

در مورد تکلیف می‌توانید با دوستان خود مشورت کنید ولی تکلیف باید کار خود شما باشد.

-۱

الف) گرامر زیر را فاکتوری از چپ نمایید.

$S \rightarrow S + S \mid S * P$
 $P \rightarrow P * P \mid P * I$
 $I \rightarrow -I \mid (S) \mid D$
 $D \rightarrow 0 \mid 1N$
 $N \rightarrow 0 \mid 1 \mid NN \mid \epsilon$

$S \rightarrow S + S'$
 $S' \rightarrow S \mid P$
 $P \rightarrow P * P'$
 $P' \rightarrow P \mid I$
 $I \rightarrow -I \mid (S) \mid D$
 $D \rightarrow 0 \mid 1N$
 $N \rightarrow 0 \mid 1 \mid NN \mid \epsilon$

ب) بازگشت از چپ گرامر زیر را حذف کنید.

$S \rightarrow AaS \mid Bb \mid Sed \mid d$
 $A \rightarrow BbA \mid Sa \mid c$
 $B \rightarrow Bb \mid SaB \mid g$

S
 A_1
 A_2
 B
 A_3

مرحله اول (i=1): حذف بازگشت از چپ از A_1 که همان S است.

$S \rightarrow AaS' \mid Bbs' \mid ds'$
 $S' \rightarrow eds' \mid \epsilon$

مرحله دوم (i=2): قرار دادن توالین مرتبط با A_1 (همان S) در A_2 (همان A)

$A \rightarrow BbA \mid AaS'a \mid Bbs'a \mid ds'a \mid c$

حذف بازگشت از چپ از توالین A

$A \rightarrow BbAA' \mid Bbs'aA' \mid ds'aA' \mid cA'$
 $A' \rightarrow aS'aA' \mid \epsilon$

مرحله سوم (i=3): قرار دادن توالین مرتبط با A_1 (همان S) در A_2 (همان A) و A_3 (همان B)

$B \rightarrow Bb \mid AaS'aB \mid Bbs'aB \mid ds'aB \mid g$ ← قرار دادن توالین S

$B \rightarrow Bb \mid BbAA'aS'aB \mid Bbs'aA'aS'aB \mid ds'aA'aS'aB \mid cA'aS'aB \mid Bbs'aB \mid ds'aB \mid g$ → قرار دادن توالین A

حذف بازگشت از چپ از توالین B :

$B \rightarrow ds'aA'aS'aBB' \mid cA'aS'aBB' \mid ds'aBB' \mid gB'$
 $B' \rightarrow bB' \mid aAA'aS'aBB' \mid bs'aA'aS'aBB' \mid bs'aBB'$

$S \rightarrow ABmCSDE \mid e$
 $A \rightarrow aA \mid Bd$
 $B \rightarrow bB \mid \epsilon$
 $C \rightarrow gC \mid \epsilon$
 $D \rightarrow cD \mid e$
 $E \rightarrow nE \mid \epsilon$

الف) first و follow برای غیر پایانه های گرامر بالا را به دست آورید. و همچنین تعیین کنید گرامر بالا LL(1) است یا خیر؟

$first(S) = \{a, b, d, e\}$ $follow(S) = \{\$, c, e\}$
 $first(A) = \{a, b, d\}$ $follow(A) = \{b, m\}$
 $first(B) = \{b, \epsilon\}$ $follow(B) = \{m, d\}$
 $first(C) = \{g, \epsilon\}$ $follow(C) = \{a, b, d, e\}$
 $first(D) = \{c, e\}$ $follow(D) = \{n, c, e, \$\}$
 $first(E) = \{n, \epsilon\}$ $follow(E) = \{\$, c, e\}$

بررسی LL(1) بودن:

$first(ABmCSDE) \cap \{e\} = \{a, b, d\} \cap \{e\} = \emptyset$
 $first(aA) \cap first(Bd) = \{a\} \cap \{b, d\} = \emptyset$
 $first(bB) \cap \{\epsilon\} = \emptyset$, $first(bB) \cap follow(B) = \{b\} \cap \{m, d\} = \emptyset$
 $first(gC) \cap \{\epsilon\} = \emptyset$, $first(gC) \cap follow(C) = \{g\} \cap \{a, b, d, e\} = \emptyset$
 $first(cD) \cap \{e\} = \emptyset$
 $first(nE) \cap \{\epsilon\} = \emptyset$, $first(nE), follow(E) = \{n\} \cap \{\$, c, e\} = \emptyset$

گرامر LL(1) است.

