

1.

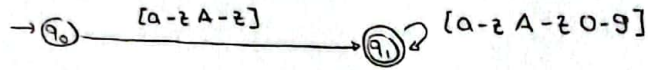
a) ID:  $[a-zA-Z][a-zA-Z0-9]^*$

ASSING: '='

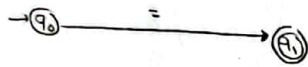
REAL:  $[0-9]^*.[0-9]^*$

NUM:  $[0-9]^+$

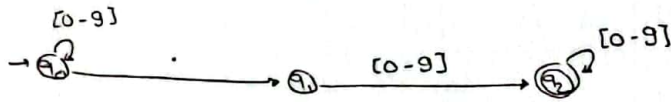
b) ID



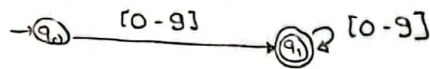
ASSING



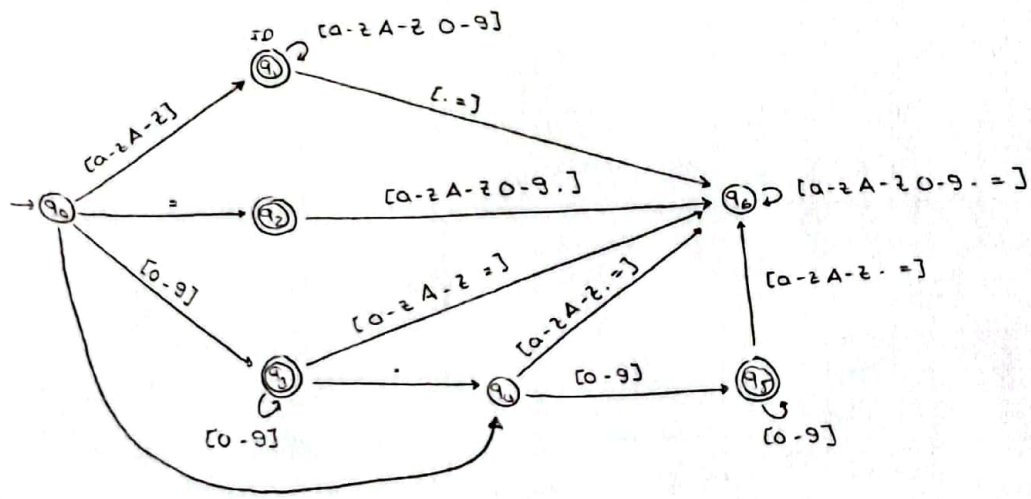
REAL



NUM



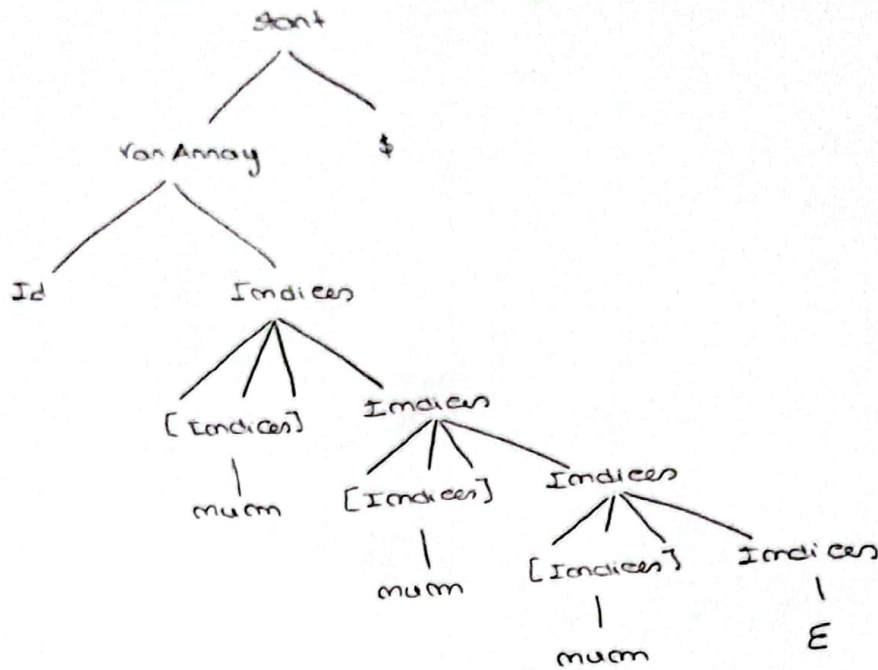
c)



2.

