$$

$$[۲] \quad S \rightarrow L$$

$$[۳] \quad L \rightarrow id$$

$$[۴] \quad L \rightarrow *L$$

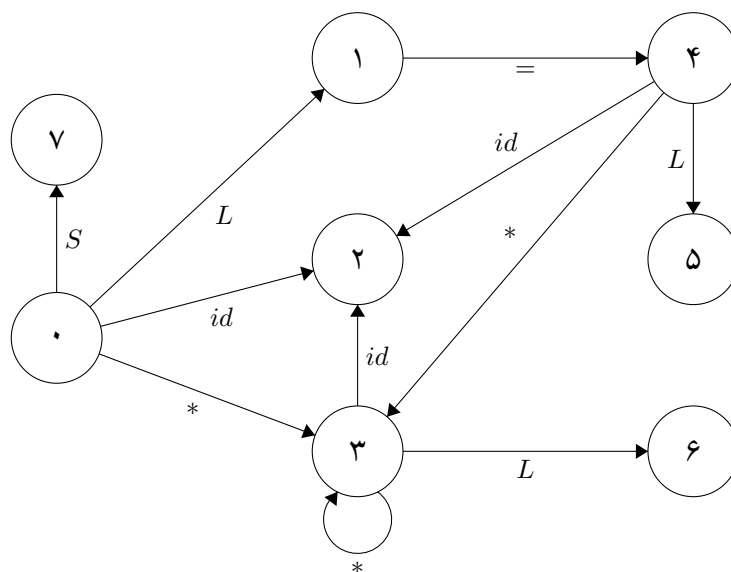
(الف) ترنزشن دیاگرام (۱) SLR و جدول پارس گرامر داده شده را بدست بیاورید

(ب) آیا این گرامر رشته "ID=ID*ID*" را قبول می‌کند؟ فرض کنید که کاربر این رشته را وارد سیستم کرده است. تمام مراحل پارس این رشته را با استفاده از تجزیه‌گری LR با بازیابی پنیک مود نشان دهید.

حل:

۱ (الف)

در ابتدا ارتباط state ها را به هم با یک FSM نمایش داده و در ادامه مجموعه rule های مربوط به هر state را مشخص می‌کنیم.



$$State \bullet = \{S' \rightarrow .S, \quad S \rightarrow .L = L, \quad S \rightarrow .L, \quad L \rightarrow .id, \quad L \rightarrow .*id\}$$

$$State \backslash = \{S \rightarrow L. = L, \quad S \rightarrow L.\}$$

$$State \mathfrak{Y} = \{L \rightarrow id.\}$$

$$State \mathfrak{Z} = \{L \rightarrow *.L, \quad L \rightarrow .id, \quad L \rightarrow .*L\}$$

$$State \mathfrak{C} = \{S \rightarrow L = .L, \quad L \rightarrow .id, L \rightarrow .*L\}$$

$$State \mathfrak{D} = \{S \rightarrow L = L.\}$$

$$State \mathfrak{F} = \{L \rightarrow *.L.\}$$

$$State \mathfrak{V} = \{S' \rightarrow S.\}$$

به کمک ترنزشن دیاگرام بالا و first و follow ها داریم:

$$\begin{aligned}
first(S) &= \{id, *\} \\
follow(S) &= \{\$ \} \\
first(L) &= \{id, *\} \\
follow(L) &= \{=, \$ \}
\end{aligned}$$

state	=	*	id	\$		S	L
۰		۳S	۲S			۷	۱
۱	۴S			۲R			
۲	۳R			۳R			
۳		۳S					۶
۴		۳S	۲S				۵
۵				۱R			
۶	۴R			۴R			
۷				ACCEPT			

(ب)

Stack	Input	Action
۰	$id = id * id * \$$	۲Shift
۰id۲	$= id * id * \$$	۳Reduce
۰L۱	$= id * id * \$$	۴Shift
۰L۱ = ۴	$id * id * \$$	۲Shift
۰L۱ = ۴id۲	$*id * \$$	error , pop stack
۰L۱ = ۴	$*id * \$$	error , skip input
۰L۱ = ۴	\$	push next to stack
۰L۱ = ۴L۵	\$	۱Reduce
۰SV	\$	\$ ACCEPT

به کمک روش panic mode که در اسلاید ۹ توضیح داده شده است، رشته ورودی که حاوی error بود را اکسپت کردیم.