ب) در صورتی که گرامر LL(1) نیست، گرامر معادل LL(1) را برای آن نوشته و سپس جدول تجزیه LL(1) را بسازید.

	a	b	c	d	e	g	m	n	\$
S	1	1	5	1	2				3
A	3	4					6		
B		5		6					
C	8	8		8	8	7			
D			9		10				11
E			12	12				11	12

- 1) $S \rightarrow ABmCSDE$
- 2) $S \rightarrow e$
- 3) $A \rightarrow aA$
- 4) $A \rightarrow Bd$
- 5) $B \rightarrow bB$
- 6) $B \rightarrow \epsilon$
- 7) $C \rightarrow gC$
- 8) $C \rightarrow \epsilon$
- 9) $D \rightarrow cD$
- 10) $D \rightarrow e$
- 11) $E \rightarrow nE$
- 12) $E \rightarrow \epsilon$

ج) رشته "annaemgnnecaaan" را با به کارگیری روش رفع خطای panic mode تجزیه کنید. (مجموعه)

همزمانی را مجموعه های follow در نظر بگیرید.

stack	ردی	عمل	stack	ردی	عمل
S \$	annaemgnnecaaan\$	استفاده از قانون 1	CSDE\$	necann\$	خطا، و نه رسیدن به متن
ABmCSDE\$	annaemgnnecaaan\$	استفاده از قانون 3	CSDE\$	ecann\$	استفاده از 8
aABmCSDE\$	annaemgnnecaaan\$	match a	SDE\$	ecann\$	استفاده از 2
ABmCSDE\$	nnaemgnnecaaan\$	خطا، و نه رسیدن به متن	EDE\$	ecann\$	match c
ABmCSDE\$	naemgnnecaaan\$	"	DE\$	cann\$	استفاده از 9
ABmCSDE\$	aemgnnecaaan\$	استفاده از قانون 3	CDE\$	cann\$	match c
aABmCSDE\$	aemgnnecaaan\$	match a	DE\$	ann\$	خطا، و نه رسیدن به متن a
ABmCSDE\$	emgnnecaaan\$	خطا، و نه رسیدن به متن c	DE\$	nn\$	خطا، و نه رسیدن به متن D
ABmCSDE\$	mgnnecaaan\$	خطا، و نه رسیدن به متن A	E\$	nn\$	استفاده از 11
BmCSDE\$	mgnnecaaan\$	استفاده از قانون 6	nE\$	nn\$	match n
mCSDE\$	mgnnecaaan\$	match m	E\$	n\$	استفاده از 11
CSDE\$	gnnecann\$	استفاده از 7	nE\$	n\$	match n
gCSDE\$	gnnecann\$	match g	E\$	\$	استفاده از 12
CSDE\$	nnecann\$	خطا، و نه رسیدن به متن	\$	\$	

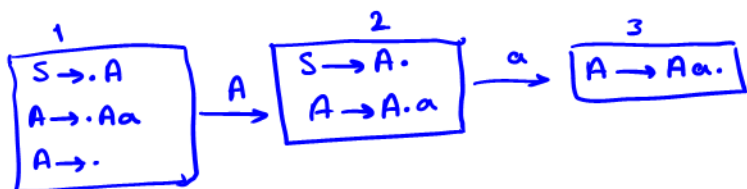
۳- در پارسره‌های بر پایه shift/reduce مزیت داشتن بازگشت از چپ بر داشتن بازگشت از راست چیست؟

در صورتیکه از بازگشت از چپ استفاده شود، حافظه استک کمتر مورد نیاز است. برای مثال دو گرامر زیر که برای زبان a^* نوشته شده اند را در نظر بگیرید:

$$\begin{array}{ll} S \rightarrow A & S \rightarrow A \\ A \rightarrow Aa | \epsilon & A \rightarrow aA | \epsilon \end{array}$$