(1) id [num] [num] [num]

id [num] [enum] [enum]



b)

	(simb terminais) Nullable	(em frente) First	(atrás e encerrado) Follow
start	N	{id}	$\emptyset$
VarArray	N	{id}	{ '\$', ']' }
Indices	S	{ '[' }	{ '\$', ']', ']' }
Index	N	{ num, id }	{ ']' }

	\$	id	num	]	[
start		start $\rightarrow$ VarArray \$			
VarArray		VarArray $\rightarrow$ id Indices			
Indices	Indices $\rightarrow$ E			Indices $\rightarrow$ E	Indices $\rightarrow$ [Indices] Indices
Index		Index $\rightarrow$ id Index $\rightarrow$ VarArray	Index $\rightarrow$ num		

Logo a gramática não é LL(1)  
(tem 2 produções)

c) start  $\rightarrow$  VarArray \$

VarArray  $\rightarrow$  id Indices

Indices  $\rightarrow$  [Index] Indices  
| E

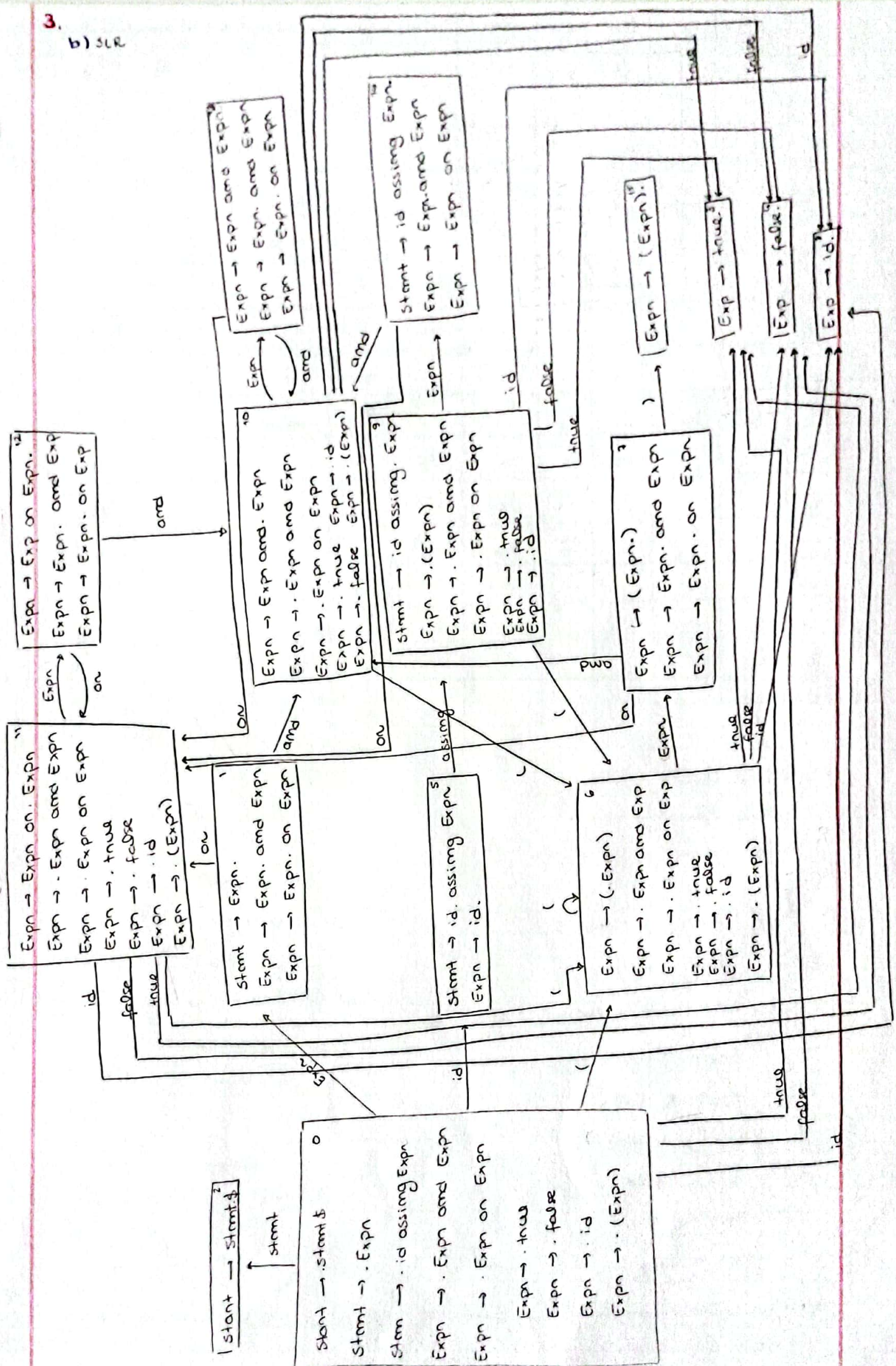
Index  $\rightarrow$  VarArray  
| num

nesta célula deixa de haver 2 produções, por isso, existe 1 produção em cada célula preenchida, logo é LL(1)



3.

b) SLR





	\$	)	(	id	false	true	on	and	=	stmt	Expr
0			\$6	\$5	\$4	\$3				\$2	\$1
1	$\pi_1$						$s_{11}$	$s_{10}$			
2	acc										
3	$\pi_5$	$\pi_5$					$\pi_5$	$\pi_5$			
4	$\pi_6$	$\pi_6$					$\pi_6$	$\pi_6$			
5	$\pi_4$		$\pi_4$				$\pi_4$	$\pi_4$	$s_9$		
