Marco Túlio Alves de Barros

LISTA_8 - EXERCICIO_1_LETRA_a

a) id := num ; id := id + (id := num + num , id)

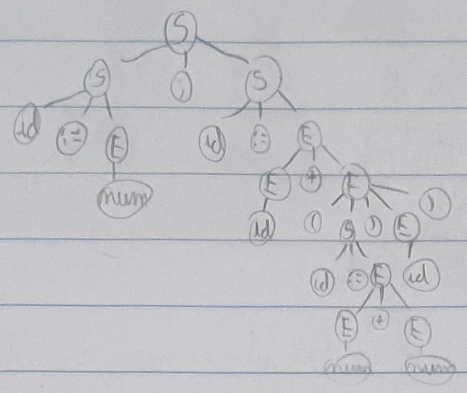
Left-most : S => S ; S => id := E ; S => id := num ; S

=> id := num ; id := E => id := num ; id := E + E

=> id := num ; id := id + E => id := num ; id := id + (S , E)

=> id := num ; id := id + (id := E , E) => id := num ; id := id + (id := E + E , E)

=> id := num ; id := id + (id := num + num , id)



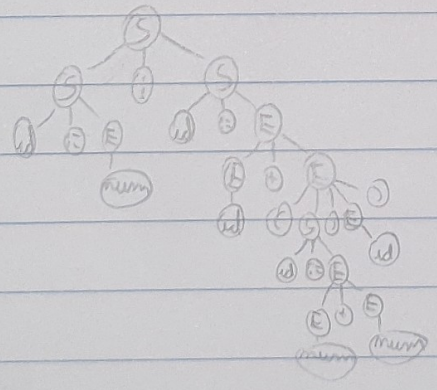
Right-mos : S => S ; S => S ; id := E => S ; id := E + E => S ; id := E + (S , E)

=> S ; id := E + (S , id) => S ; id := E + (id := E , id)

=> S ; id := E + (id := E + E , id) => S ; id := id + (id := num + num id)

=> id := E ; id := id + (id := num + num , id)

=> id := num ; id := id + (id := num + num , id)



As árvores para derivações left most e right most são, de fato, iguais

Marco Túlio Alves de Barros

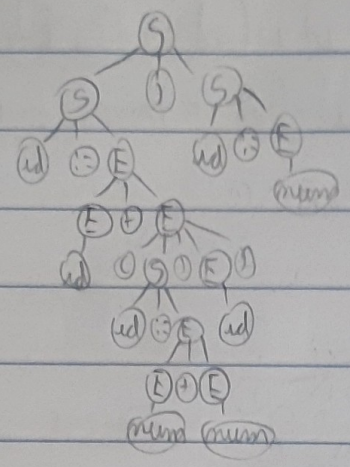
LISTA_8_EXERCICIO_1_LETRA_b

Left most: $S \Rightarrow S;S \Rightarrow id := E;S \Rightarrow id := E+E;S$

$\Rightarrow id := id + (S,E);S \Rightarrow id := id + (id := E, E);S$

$\Rightarrow id := id + (id := E+E, E);S \Rightarrow id := id + (id := num + num, num);S$

$\Rightarrow id := id + (id := num + num, num);\ id := num$

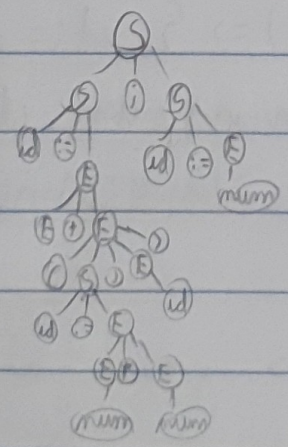


Right most: $S \Rightarrow S;S \Rightarrow S;id := E \Rightarrow S;id := num \Rightarrow id := E;\ id := num$

$\Rightarrow id := E+E;\ id := num \Rightarrow id := E + (S,E);\ id := num$

$\Rightarrow id := E + (id := E, id);\ id := num$

$\Rightarrow id := E + (id := E+E, id);\ id := num \Rightarrow id := id + (id := num + num, id);\ id := num$



Marco Júlio Alves de Barros

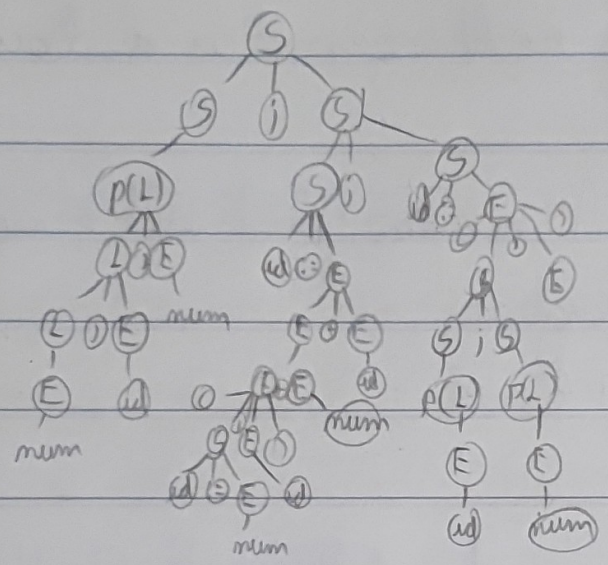
LISTA_8_EXERCICIO_1_LETRA_c

Leftmost $S \Rightarrow S;S \Rightarrow print(L);S \Rightarrow print(L,E);S \Rightarrow print(L,E,E);S$

$\Rightarrow print(num, id, num);S, S \Rightarrow \ldots id := E + E \Rightarrow E+E+E \Rightarrow (S,E)+E+E, S \Rightarrow \ldots id := (S,E);S$

$\Rightarrow \ldots id := (id := num, id) + num + id, S \Rightarrow \ldots id := E$

$\Rightarrow \ldots id := (S,E) \Rightarrow \ldots id := (S,S,E) \Rightarrow \ldots id := (print(L); print(L), E+E$



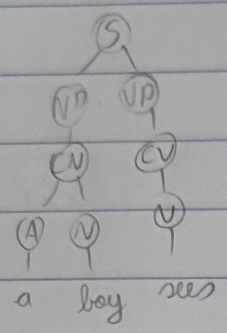
NÃO FUNCIONA

Marco Túlio Alves de Barros

LISTA_8_EXERCICIO_2_LETRA_a

Left: S => NP VP => CN VP => A N VP => a boy CV
=> a boy V => a boy sees

Right: S => NP VP => NP CV => NP V => NP sees
=> CV sees => A N sees => a boy sees

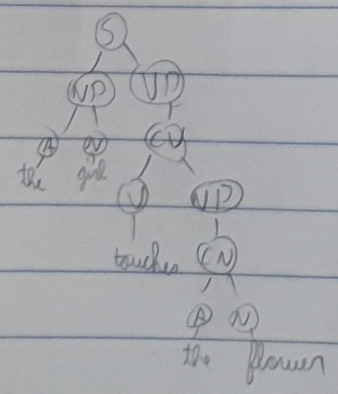


As árvores para left e right most são iguais

LISTA_8_EXERCICIO_2_LETRA_b

Left: S => NP VP => CN VP => A N VP => the girl CV
=> the girl V NP => the girl Touches CN => ... A N
=> the girl touches the flower

Right: S => NP VP => NP CV => NP V NP => NP V CN
=> NP V A N => NP Touches the flower => A N ...
=> the girl touches the flower



As árvores para left e right most são iguais

Marco Túlio Alves de Barros

LISTA_8_EXERCICIO_3_LETRA_a

A cadeia print num = num NÃO é gerado (E não gera num num)

LISTA_8_EXERCICIO_3_LETRA_b

A cadeia "; print num = num end" não é gerado pois o símbolo inicial S não tem produção que começa gerando ;

LISTA_8_EXERCICIO_3_LETRA_c

A cadeia "if num = num then print else end" não é gerado pois a produção é print E (E não gera ε) e S não gera end

LISTA_8_EXERCICIO_3_LETRA_d

A cadeia não é gerado pois ao escrever a derivação

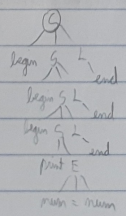
S => begin S L o resultado foi apenas begin end (gerado pela produção L -> end) mas o S simplesmente desapareceu.