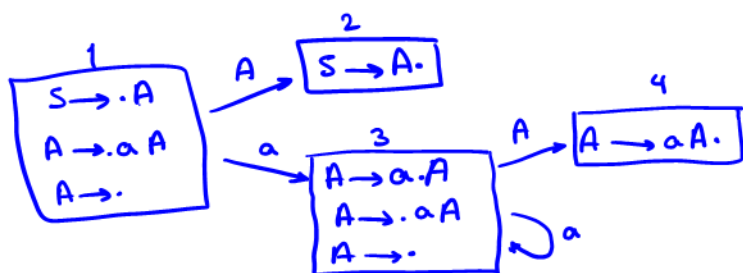
(1) (2)

در ادلی که بازگشت از چپ دارد، حافظه مورد نیاز از درجه $O(1)$ است و در دوم حافظه از درجه $O(n)$ (n تعداد تکرار a) اگر DFA مربوط به این گرامر رسم کنیم:



$$\begin{aligned} follow(S) &= \{ \$ \} \\ follow(A) &= \{ \$, a \} \end{aligned}$$

گرامر ۱:



$$\begin{aligned} follow(S) &= \{ \$ \} \\ follow(A) &= \{ \$ \} \end{aligned}$$

گرامر ۲:

- در گرامر اول، حلقه نداریم و با هر بار دیدن A یک کاسه داریم. بنابراین حافظه استک از درجه $O(1)$ است.
- در گرامر دوم به تعداد a عمل shift انجام می‌شود سپس کاسه انجام می‌شود. بنابراین حافظه استک از درجه $O(n)$ است.

۴- گرامر زیر برای تعریف متغیرهای از نوع صحیح و یا اعشاری داده شده است:

$$\begin{aligned} A' &\rightarrow A \\ A &\rightarrow BD; A \mid \epsilon \\ B &\rightarrow i \mid f \\ D &\rightarrow D, a \mid a \end{aligned}$$

$$\begin{aligned} follow(A') &= \{ \$ \} \\ follow(A) &= \{ \$ \} \\ follow(B) &= \{ a \} \\ follow(D) &= \{ ', , \} \end{aligned}$$

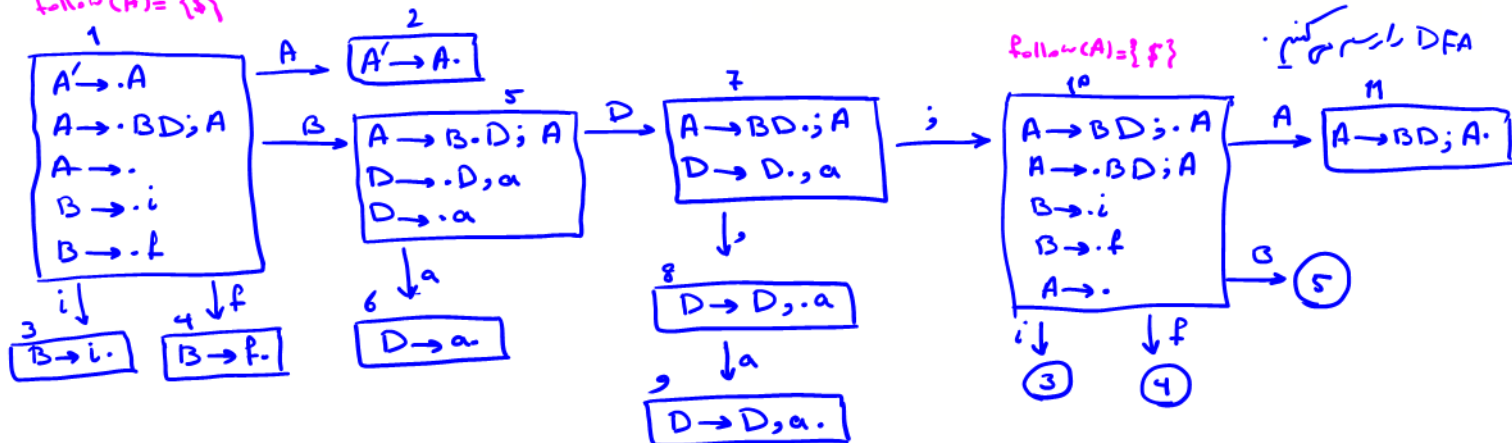
الف) تعیین کنید LL(1) است یا خیر؟

گرامر (۱) نیست، چون بازگشت از چپ دارد.