6				$s_8$	$s_4$	$s_3$					$g_7$
7			$s_{15}$				$s_{11}$	$s_{10}$			
8	$\pi_4$	$\pi_4$					$\pi_4$	$\pi_4$			
9			$s_6$	$s_8$	$s_4$	$s_3$					$g_{14}$
10			$s_6$	$s_8$	$s_4$	$s_3$					$g_{13}$
11			$s_6$	$s_8$	$s_4$	$s_3$					$g_{12}$
12	$\pi_4$	$\pi_4$					$s_{11}/\pi_4$	$s_{10}/\pi_4$			
13	$\pi_3$	$\pi_3$					$s_{11}/\pi_3$	$s_{10}/\pi_3$			
14	$\pi_2$						$s_{11}$	$s_{10}$			
15	$\pi_8$	$\pi_8$					$\pi_8$	$\pi_8$			

Follow (stmt) =  $\{ \emptyset \}$

Follow (stmt) =  $\{ \$ \}$

Follow (Expr) =  $\{ \text{and}, \text{on}, ), \$ \}$

$\pi_0$ : stmt  $\rightarrow$  stmt \$

$\pi_1$ : stmt  $\rightarrow$  Expr

$\pi_2$ : stmt  $\rightarrow$  id assign Expr

$\pi_3$ : Expr  $\rightarrow$  Expr and Expr

$\pi_4$ : Expr  $\rightarrow$  Expr on Expr

$\pi_5$ : Expr  $\rightarrow$  true

$\pi_6$ : Expr  $\rightarrow$  false

$\pi_7$ : Expr  $\rightarrow$  id

$\pi_8$ : Expr  $\rightarrow$  ( Expr )

a) A gramática não é SRL porque tem empilhados e como não é SRL também não pode ser LR(0)

→ c) start → var Array

stmt → Expr

1 id = Expr

Expr → Expr op Term

1 Term

Term → Term and Factor

1 Factor

Factor → true

1 false

1 id

1 (Expr)

4

a) fib:

1 m := param [0]