LISTA_8_EXERCICIO_3_LETRA_e

S => begin S L => begin begin S L L => begin begin begin S L L L

=> begin begin begin print E L L L

=> begin begin begin print num = num end end end

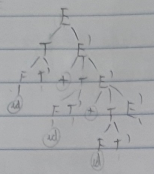


Marco Túlio Alves de Barros

LISTA 8 - EXERCICIO - 4 - LETRA - a

$E \Rightarrow TE' = T + TE' \Rightarrow T + T + TE' \Rightarrow T + T + T \Rightarrow T + T + FT'$

$\Rightarrow T + T + id \Rightarrow T + TF' + id \Rightarrow TF' + id + id \Rightarrow id + id + id$



LISTA - 8 - EXERCICIO - 4 - LETRA - b

Não é possível gerar essa cadeia pois na derivação $T \Rightarrow FT' \Rightarrow F*FT'$ para gerar o *, aparecem dois F que não podem ser consumidos

LISTA - 8 - EXERCICIO - 4 - LETRA - c

Não é possível gerar essa cadeia pois nela o parênteses não foi fechado, e a única produção possível é $F \rightarrow (E)$

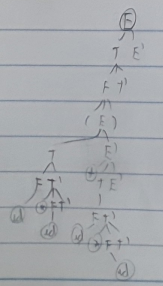
LISTA - 8 - EXERCICIO - 4 - LETRA - d

$E \Rightarrow TE' \Rightarrow T \Rightarrow FT' \Rightarrow F \Rightarrow (E) \Rightarrow (TE') \Rightarrow (T + TE')$

$\Rightarrow (T + T) \Rightarrow (T + FT') \Rightarrow (T + F*FT') \Rightarrow (T + F*F)$

$\Rightarrow (T + id * id) \Rightarrow (FT' + id * id) \Rightarrow (F*FT' + id * id)$

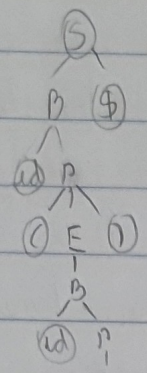
$\Rightarrow (id * id + id * id)$



Marco Túlio Alves de Barros

LISTA_8_EXERCICIO_5_LETRA_a

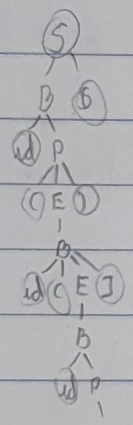
$S \Rightarrow B\$ \Rightarrow id\,P\,\$ \Rightarrow id\,(E)\,\$ \Rightarrow id\,(B)\,\$ \Rightarrow id\,(id\,P)\,\$ \Rightarrow id\,(id)\,\$$



LISTA_8_EXERCICIO_5_LETRA_b

$S \Rightarrow B\$ \Rightarrow id\,P\,\$ \Rightarrow id\,(E)\,\$ \Rightarrow id\,(B)\,\$ \Rightarrow id\,(id\,(E])\,\$ \Rightarrow id\,(id\,(B])\,\$$

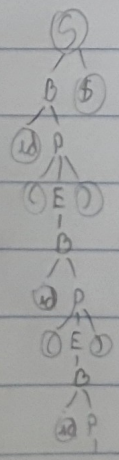
$\Rightarrow id\,(id\,(id\,P])\,\$ \Rightarrow id\,(id\,(id])\,\$$



LISTA_8_EXERCICIO_5_LETRA_c

$S \Rightarrow B\$ \Rightarrow id\,P\,\$ \Rightarrow id\,(E)\,\$ \Rightarrow id\,(B)\,\$ \Rightarrow id\,(id\,P)\,\$ \Rightarrow id\,(id\,(E))\,\$$

$\Rightarrow id\,(id\,(B))\,\$ \Rightarrow id\,(id\,(id\,P))\,\$ \Rightarrow id\,(id\,(id))\,\$$



Marco Túlio Alves de Barros

LISTA_8_EXERCICIO_5_LETRA_d

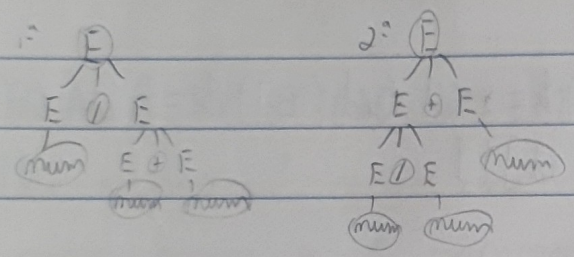
$S \Rightarrow B\$ \Rightarrow id(E]\$ \Rightarrow id(E,B]\$ \Rightarrow id(id,id]\$$ ou $id(id]\$$,

Não é possível gerar essa cadeia pois a linguagem não aceita os terminais "id" juntos, deve haver um outro terminal entre eles

LISTA_8_EXERCICIO_6_LETRA_a

1º $E \Rightarrow E/E \Rightarrow num/E \Rightarrow num/E+E \Rightarrow num/num+num$

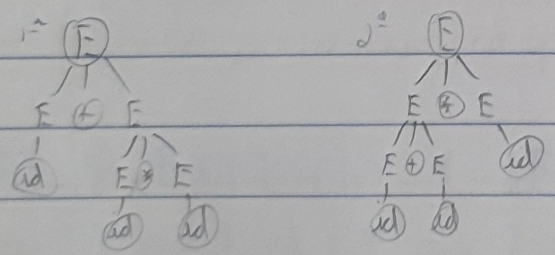
2º $E \Rightarrow E+E \Rightarrow E/E+E \Rightarrow num/num+num$



LISTA_8_EXERCICIO_6_LETRA_b

1º $E \Rightarrow E+E \Rightarrow id+E \Rightarrow id+E*E \Rightarrow id+id*id$

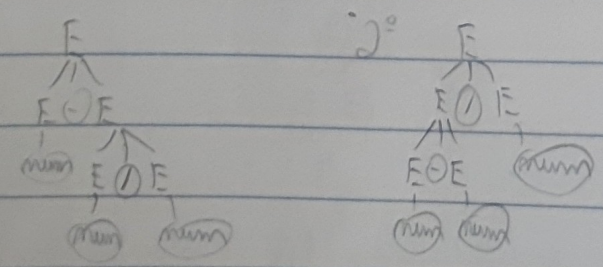
2º $E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow id+id*id$



LISTA_8_EXERCICIO_6_LETRA_c

1º $E \Rightarrow E-E \Rightarrow num-E \Rightarrow num-E/E \Rightarrow num-num/num$

2º $E \Rightarrow E/E \Rightarrow E-E/E \Rightarrow num-num/num$

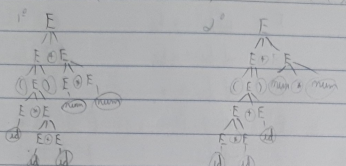


Marco Túlio Alves de Barros

LISTA_8_EXERCICIO_6_LETRA_d

1º $E \Rightarrow E + E \Rightarrow (E) + E \Rightarrow (E * E) + E \Rightarrow (E + E + E) + E$
$\Rightarrow (id * id + id) + E * E \Rightarrow (id * id + id) + num * num$

2º $E \Rightarrow E + E \Rightarrow (E) + E \Rightarrow (E + E) + E \Rightarrow (E + E + E) + E$
$\Rightarrow (id * id + id) + E * E \Rightarrow (id * id + id) + num * num$



LISTA_10_EXERCICIO_1

	NULLABLE	START	FOLLOW	(	a	e	i	b
S	NO	(, a	e	S→(E)SS'	S→a			
S'	YES	e				S'→eS		
E	NO	b	)					E→b

Então a gramática é LL(1), não é ambígua, pois não há duas entradas para uma mesma linha e coluna

LISTA_10_EXERCICIO_2

	NULLABLE	START	FOLLOW	$	+	*	(	)	id
S	N	(, id					S→E$		S→E$
E	N	(, id	$,)				E→TE'		E→TE'
E'	S	+	$,)	E'→	E'→+TE'			E'→	
T	N	(, id	+, $,)				T→FT'		T→FT'
T'	S	*	+, $,)	T'→	T'→	T'→*FT'		T'→	
F	N	(, id	*, +,), $				F→(E)		F→id

Marco Túlio Alves de Barros

LISTA_10_EXERCICIO_2_LETRA_A

PILHA	CADEIA	Linha	Coluna	Reconhecido
S	id + id * id $	S	id	
E $	id + id * id $	E	id	
TE' $	id + id * id $	T	id	
FT'E' $	id + id * id $	F	id	
id T'E' $	id + id * id $	-	-	id
T'E' $	+ id * id $	T'	+	
E' $	+ id * id $	E'	+	
+TE' $	+ id * id $	-	-	+
TE' $	id * id $	T	id	
FT'E' $	id * id $	F	id	
id T'E' $	id * id $	-	-	id
T'E' $	* id $	T'	*	
*FT'E' $	* id $	-	-	*
FT'E' $	id $	F	id	
id T'E' $	id $	-	-	id
T'E' $	$	T'	$	
$	$	-	-	$