ب) تعیین کنید این گرامر SLR(1) است یا خیر؟

در صورتی که SLR(1) است، جدول پارس را برای آن تشکیل دهید.

$$follow(A) = \{ \$ \}$$



گرامر SLR(1) است.

	$\$$	α	d	f	i	j	D	B	A
1	r_2			s_4	s_3		5	2	
2	accept								
3		r_3							
4		r_3							
5		s_6					7		
6	r_6		r_6						
7	s_{10}		s_8						
8		s_9							
9	r_5		r_5						
10	s_3	s_4					5	11	
11	r_1								

۵- تعیین کنید هر یک از گرامرهای زیر LR(0), SLR(1), LALR(1) و یا CLR(1) هست یا خیر؟

(الف)

$$S' \rightarrow S$$

$$S \rightarrow ABdD \mid bD$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow b$$

$$D \rightarrow ab \mid \epsilon$$

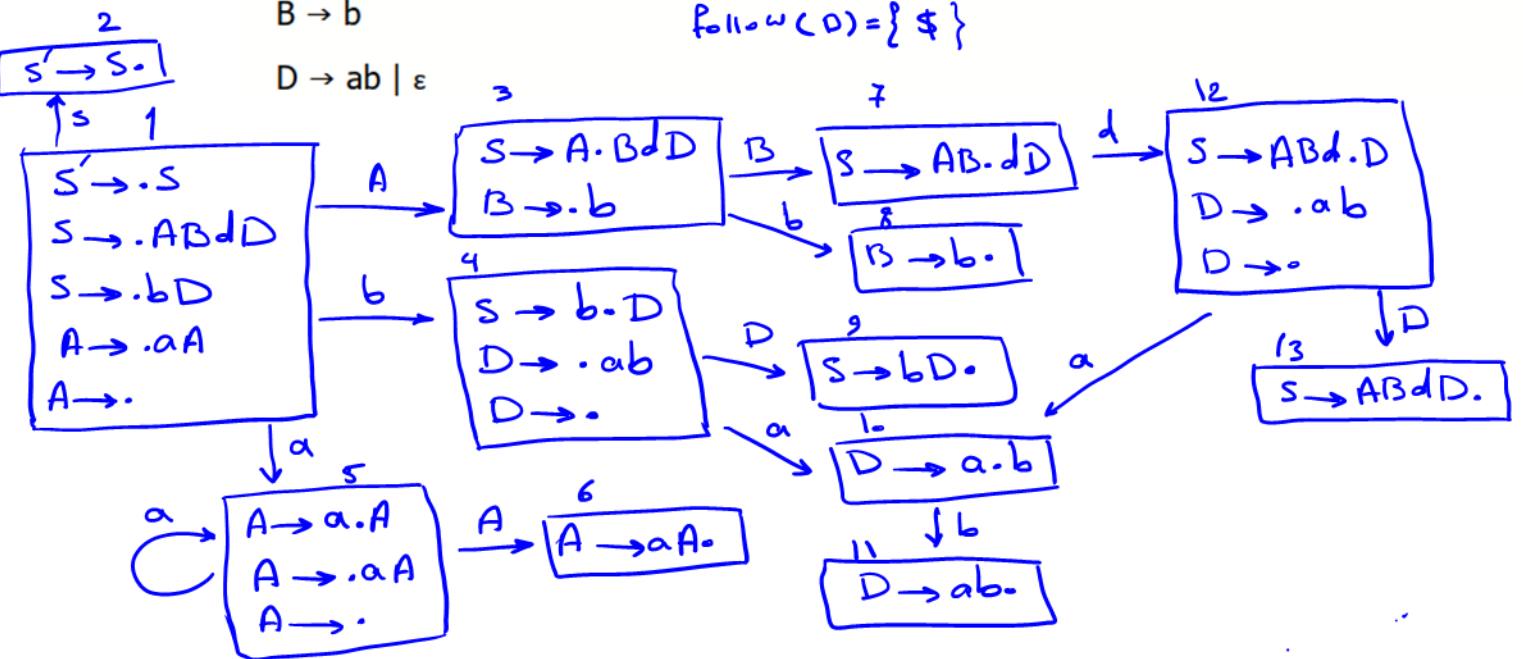
$$\text{follow}(S') = \{\$ \}$$

$$\text{follow}(S) = \{\$ \}$$

$$\text{follow}(A) = \{b\}$$

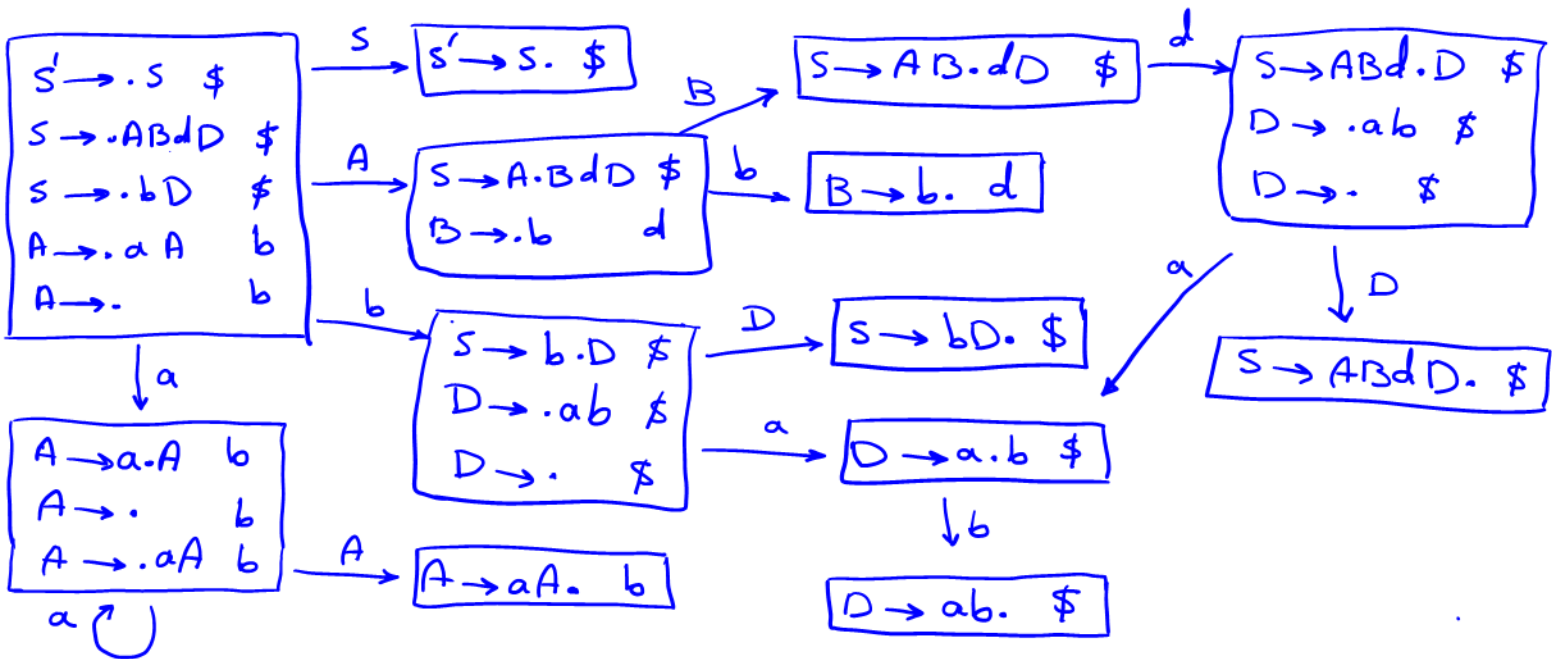
$$\text{follow}(B) = \{d\}$$

$$\text{follow}(D) = \{\$ \}$$



LR(0) نیست زیرا در حالت های 1، 4، 5، 12 هم می توانیم سیگنال انجام دهیم و هم کما هست.