2 if m != 1 go to 5

3 t<sub>1</sub> := 1

4 return t<sub>1</sub>

5 if m != 2 go to 8

6 t<sub>0</sub> := 1

7 return t<sub>0</sub>

8 t<sub>2</sub> := m - 2

9 t<sub>3</sub> := m - 1

10 param t<sub>2</sub>

11 t<sub>4</sub> := call fib t<sub>2</sub>, 1

12 param t<sub>3</sub>

13 t<sub>5</sub> := call fib t<sub>3</sub>, 1

14 t<sub>6</sub> := t<sub>4</sub> + t<sub>5</sub>

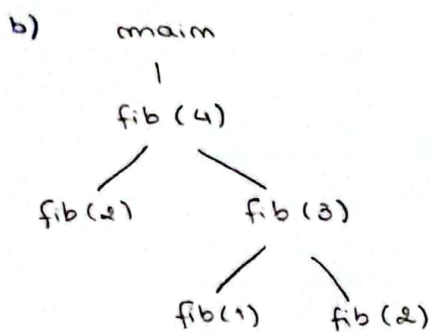
15 return t<sub>6</sub>

main:

16 param 4

17 t<sub>1</sub> := call fib 4, 1

18 return t<sub>1</sub>

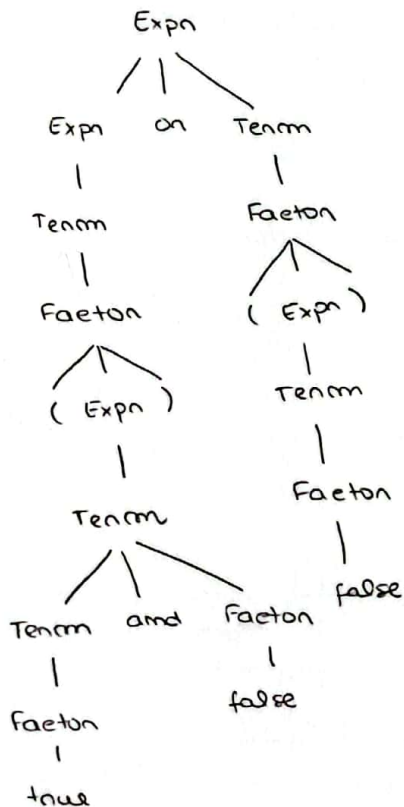


c) Parâmetros formais, valores de retorno, old stack pointer, frame pointer, alternativas, ...

Frame 2020/20 - monomodal

1.

a) (true and false) on (false)



b)

	Nullable	First	Follow
Expr	N	{true, false, (}	{on, )}
Term	N	{true, false, (}	{and, on, )}
Factor	N	{true, false, (}	{and, on, )}



	$\&$	on	and	true	false	(	)
Expr				Expr $\rightarrow$ Expr on Term Expr $\rightarrow$ Term	Expr $\rightarrow$ Expr on Term Expr $\rightarrow$ Term	Expr $\rightarrow$ Expr on Term Expr $\rightarrow$ Term	
Term				Term $\rightarrow$ Term and Factor Term $\rightarrow$ Factor	Term $\rightarrow$ Term and Factor Term $\rightarrow$ Factor	Term $\rightarrow$ Term and Factor Term $\rightarrow$ Factor	
Factor				Factor $\rightarrow$ true	Factor $\rightarrow$ false	Factor $\rightarrow$ (Expr)	

Existe mais do que uma produç o na mesma c lula, logo a gram tica n o   LL(1)

c) Eliminar recurs idade   esquerda

Expr  $\rightarrow$  Term Expr<sub>1</sub>

Expr<sub>1</sub>  $\rightarrow$  on Term Expr<sub>1</sub>

Term  $\rightarrow$  |  $\epsilon$

Term  $\rightarrow$  Factor Term<sub>1</sub>

Term<sub>1</sub>  $\rightarrow$  and Factor Term<sub>1</sub>

|  $\epsilon$

Factor  $\rightarrow$  true

| false

| (Expr)

	nullable	First	Follow
Expr	N	{true, false, (}	{})}
Expr <sub>1</sub>	S	{on}	{})}
Term	N	{true, false, (}	{on, )}
Term <sub>1</sub>	S	{and}	{on, )}
Factor	N	{true, false, (}	{and, )}