∴ A cadeia foi aceita -> id + id * id $ ✓

Marco Túlio Alves de Barros

LISTA_10_EXERCICIO_2_LETRA_B

Pilha	Cadeia	Linha / Coluna	Reconhecido
S	id * + id $	S id	
E$	id * + id $	E id	
TE'$	id * + id $	T id	
FT'E'$	id * + id $	F id	
idT'E'$	id * + id $	- -	id
T'E'$	* + id $	T' *	
*FT'E'$	* + id $	- -	*
FT'E'$	+ id $	F +	

∴ Não é aceita id * + id $ X

LISTA_10_EXERCICIO_2_LETRA_C

Pilha	Cadeia	Linha / Coluna	Reconhecido
S	id * id ($	S id	
E$	id * id ($	E id	
TE'$	id * id ($	T id	
FT'E'$	id * id ($	F id	
idT'E'$	id * id ($	- -	id
T'E'$	* id ($	T' *	
*FT'E'$	* id ($	- -	*
FT'E'$	id ($	F id	
idT'E'$	id ($	- -	id
T'E'$	($	T' (	

∴ Não é aceita id * id ($ X

Marco Túlio Alves de Barros

LISTA - 10 - EXERCICIO - 2 - LETRA - D

Pilha	Cadeia	Linha	Coluna	Reconhecido
S	(id+id+id*id)$	S	(	
E$	(id*id+id*id)$	E	(	
TE'$	(id*id+id*id)$	T	(	
FT'E'$	(id+id+id*id)$	F	(	
(E)T'E'$	(id*id+id+id)$	-	-	(
E)T'E'$	id*id+id*id)$	E	id	
TE')T'E'$	id*id+id+id)$	T	id	
FT'E')T'E'$	id*id+id+id)$	F	id	
idT'E')T'E'$	id*id+id+id)$	-	-	id
T'E')T'E'$	*id+id+id)$	T'	*	
*FT'E')T'E'$	*id+id+id)$	-	-	*
FT'E')T'E'$	id+id*id)$	F	id	
idT'E')T'E'$	id+id*id)$	-	-	id
T'E')T'E'$	+id*id)$	T'	+	
E')T'E'$	+id*id)$	E'	+	
+TE')T'E'$	+id*id)$			+
TE')T'E'$	id*id)$	T	id	
FT'E')T'E'$	id*id)$	F	id	
idT'E')T'E'$	id*id)$			id
T'E')T'E'$	*id)$	T'	*	
*FT'E')T'E'$	*id)$			*
FT'E')T'E'$	id)$	F	id	
idT'E')T'E'$	id)$			id
T'E')T'E'$	)$	T', E'	)	
)T'E'$	)$			)
T'E'$	$	T', E'	$	$

∴ A cadeia foi reconhecida => (id+id+id*id)$

Marca Lúcia Alves de Barros

LISTA_10_EXERCICIO_3

	NULLABLE	FIRST	FOLLOW
S'	NO	{ \WORD, end, begin, WORD, \begin {WORD's, $	
S	YES	{ \WORD, end, begin, WORD, \begin {WORD's	$, }, \end {WORD's
B	NO	\begin {WORD's	{ \WORD, end, begin, WORD, \begin {WORD's, \end {WORD's
E	NO	\end {WORD}	
X	NO	{ \WORD, end, begin, WORD, \begin {WORD's	{ \WORD, end, begin, WORD, \begin {WORD's

	\begin {WORD's	\end {WORD's	\WORD	WORD	begin	end	{	}	$
S' ->	S$		S$	S$	S$	S$	S$		S$
S ->	XS	S ->	XS	XS	XS	XS	XS	S ->	S ->
B ->	\begin {WORD's								
E ->		\end {WORD's							
X ->	BSE		\WORD	WORD	begin	end	{S's		

LISTA_11_EXERCICIO_1_LETRA_A

n expr -> n term E n term -> n factor T n factor -> n primary F

E -> + n term E T -> n factor T F -> * F

E -> T -> F ->

Gramática Resultante: S -> n expr $

n expr -> n term E n factor -> n primary F

E -> + n term E F -> * F

E -> F ->

n term -> n factor T n primary -> a

T -> n factor T n primary -> b

T ->

Marco Túlio Alves de Barros

LISTA - 11 - EXERCICIO - 1 - LETRA - B

	NULL	First	Follow
S	N	a b	
nexpr	N	a b	$
E	Y	+	$
nterm	N	a b	+ $
T	Y	a b	+ $
nfactor	N	a b	a b + $
F	Y	*	a b + $
nprimary	N	a b	* a b + $

	$	a	b	+	*
S		→ nexpr $	→ nexpr $		
nexpr		→ nterm E	→ nterm E		
E →				→ + nterm E	
nterm		→ nfactor T	→ nfactor T		
T →		→ nfactor T	→ nfactor T	T →	
nfactor		→ nprimary F	→ nprimary F		
F →		F →	F →	F →	→ * F
nprimary		→ a	→ b		

∴ A gramática é LL(1) como não há multiplas entradas numa única célula

LISTA - 11 - EXERCICIO - 2 - LETRA - A

	NULL	First	Follow
S	N	u	
B	N	w	x, y, z, v
D	Y	y, x	z
E	Y	y	x, z
F	Y	x	z

LISTA - 11 - EXERCICIO - 2 - LETRA - B

	u	v	w	x	y	z
S →	uBDZ					
B →			Bv u			
D →				EF	EF	EF
E →				λ	y	λ
F →				x		λ

Marco Túlio Alves de Barros

LISTA_11_EXERCICIO_2_LETRA_C

Observe que há duas entradas (duas produções diferentes) para uma mesma linha e coluna, isso prova que a gramática é ambígua. É devido a recursividade a esquerda em $B \to B\,v$

LISTA_11_EXERCICIO_2_LETRA_D

Removendo a recursividade a esquerda

$B \to u\,B'$
$B' \to v\,B'$
$B' \to$

$\therefore$

$S \to u\,B\,D\,z$
$B \to u\,B'$
$B' \to v\,B'$
$B' \to$

$D \to E\,F$
$E \to y$
$E \to$

$F \to x$
$F \to$

	Null	First	Follow
S	N	u	
B	N	u	x y z
B'	y	v	x y z
D	y	x, y	z
E	y	y	x z
F	y	x	z

u	v	u	x	y	z
uBDz					
		uB'			
	vB'		B' →	B' →	B' →
			EF	EF	EF
			E →	y	E →
			x		F →

Após transformada, a gramática aceita a mesma linguagem e é LL(1)

Marco Túlio Alves de Barros

LISTA - II - EXERCICIO - 3

Gramática Resultante:

$A \to S\ \$$

$S \to id := E\ S'$

$S \to print\ (L)\ S'$

$S' \to ;\ S\ S'$

$S' \to$

$E \to id\ E'$

$E \to num\ E'$

$E \to (S,E)\ E'$

$E' \to +\ E\ E'$

$E' \to$

$L \to E\ L'$

$L' \to ,\ E\ L'$

$L' \to$

	NULL	First	Follow
A	N	print id	
S	N	print id	$, ;
S'	Y	;	$,
E	N	id num (	;) + , $
E'	Y	+	$,
L	N	id num (	)
L'	Y	,	)

	$	id	:=	print	(	)	;	num	,	+
A →		S$		S$						
S →		id := E S'		print (L) S'						
S' →	λ						; S S'		λ	
E →		id E'			(S,E) E'			num E'		
E' →	λ								λ	+ E E'
L →		E L'			E L'			E L'		
L' →						λ			, E L'	

∴ A linguagem resultante é LL(1) e apresenta as tabelas apresentadas acima. Veja que não apresenta múltiplas entradas por célula

Marco Túlio Alves de Barros

LISTA_11_EXERCICIO_4

$E \to E+T$	$F \to O$		$S \to E\$$	
$E \to T$	$O \to (E)$	$\Rightarrow$	$E \to TE'$	$F \to OY$
$T \to F*T$	$O \to num$		$E' \to +TE'$	$Y \to ++$
$T \to F$	$O \to id$		$E' \to$	$Y \to$
$F \to O++$			$T \to FX$	$O \to (E)$
			$X \to *T$	$O \to num$
			$X \to$	$O \to id$

	NULL	First	
S	N	id num (	
E	N	id num (	\$,)
E'	Y	+	\$)
T	N	id num (	+ $)
X	Y	*	+ $)
F	N	id num (	* + $)
Y	Y	++	* + $)
O	N	id num (	++ * + $)

	id	num	++	*	+	(	)	\$
$S \to$	$E\$$	$E\$$				$E\$$		
$E \to$	TE'	TE'				TE'		
$E' \to$					$+TE'$		λ	λ
$T \to$	FX	FX				FX		
$X \to$				$*T$	λ		λ	λ
$F \to$	OY	OY				OY		
$Y \to$			$++$	λ	λ		λ	λ
$O \to$	id	num				(E)		