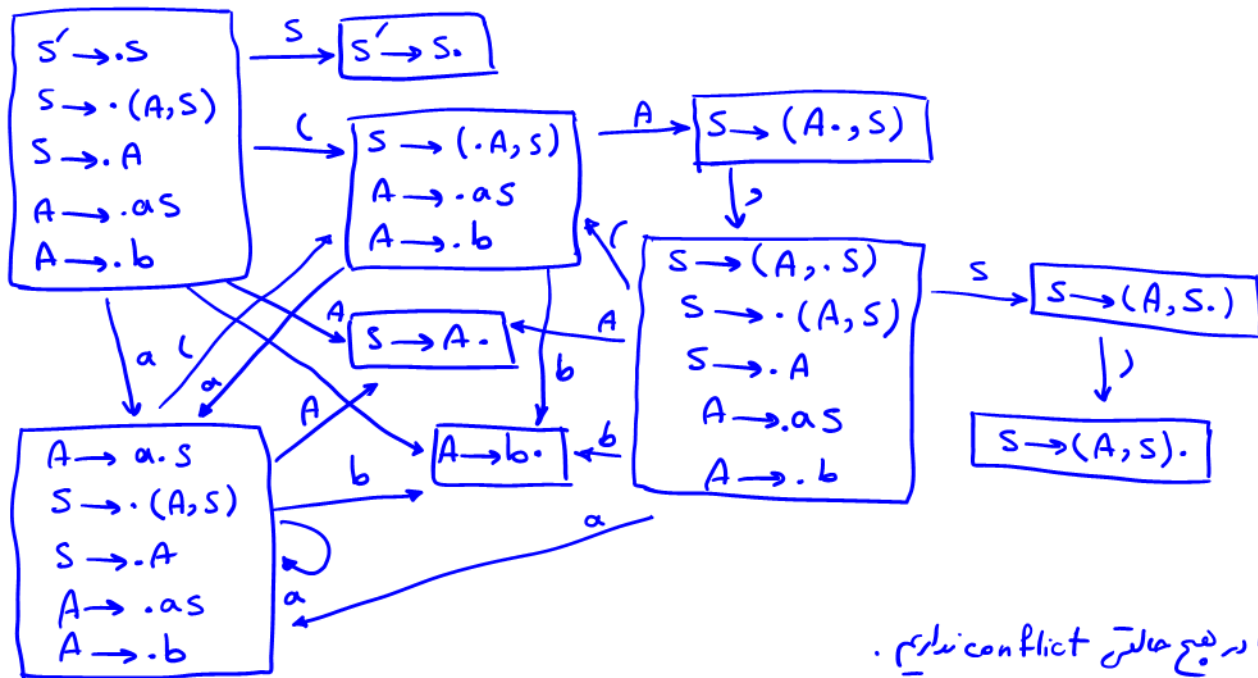
SLR(1) نیست. در حالت 1، در صورتیکه بر ورودی a رسیدن شود هم می توانیم با استفاده از جدول $A \rightarrow \epsilon$ کاهش دهیم و هم سیگنال دهیم.



در حالت ۱، به انداز فردی b ، هم می‌توانیم shift دهیم و هم با استفاده از قانون $A \rightarrow \epsilon$ ، کاهش دهیم. پس این گرامر (LR(0) و ALRC(0)) هم نیست.

$S' \rightarrow S$ $\text{follow}(S') = \{ \$ \}$
 $S \rightarrow (A, S) \mid A$ $\text{follow}(S) = \{ \$, ,, , \}$
 $A \rightarrow aS \mid b$ $\text{follow}(A) = \{ ,, , \$, , \}$

(ب)



- (RC) است. زیرا در هیچ حالتی conflict نداریم.

- چون (RC) است در نتیجه (LR(0), SLR(0), LR(1), ALRC(1), CLRC(1)) نیز هست.