First (Expr) = First (Term) =

= First (Factor) = {true, false, (}

First (Expr<sub>1</sub>) = {on}

First (Term<sub>1</sub>) = {and}

Follow (Expr) = {)}

Follow (Expr<sub>1</sub>) = Follow (Expr) = {)}

Follow (Term) = First (Expr<sub>1</sub>)  $\cup$

$\cup$  Follow (Expr<sub>1</sub>) = {on, )}

Follow (Term<sub>1</sub>) = Follow (Term) = {on, )}

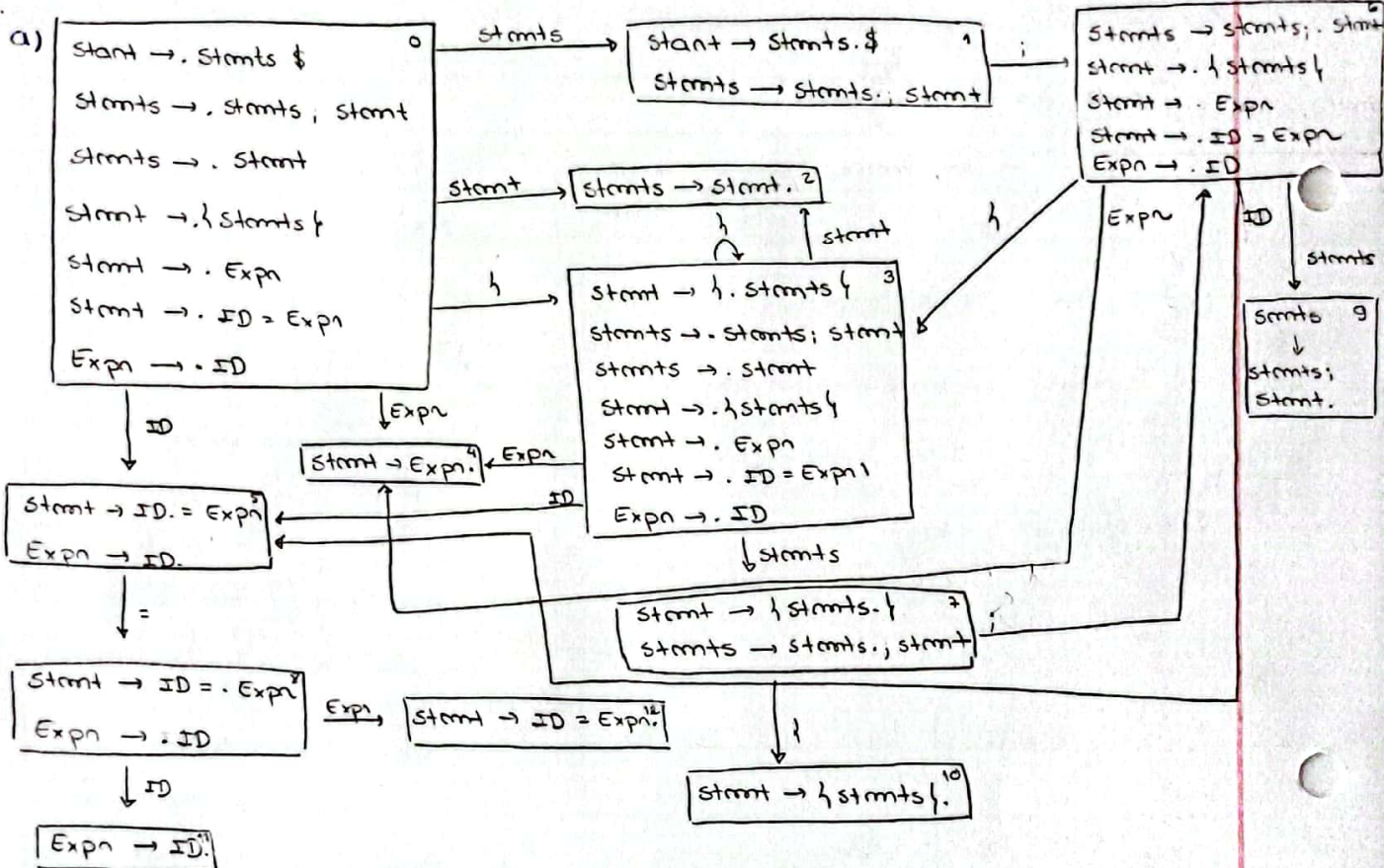
Follow (Factor) = First (Term<sub>1</sub>)  $\cup$