Marco Túlio Alves de Barros

LISTA_11_EXERCICIO_4_LETRA_A

Pilha	Cadeia	L I C	Reconheceu
S	num + id ++ $	S num	
E$	"	E num	
TE'$	"	T num	
FXE'$	"	F num	
OYXE'$	"	O num	num
YXE'$	+ id ++ $	Y +	λ
XE'$	+ id ++ $	X +	λ
E'$	+ id ++ $	E' +	
+TE'$	"	T +	+
TE'$	id ++ $	T id	
FXE'$	"	F id	
OYXE'$	"	O id	id
YXE'$	++ $	Y ++	++
XE'$	$	X E' $	λ
$	$		$

∴ A cadeia foi reconhecida

Marco Túlio Alves de Barros

LISTA-11-EXERCICIO-4-LETRA-B

Pilha	Cadeia	L/C	Reconhece
S	num++*num $	S num	
E$	"	E num	
TE'$	"	T num	
FXE'$	"	F num	
OYXE'$	"	O num	num
YXE'$	++*num $	Y ++	++
XE'$	*num $	X *	*
TE'$	num $	T num	
FXE'$	"	F num	
OYXE'$	"	O num	num
YXE'$	$	Y $	λ
XE'$	$	X $	λ
E'$	$	E' $	λ
$	$		$

∴ A cadeia foi reconhecida

LISTA-11-EXERCICIO-4-LETRA-C

Pilha	Cadeia	L/C	Reconhece
⋮	⋮	⋮	⋮
*TE'$	*id+num$		*
TE'$	id+num$	T id	
FXE'$	"	F id	
OYXE'$	"	O id	id
YXE'$	+num$	Y,X +	λ
+TE'$	+num$	T num	+
OYXE'$	num$	O num	num
$	$		$

∴ A cadeia foi reconhecida

Marco Túlio Alves de Barros

LISTA_11_EXERCICIO_4_LETRA_D

Pilha	Cadeia	LIC	Reconhecer
⋮	⋮	⋮	⋮
*TE'$	*(id+num)++$	T (	*
FXE'$	(id+num)++$	F (	
OYXE'$	(id+num)++$	O (	(
E)YXE'$	id+num)++$	E id	
TE')YXE'$	"	T id	
FXE')YXE'$	"	F id	
OYXE')YXE'$	"	O id	id
YXE')YXE'$	+num)++$	E' +	
+TE')YXE'$	+num)++$	T num	+
FXE')YXE'$	num)++$	F num	
OYXE')YXE'$	"	O num	num
YXE')YXE'$	)++$	X, Y, E)	λ
)YXE'$	)++$	Y ++	)
++XE'$	++$	X, E' $	λ
$	$		$

∴ A cadeia foi reconhecida

Marco Túlio Alves de Barros

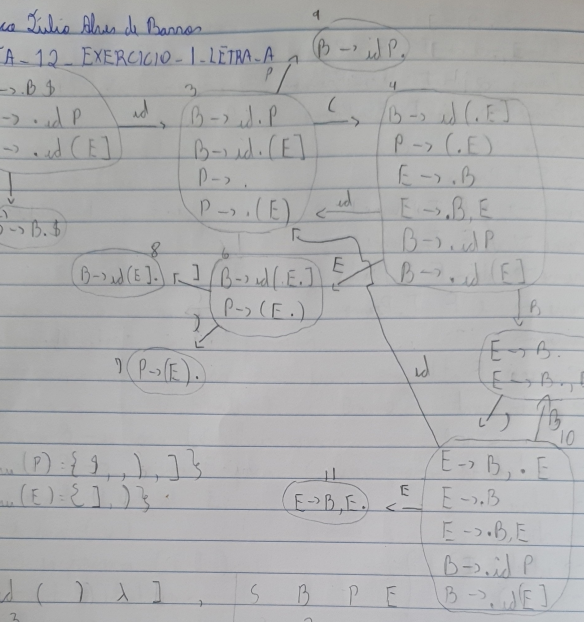
LISTA - 11 - EXERCICIO - 4 - LETRA - E

Pilha	Cadeia	L/C		Reconhecer
S	id * (id ++ num) $	S	id	
E$	"	E	id	
TE'$	"	T	id	
FXE'$	"	F	id	
OYXE'$	"	O	id	id
YXE'$	* (id ++ num) $	X	*	
*TE'$	"	T	(	*
FXE'$	(id ++ num) $	F	(	
OYXE'$	"	O	(	(
E)YXE'$	id ++ num) $	E	id	
TE')YXE'$	"	T	id	
FXE')YXE'$	"	F	id	
OYXE')YXE'$	"	O	id	id
YXE')YXE'$	++ num) $	Y	++	++
XE')YXE'$	num) $	—		

// X não lê num então a cadeia foi recusada

Marco Julio Alves de Barros

LISTA - 12 - EXERCICIO - 1 - LETRA - A



Follow (P) = { $, ,) ,] }

Follow (E) = {] ,) }

	$	id	(	)	λ	]	,	S	B	P	E
1		s3							g2		
2	a										
3	r3	s4	r3	r3	r3	r3					
4		s3							g5	g9	g6
5				r5		r5	r10				
6				s7		s8					
7	r4	r4	r4	r4	r4	r4	r4				
8	r2	r2	r2	r2	r2	r2	r2				
9	r1	r1	r1	r1	r1	r1	r1				
10		s3							g10	g11	
11	r6	r6	r6	r6	r6	r6	r6				

∴ É SLR mas não LR(0)

Marco Túlio Alves de Barros

LISTA_12_EXERCICIO_1_LETRA_B

$S \Rightarrow B\$ \Rightarrow id\,P \Rightarrow id(E) \Rightarrow id(B) \Rightarrow id(id) \Rightarrow id(id)\$$

∴ cadeia é aceita

LISTA_12_EXERCICIO_1_LETRA_C

$S \Rightarrow B\$ \Rightarrow id\,P \Rightarrow id(E) \Rightarrow id(B) \Rightarrow id(id(E)) \Rightarrow id(id(id))\$$

∴ cadeia é aceita

LISTA_12_EXERCICIO_1_LETRA_D

$S \Rightarrow B\$ \Rightarrow id\,P \Rightarrow id(E) \Rightarrow id(B) \Rightarrow id(id(E)) \Rightarrow id(id(id))\$$