مسئله ۷.

به CFG زیر بر روی الفبای $\Sigma = \{a, b, x\}$ توجه کنید:

- ۱ $S \rightarrow A|xb$
- ۲ $A \rightarrow aAb|B$
- ۳ $B \rightarrow x$

دیاگرام انتقال و جدول تجزیه SLR(۱) (از پایین به بالا) را برای این دستور زبان بسازید. آیا این دستور زبان SLR(۱) است؟ چرا؟

حل:

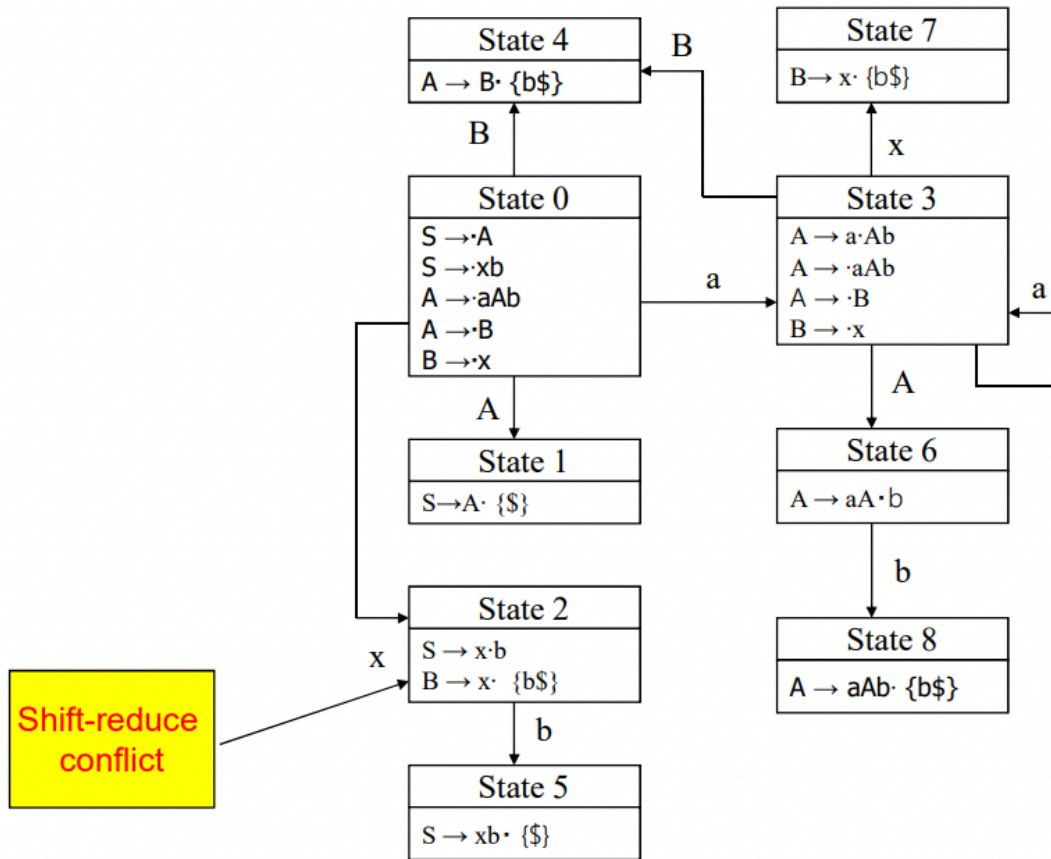
SLR(1) Go_to table

	0	1	2	3	4	5	6	7	8
A				6					
B	4			4					
a	3			3					
b			5				8		
x	2			7					

SLR(1) Action table

state token	0	1	2	3	4	5	6	7	8
b			R5/S		R4		S	R5	R3
\$		R1	R5		R4	R2		R5	R3
a	S			S					
x	S			S					

SLR(1) Transition Diagram



همانطور که مشاهده میشود در جدول یه کانفلیکت از نوع shift – reduce برخورد کرده ایم. بنابراین گرامر داده شده نمیتواند SLR(1) باشد.