$\cup$  Follow (Term<sub>1</sub>) = {and, )}



	\$	or	and	true	false	(	)
Expr				Expr $\rightarrow$ Term Expr1	Expr $\rightarrow$ Term Expr1	Expr $\rightarrow$ Term Expr1	
Expr1			Expr1 $\rightarrow$ or Term Expr1				Expr1 $\rightarrow$ E
Term				Term $\rightarrow$ Factor Term1	Term $\rightarrow$ Factor Term1	Term $\rightarrow$ Factor Term1	
Term1		Term1 $\rightarrow$ E	Term1 $\rightarrow$ and Factor Term1				Term1 $\rightarrow$ E
Factor				Factor $\rightarrow$ true	Factor $\rightarrow$ false	Factor $\rightarrow$ ( Expr )	

2.



b)

	\$	;	{	}	=	ID	stmts	stmt	Expr
0			S <sub>3</sub>			S <sub>5</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>4</sub>
1	a <sub>cc</sub>	S <sub>6</sub>							
2	r <sub>3</sub>	r <sub>3</sub>		r <sub>3</sub>					
3			S <sub>3</sub>			S <sub>5</sub>	q <sub>4</sub>	q <sub>2</sub>	q <sub>4</sub>
4	r <sub>5</sub>	r <sub>5</sub>		r <sub>5</sub>					
5	r <sub>7</sub>	r <sub>7</sub>		r <sub>7</sub>	S <sub>8</sub>				
6			S <sub>3</sub>			S <sub>5</sub>	q <sub>9</sub>		q <sub>4</sub>
7		S <sub>6</sub>	S <sub>10</sub>						
8						S <sub>11</sub>			q <sub>12</sub>
9	r <sub>2</sub>	r <sub>4</sub>		r <sub>2</sub>					
10	r <sub>4</sub>	r <sub>4</sub>		r <sub>4</sub>					
11	r <sub>2</sub>	r <sub>4</sub>		r <sub>2</sub>					
12	r <sub>6</sub>	r <sub>6</sub>		r <sub>6</sub>					

Follow (start) =  $\emptyset$

Follow (stmts) =  $\{ \$, ;, \{, \} \}$

Follow (stmt) =  
= Follow (stmts) =  
=  $\{ \$, ;, \{, \} \}$

Follow (Expr) =  
= Follow (stmt) =  
=  $\{ \$, ;, \{, \} \}$

A gramática é SRL  
pois não apresenta  
nenhum conflito  
(shift/reduce nem  
reduce/reduce)



c)	Pilha	Entrada	Ata
	0	{ id = id ; id \$	S <sub>3</sub>
	0, 13	id = id ; id \$	S <sub>5</sub>
	0, 13, ID5	= id ; id \$	S <sub>8</sub>
	0, 13, ID5, = 8	id ; id \$	S <sub>11</sub>
	0, 13, ID5, = 8, ID 11	}; id \$	r <sub>4</sub>
	0, 13, ID5, = 8, Expr 12	}; id \$	r <sub>6</sub>
	0, 13, stmt 2	}; id \$	r <sub>3</sub>
	0, 13, stmts 7, } 10	}; id \$	r <sub>4</sub>
	0, stmt 2	; id \$	r <sub>3</sub>
	0, stmt 1	; id \$	S <sub>6</sub>
	0, stmts 1, ; 6	id \$	S <sub>5</sub>
	0, stmts 1, ; 6, ID5	\$	r <sub>2</sub>
	0, stmts 1, ; 6, Expr 4	\$	r <sub>5</sub>
	0, stmts 1, ; 6, stmts 9	\$	r <sub>2</sub>
	0, stmts 1	\$	acc

3.

a) fib:

```

1 m := param[0]
2 if m != 1 go to 5
3 t1 := 1
4 return t1
5 if m != 2 go to 8
6 t2 := 1
7 return t2
8 t3 := m - 2
9 t4 := m - 1
10 param t3
11 t5 := call fib t3, 1
12 param t4
13 t6 := call fib t4, 1
14 t2 := t5 + t6
15 return t2

```

main:

```

16 param 4
17 t8 := call fib, 1
18 return t8

```

b)