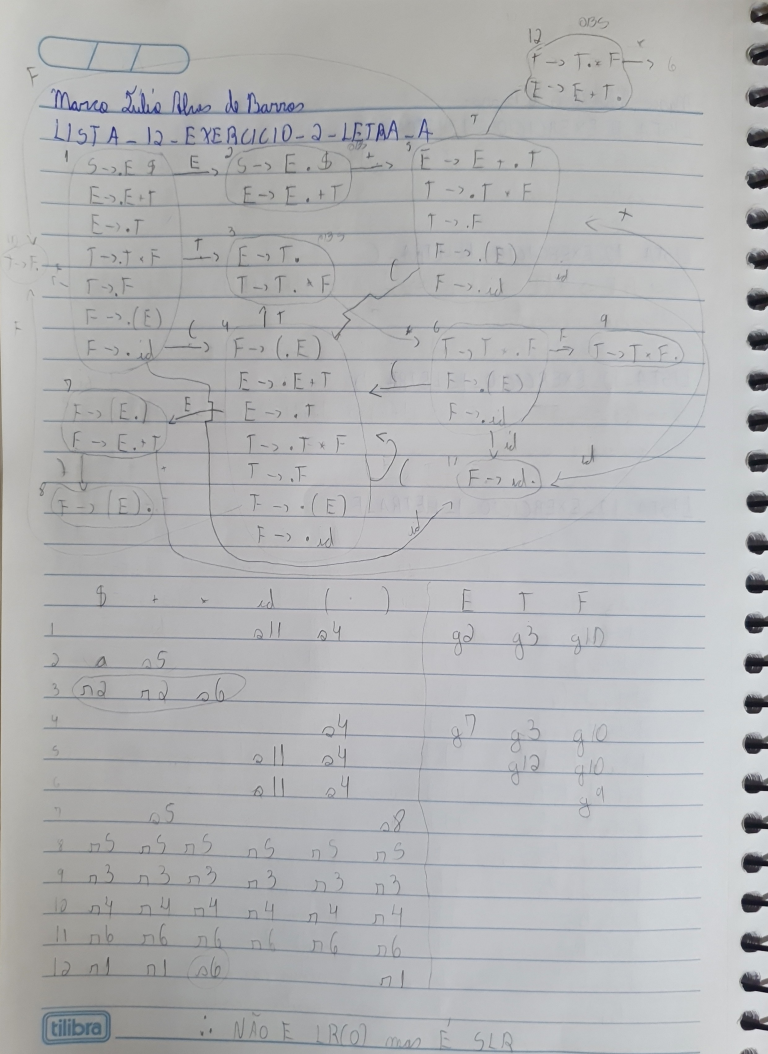
∴ cadeia é aceita

LISTA_12_EXERCICIO_1_LETRA_E

Pilha	Cadeia	Ação
1	$id(id(id))\$$	shift 3
id_3	$(id(id))\$$	shift 4
$id_3(_4$	$id(id))\$$	shift 3
$id_3(_4id_3$	$(id))\$$	shift 4
$id_3(_4id_3(_4$	$id))\$$	shift 3
$id_3(_4id_3(_4id_3$	$))\$$	reduz por 3
$id_3(_4id_3(_4id_3P_9$	$))\$$	reduz por 1
$id_3(_4id_3(_4B_5$	$))\$$	reduz por 5
$id_3(_4id_3(_4E_6$	$))\$$	shift 7
$id_3(_4id_3(_4E_6)_7$	$)\$$	reduz por 4
$id_3(_4id_3P_9$	$)\$$	reduz por 1
$id_3(_4B_5$	$)\$$	reduz por 5
$id_3(_4E_6$	$)\$$	shift 8
$id_3(_4E_6)_8$	$\$$	reduz por 2
B_1	$\$$	aceita a cadeia

Marco Túlio Alves de Barros

LISTA - 12 - EXERCÍCIO - 2 - LETRA - A



	$	+	*	id	(	)	E	T	F
1				s11	s4		g2	g3	g10
2	a	s5							
3	r2	r2	s6						
4					s4		g7	g3	g10
5				s11	s4			g12	g10
6				s11	s4				g9
7		s5				s8			
8	r5	r5	r5	r5	r5	r5			
9	r3	r3	r3	r3	r3	r3			
10	r4	r4	r4	r4	r4	r4			
11	r6	r6	r6	r6	r6	r6			
12	r1	r1	s6			r1			

∴ NÃO É LR(0) mas É SLR

Marco Túlio Alves de Barros

LISTA_12_EXERCICIO_2_LETRA_B

Pilha	cadeia	action
1	id + id * id $	shift 11
1 id11	+ id * id $	reduz por 6
1 F_{10}	+ id * id $	reduz por 4
1 T_3	+ id * id $	reduz por 2
1 E_2	+ id * id $	shift 5
E_2 +5	id * id $	shift 11
E_2 +5 id11	* id $	reduz por 6
E_2 +5 F_{10}	* id $	reduz por 4
E_2 +5 T_{12}	* id $	shift 6
E_2 +5 T_{12} *6	id $	shift 11
E_2 +5 T_{12} *6 id11	$	reduz por 6
E_2 +5 T_{12} *6 F_9	$	reduz por 3
E_2 +5 T_{12}	$	reduz por 1
E_1	$	∴ aceita a cadeia

LISTA_12_EXERCICIO_2_LETRA_C

Pilha	Cadeia	action
1	id * + id $	shift 11
1 id11	* + id $	reduz por 6
1 F_{10}	* + id $	reduz por 4
1 T_3	* + id $	shift 6
1 T_3 *6	+ id $	Recusada

LISTA_12_EXERCICIO_2_LETRA_D

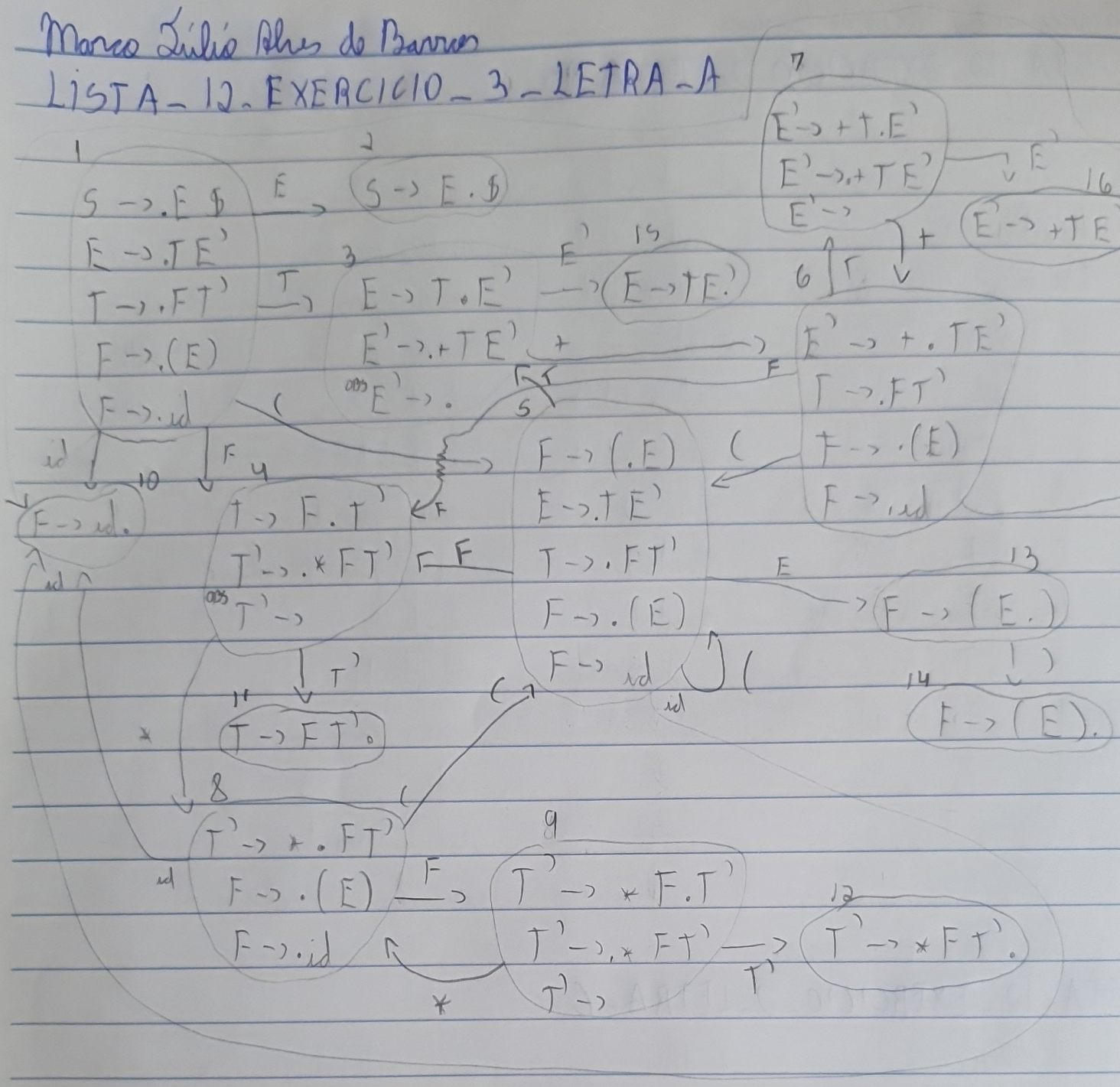
CADEIA RECUSADA

LISTA_12_EXERCICIO_2_LETRA_E

Cadeia Aceita T => (E) => (E+T) => (E+T*F) => (T+F+T*F)

Marco Túlio Alves de Barros

LISTA_12_EXERCICIO_3_LETRA_A



Não vai ser LR(0) pois há conflito de shift e redução nos casos de produções que aceitam cadeia vazia

Follow(E') = $,)

Follow(T') = $,), +

Marco Túlio Alves de Barros

	$	+	*	id	(	)	x	E	E'	T	T'	F
1				s10	s5			g2		g3		g4
2	A											
3	r3	s6				r3			g15			
4	r6	r6	s8			r6					g11	
5				s10	s5			g13		g3		g4
6				s10	s5					g7		g4
7	r3	s6				r3			g16			
8				s10	s5							g9
9	r6	r6	s8			r6					g12	
10	r8	r8	r8	r8	r8	r8	r8					
11	r4	r4	r4	r4	r4	r4	r4					
12	r5	r5	r5	r5	r5	r5	r5					
13						s14						
14	r7	r7	r7	r7	r7	r7	r7					
15	r1	r1	r1	r1	r1	r1	r1					
16	r2	r2	r2	r2	r2	r2	r2					