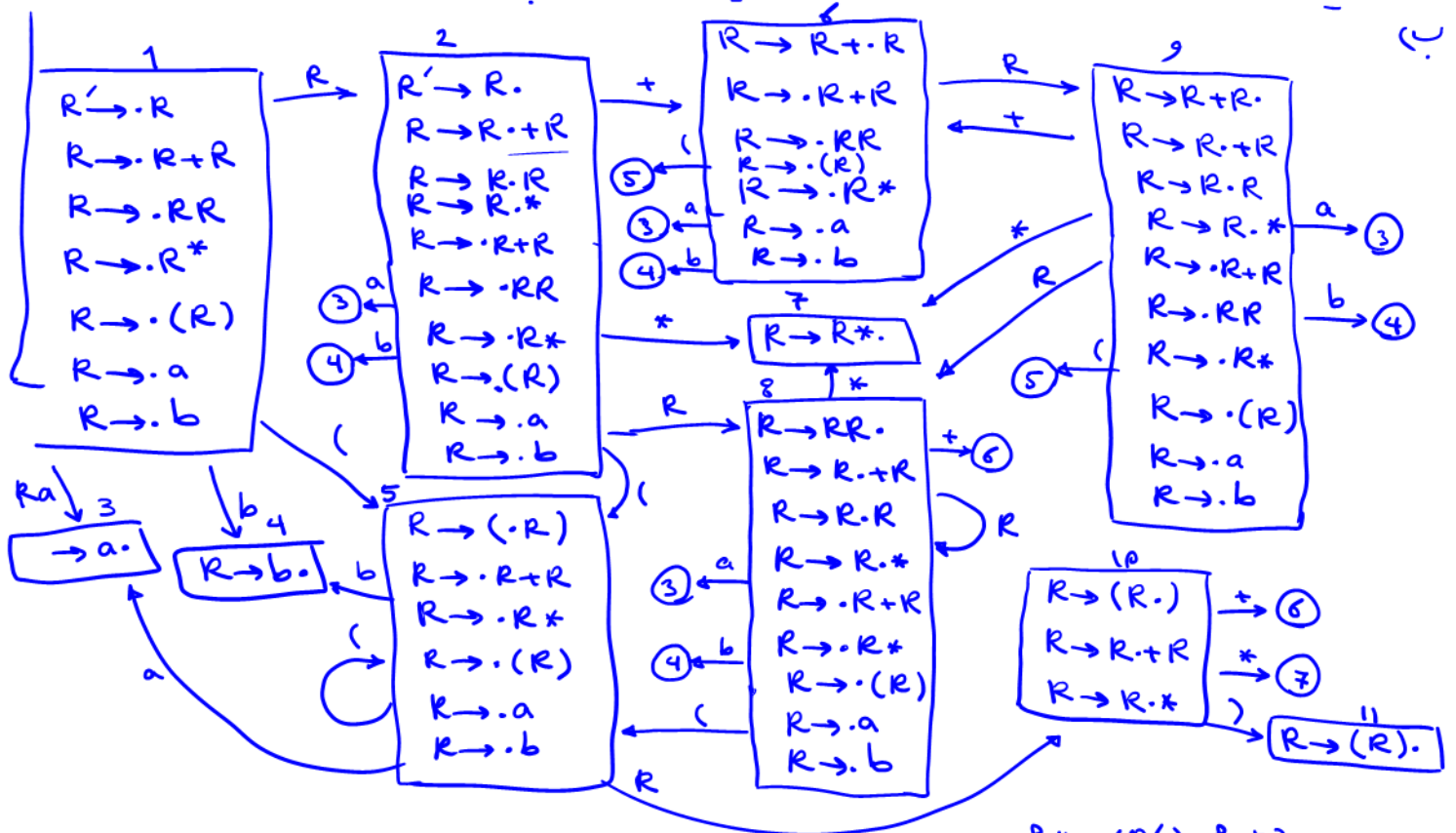
ب) جدول تجزیه (1)SLR گرامر مبهم عبارت های منظم زیر را ساخته و تداخل ها را به گونه ای رفع کنید که بتوان

عبارت های منظم را به صورت صحیح تجزیه کرد.

$$R' \rightarrow R$$

$$R \rightarrow R + R \mid RR \mid R^* \mid (R) \mid a \mid b$$

الف) زیرا اگر ارض زنج ابرام شده دارای توازن شده و پیچیده تر است که در برخی مواقع ¹ فهم آنرا مشکل می کند. و همچنین باعث می شود سرعت اشتقاقی کاهش پیدا کند.



$$\text{Follow}(R') = \{ \$ \}$$

$$\text{Follow}(R) = \{ \$, +, *,), a, b,) \}$$

اولیٰ عملہا

پرائز

سہ

الحق

2.

[illegible]