LISTA_12_EXERCICIO_3_LETRA_B

Cadeia aceita

LISTA_12_EXERCICIO_3_LETRA_C

Cadeia recusada

LISTA_12_EXERCICIO_3_LETRA_D

Cadeia recusada

Marco Túlio Alves de Barros

LISTA - 10 - EXERCICIO - 3 - LETRA - E

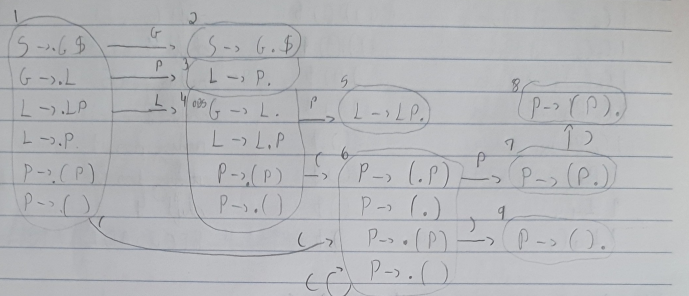
Pilha	Cadeia	action
$1(_5$	$(id * id + id * id)\$$	shift 5
$1(_5 id_{10}$	$* id + id * id)\$$	reduz por 8
$1(_5 F_4$	$* id + id * id)\$$	shift 8
$1(_5 F_4 *_8$	$id + id + id)\$$	shift 10
$1(_5 F_4 *_8 id_{10}$	$+ id * id)\$$	reduz por 8
$1(_5 F_4 *_8 F_9$	$+ id * id)\$$	reduz por 6
$1(_5 F_4 *_8 F_9 T'_{12}$	$+ id * id)\$$	reduz por 5
$1(_5 F_4 T'_{11}$	$+ id * id)\$$	reduz por 4
$1(_5 T_3$	$+ id * id)\$$	shift 6
$1(_5 T_3 +_6$	$id * id)\$$	shift 10
$1(_5 T_3 +_6 id_{10}$	$* id)\$$	reduz por 8
$1(_5 T_3 +_6 F_4$	$* id)\$$	shift 8
$1(_5 T_3 +_6 F_4 *_8$	$id)\$$	shift 10
$1(_5 T_3 +_6 F_4 *_8 id_{10}$	$)\$$	reduz por 8
$1(_5 T_3 +_6 F_4 *_8 F_9$	$)\$$	reduz por 6
$1(_5 T_3 +_6 F_4 *_8 F_9 T'_{12}$	$)\$$	reduz por 5
$1(_5 T_3 +_6 F_4 T'_{11}$	$)\$$	reduz por 4
$1(_5 T_3 +_6 T_7$	$)\$$	reduz por 3
$1(_5 T_3 +_6 T_7 E'_{16}$	$)\$$	reduz por 2
$1(_5 T_3 E'_{15}$	$)\$$	reduz por 1
$1(_5 E_{13}$	$)\$$	shift 14
$1(_5 E_{13})_{14}$	$\$$	reduz por 7
$1 F_4 T'_{11}$	$\$$	reduz por 6 e 4
$1 T_3 E'_{15}$	$\$$	reduz por 3 e 1
$1 E_2$	$\$$	ACEITA

Marcos Túlio Alves de Barros

LISTA_10_EXERCICIO_4_LETRA_A

0. S -> G $ 2. L -> LP 4. P -> (P)

1. G -> L 3. L -> P 5. P -> ()



	$	(	)	G	L	P
1		s6		g2	g4	g3
2	A					
3	r3	r3	r3			
4	r1	s4				g5
5	r2	r2	r2			
6		s6	s9			g7
7			s8			
8	r4	r4	r4			
9	r5	r5	r5			

Follow (G) = $

A gramática não é LR(0) mas é SLR, pois no estado 4 é possível reduzir e fazer shift. Mas após o reduz ser aplicado apenas no conjunto follow de G o problema é resolvido e fica SLR

Marco Túlio Alves de Barros

LISTA_12_EXERCICIO_4_LETRA_B

Pilha	Cadeia	Action
1	(())()$	shift 6
$1C_6$	())()$	shift 6
$1C_6C_6$	))()$	shift 9
$1C_6(_6)_9$	)()$	reduz por 5
$1C_6P_7$	)()$	shift 8
$1(_6P_7)_8$	()$	reduz por 4
$1P_3$	()$	reduz por 3
$1L_4$	()$	shift 4
$1L_4C_4$	)$	RECUSADA ?

LISTA_1_EXERCICIO_4_LETRA_C

Pilha	Cadeia	Action
1	()())$	shift 6
$1C_6$	)())$	shift 9
$1(_6)_9$	())$	reduz por 5
$1P_3$	())$	reduz por 3
$1L_4$	())$	shift 4
$1L_4C_4$	))$	RECUSADA ?

Marco Túlio Alves de Barros

LISTA_12_EXERCICIO_4_LETRA_D

Pilha	Cadeia	Action
1	((((()$	shift 6
$1(_6$	(((()$	shift 6
$1(_6(_6$	((()$	shift 6
$1(_6(_6(_6$	(()$	shift 6
$1(_6(_6(_6(_6$	())$	shift 9
$1(_6(_6(_6(_6)_9$	)$	reduz por 5
$1(_6(_6(_6P_7$	)$	RECUSADA

LISTA_12_EXERCICIO_4_LETRA_E

Pilha	Cadeia	Action
1	(((()())()())$	shift 6 3 vezes
$1(_6(_6(_6$	()())()())$	shift 9
$1(_6(_6(_6)_9$	())()())$	reduz por 5
$1(_6(_6P_7$	())()())$	RECUSADA

LISTA_13_EXERCICIO_1_LETRA_A

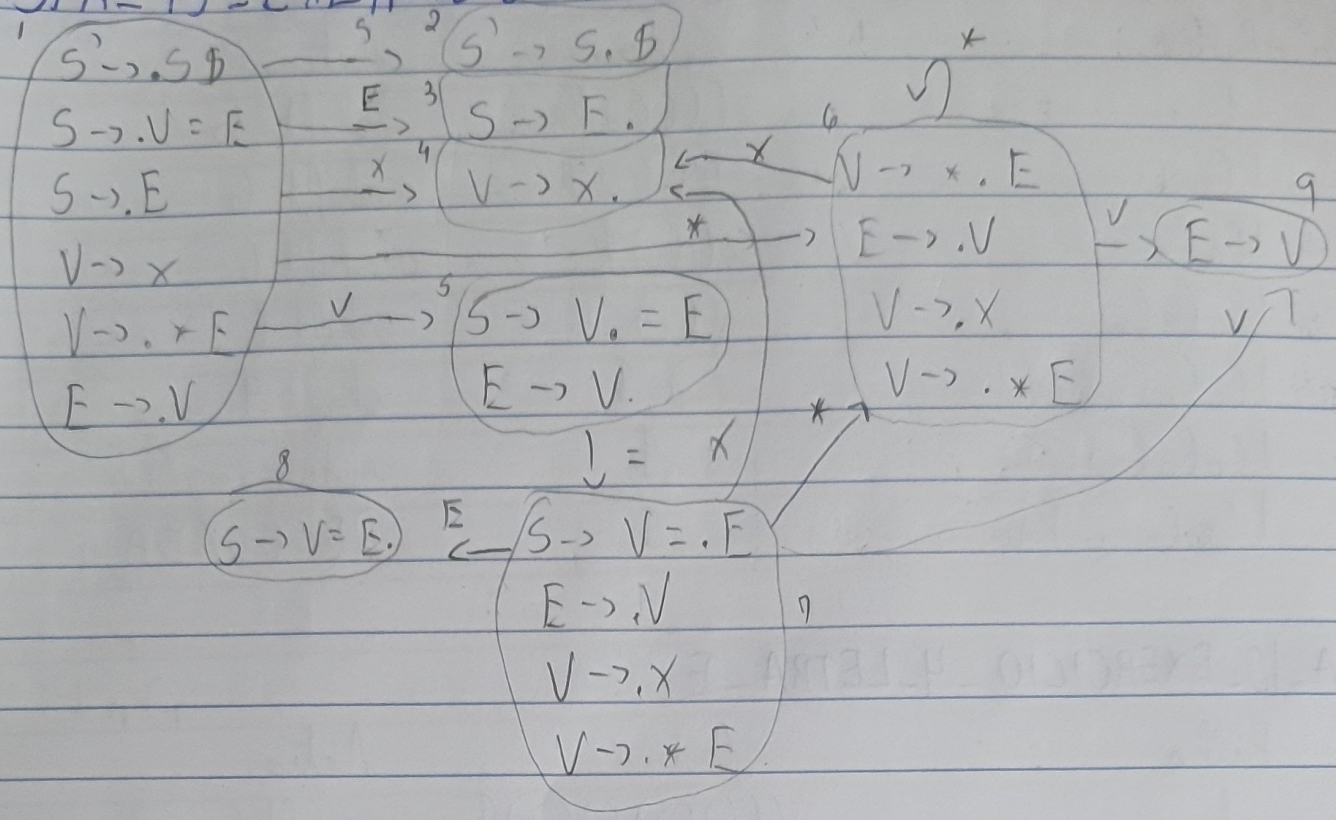
	NULL	First	Follow
S'	X	X, *	
S	X	X, *	$
E	X	X, *	$ =
V	X	X, *	= $

$	=	X	*
		S'->S$	S'->S$
		S->E	S->E
		S->V=E	S->V=E
		E->V	E->V
		V->X	V->*E

∴ NÃO É LL(1)

Marco Túlio Alves de Barros

LISTA_13_EXERCICIO_1_LETRA_B



	$	*	=	x	S	E	V
1		s6		s4	g2	g3	g5
2	A						
3	r2	r2	r2	r2			
4	r4	r4	r4	r4			
5	r3	r3	r3 s7	r3			
6		s6		s4			g9
7		s6		s4		g8	
8	r1	r1	r1	r1			
9	r3	r3	r3	r3			

∴ NÃO É LR(0)

Marco Túlio Alves de Barros

LISTA_13_EXERCICIO_1_LETRA_C

Follow(E) = {=, $}, reduz apenas nos follows de E

	$	*	=	x	S	E	V
1		s6		s4	g2	g3	g5
2	A						
3	r2						
4	r4		r4				
5	r3		r3 s7				
6		s6		s4			g9
7		s6		s4		g8	
8	r1						
9	r3		r3				

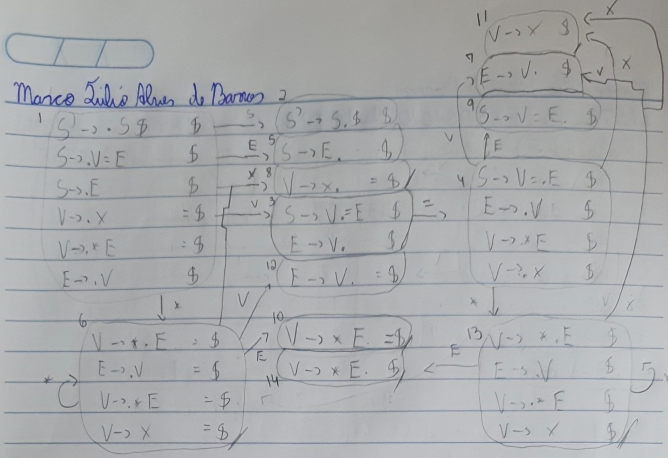
∴ Também não é SLR pois ainda há conflitos

LISTA_13_EXERCICIO_1_LETRA_D

S' -> . S $	$		S' -> . S $	$
S -> . V = E	$		S -> . V = E	$
S -> . E	$	=>	S -> . E	$
V -> . x	=		V -> . x	= $
V -> * E	=		V -> . * E	= $
E -> . V	$		E -> . V	$
V -> . x	$			
V -> * E	$			

TABELA e Automato LR(1)?

Marco Julio Alves de Barros 2



	$	*	=	x	S	E	V
1		s6		s8	g2	g5	g3
2	A						
3	r3		s4				
4		s13		s11		g9	g7
5	r2						
6		s6		s8		g10	g12
7	r3						
8	r4		r4				
9	r1						
10	r5		r5				
11	r4						
12	r3		r3				
13		s13		s11		g14	g7
14	r5						

∴ E LR(1) ✓

Marcos Túlio Alves de Barros

LISTA_13_EXERCICIO_1_LETRA_E

São iguais os estados: 6 = 13 10 = 14 ∴ Teremos 10
(com exceção do fallow) 8 = 11 7 = 12 estados

	$	*	=	x	S	E	V
1		s6		s8	g2	g5	g3
2	A						
3	r3		s4				
4		s6		s8		g9	g7
5	r2						
6		s6		s8		g10	g7
7	r3		r3				
8	r4		r4				
9	r1						
10	r5		r5				

∴ É LALR(1) ✓

LISTA_13_EXERCICIO_6_LETRA_A

S' -> . S $ $
S -> . S ; S $
S -> . id := E $
S -> . print (L) $
S -> . S ; S ;
S -> . id := E ;
S -> . print (L) ;

=>

S' -> S $ $
S -> S ; S ; S
S -> id := E ; S
S -> print (L) ; S

LISTA_13_EXERCICIO_6_LETRA_A

Marco Túlio Pelus de Barros

1
S' -> S $ $
S -> S; S $;
S -> id := E $;
S -> print (L) $;

2
S -> S. $ $
S -> S. ; S $;

3
S -> id. := E $;

4
S -> print. (L) $;

5
S -> S; S. $;
S -> S. ; S $;

6
S -> S; S. $;
S -> S. ; S $;

5
S -> S; S $;
S -> .S; S $;
S -> .id := E $;
S -> .print (L) $;

7
S -> print (.L) $;
L -> .E $;)
L -> .L, E $;)
E -> id $; +)
E -> num $; +)
E -> E+E $; +)
E -> (S,E) $; +)

8
S -> id := .E $;
E -> id $; +)
E -> num $; +)
E -> .E+E $; +)
E -> (S,E) $; +)

9
E -> num. $; +)

10
E -> id. $; +)

11
S -> print (L.) $;
L -> L. , E $;)

12
S -> print (L). $;

13
L -> E. $;)
E -> E.+ E $; +)

14
S -> id := E. $;
E -> E.+E $; +)

15
E -> E+.E $; +)
E -> id $; +)
E -> num $; +)
E -> E+E $; +)
E -> (S,E) $; +)

16
L -> L, .E $;)
E -> id $;)+
E -> num $;)+
E -> E+E $;)+
E -> (S,E) $;)+

17
E -> E.+E $;)+
E -> E+E. $;)+

18
L -> L, E. $;)
E -> E.+E $;)+

19
E -> (.S,E) $; +)
S -> .S; S $; +)
S -> .id := E $; +)
S -> .print (L) $; +)

20
E -> (S.,E) $; +)
S -> S.; S $; +)

21
E -> (S,.E)
E -> .id | .num
E -> E+E | (S,E)

22
E -> (S,E.)
E -> E.+E

23
E -> (S,E). $; +)

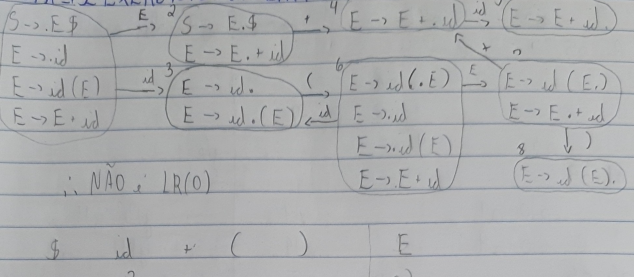
Marco Júlio Alves de Barros

	$	;	id	num	print	:=	+	,	(	)	S	E	L
1			s3			s4					g2		
2	A	s5											
3							s8						
4									s7				
5			s3			s4					g6		
6	r1	s5											
7			s10	s9					s19			g13	g11
8			s10	s9					s19			g14	
9	r5	r5					r5		r5				
10	r4	r4					r4		r4				
11								s16	s12				
12	r3	r3					r3		r3				
13	r8	r8					s15	r8	r8				
14	r2	r2					s15						
15			s10	s9					s19			g17	
16			s10	s9					s19			g18	
17	r6	r6					s17		r6				
18	r9	r9					s15	r9	r9				
19			s3			s4					g20		
20		s5						s21					
21			s10	s9					s19			g22	
22							s15			s23			
23	r7	r7					r7		r7				

NÃO SEI SE E LALR(1)

Marco Túlio Alves de Barros

LISTA_13 EXERCICIO_2_LETRA_A



∴ NÃO é LR(0)

	$	id	+	(	)	E
1		s3				g2
2	A		s4			
3	r1	r1	r1	r1 s3	r1	
4		s5				
5	r3	r3	r3	r3	r3	
6		s3				g7
7			s4		s8	
8	r2	r2	r2	r2	r2	

LISTA_13_EXERCICIO_2_LETRA_B

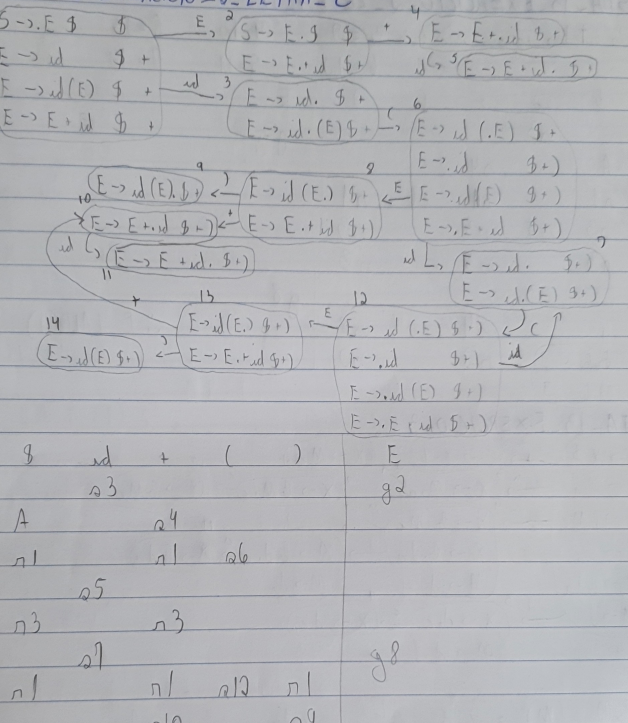
Follow (E) = { $,), + }

	$	id	+	(	)	E
1		s3				g2
2	A		s4			
3	r1		r1	s6	r1	
4		s5				
5	r3		r3		r3	
6		s3				g7
7			s4		s8	
8	r2		r2		r2	

∴ É SLR

Marco Túlio Alves de Barros

LISTA_13_EXERCÍCIO_2_LETRA_C



	$	id	+	(	)	E
1		s3				g2
2	A		s4			
3	r1		r1	s6		
4		s5				
5	r3		r3			
6		s7				g8
7	r1		r1	s12	r1	
8			s10		s9	
9	r2		r2			
10		s11				
11	r3		r3		r3	
12		s7				g13
13			s10		s14	
14	r2		r2		r2	

∴ É LR(1)

Marco Túlio Alves de Barros

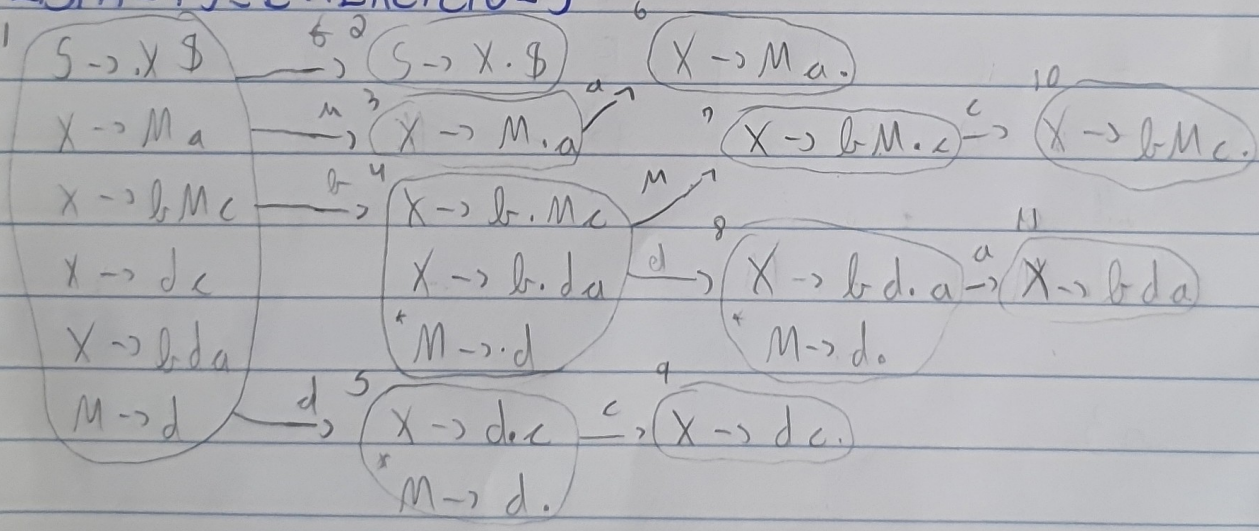
LISTA_13_EXERCICIO_2_LETRA_D

Os estados iguais são: 3 = 7 4 = 10 8 = 13
6 = 12 5 = 11 9 = 14

	$	id	+	(	)	E
1		s3				g2
2	A		s4			
3	r1		r1	s6	r1	
4	r3	s5	r3		r3	
5	r3		r3		r3	
6		s3				g8
7			s4		s8	
8	r2		r2		r2	

∴ É LALR(1)

LISTA_13_EXERCICIO_3



Follow(M) = {a, c} e redução apenas neles

∴ SLR table:

Marco Túlio Alves de Barros

	$	a	b	c	d	X	M
1			s4		s5	g2	g3
2	A						
3		s6					
4		r5		r5	s8		g7
5		r5		r5 s9			
6	r1						
7				s10			
8		s11 r5		r5			
9	r3						
10	r2						
11	r4						

∴ A gramática NÃO É SLR

LALR (1):

1
S -> X $ $
X -> .M a $
X -> b M c $
X -> d c $
X -> b d a $
M -> d $ a

$\xrightarrow{X}$ 2: S -> X. $ $

$\xrightarrow{M}$ 3: X -> M. a $

$\xrightarrow{b}$ 4: X -> b. M c $
X -> b. d a $
M -> .d $ c

$\xrightarrow{d}$ 5: X -> d. c $
M -> d. $ a

$\xrightarrow{c}$ 9: X -> d c. $

$\xrightarrow{a}$ 6: X -> M a. $

$\xrightarrow{a}$ 7: X -> b d. a $
M -> d. $ c

$\xrightarrow{a}$ 10: X -> b d a. $

$\xrightarrow{M}$ 8: X -> b M. c $

$\xrightarrow{c}$ 11: X -> b M c. $

∴ LALR (1) table:

Marco Túlio Alves de Barros

	$	a	b	c	d	X	M
1			s4		s5	g2	g3
2	A						
3		s6					
4					s7		g8
5	r5	r5		s9			
6	r6						
7	r5	s10		r5			
8				s11			
9	r3						
10	r4						
11	r2						

∴ É LALR(1)

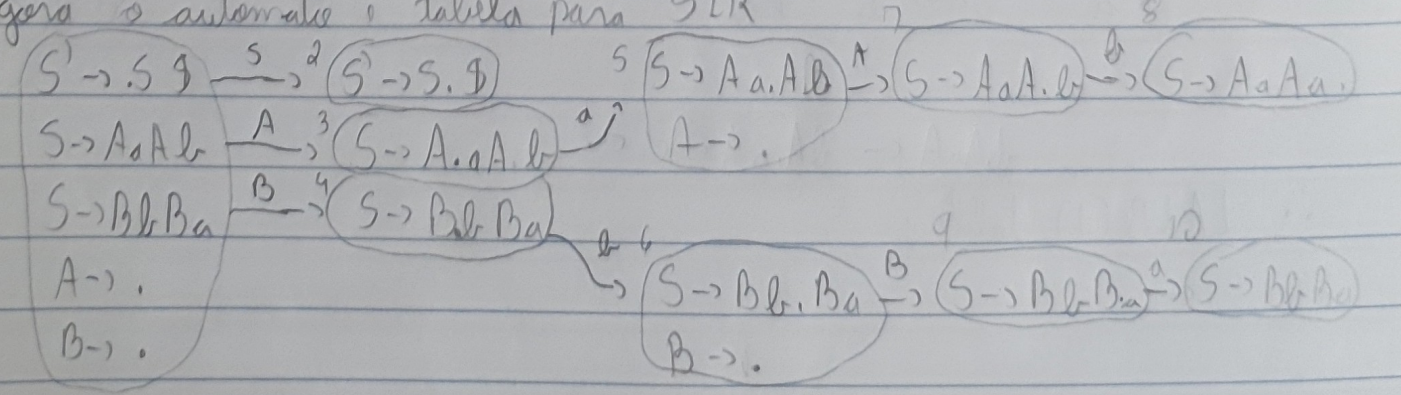
LISTA_13_EXERCICIO_4

	NULL	First	Follow
S'	N	a, b	
S	N	a, b	$
A	Y	-	a, b
B	Y	-	a, b

$	a	b
	S'->S$	S'->S$
	S->AaAb	S->BbBa
	A->	A->
	B->	B->

∴ É LL(1)

Agora o automato e tabela para SLR



Marco Túlio Alves de Barros

	$	a	b	λ	S	A	B
1		r3 r4	r3 r4		g2	g3	g4
2	A						
3		s5					
4			s6				
5		r3	r3			g7	
6		r4	r4				g9
7			s8				
8	r1						
9		s10					
10	r2						

∴ Como houve conflitos NÃO É SLR