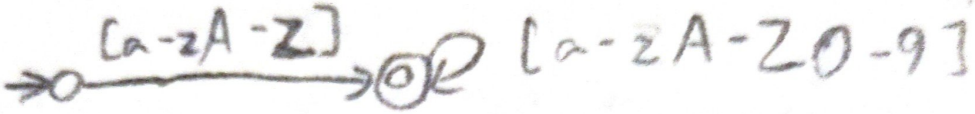
① a) ID → [a-zA-Z][a-zA-Z0-9]*

FLOAT → [0-9]*.[0-9]+ | [0-9]+.[0-9]*

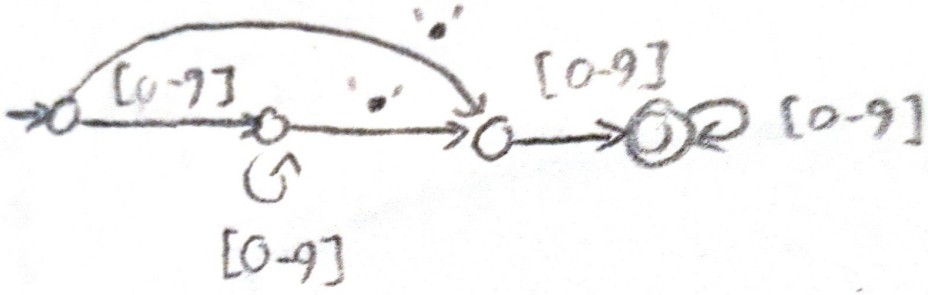
NUM → [0-9]+

OCTAL → 0[0-7]+

b) ID →



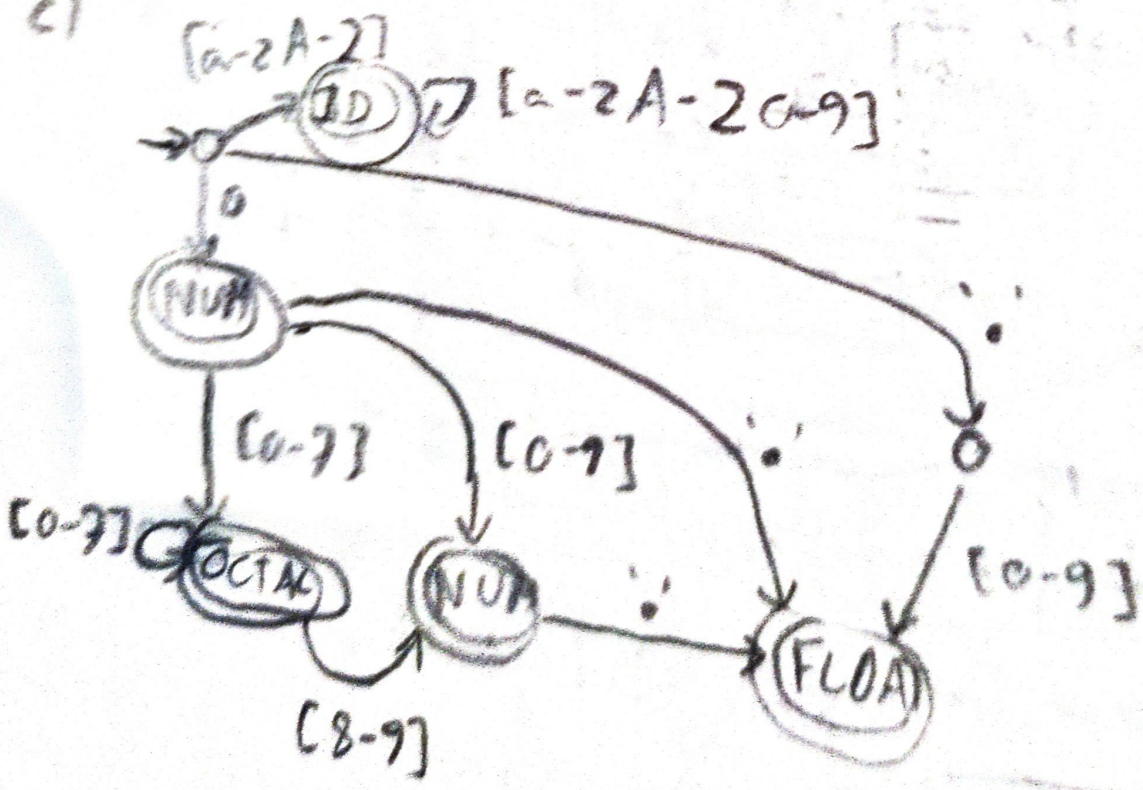
FLOAT →



NUM →

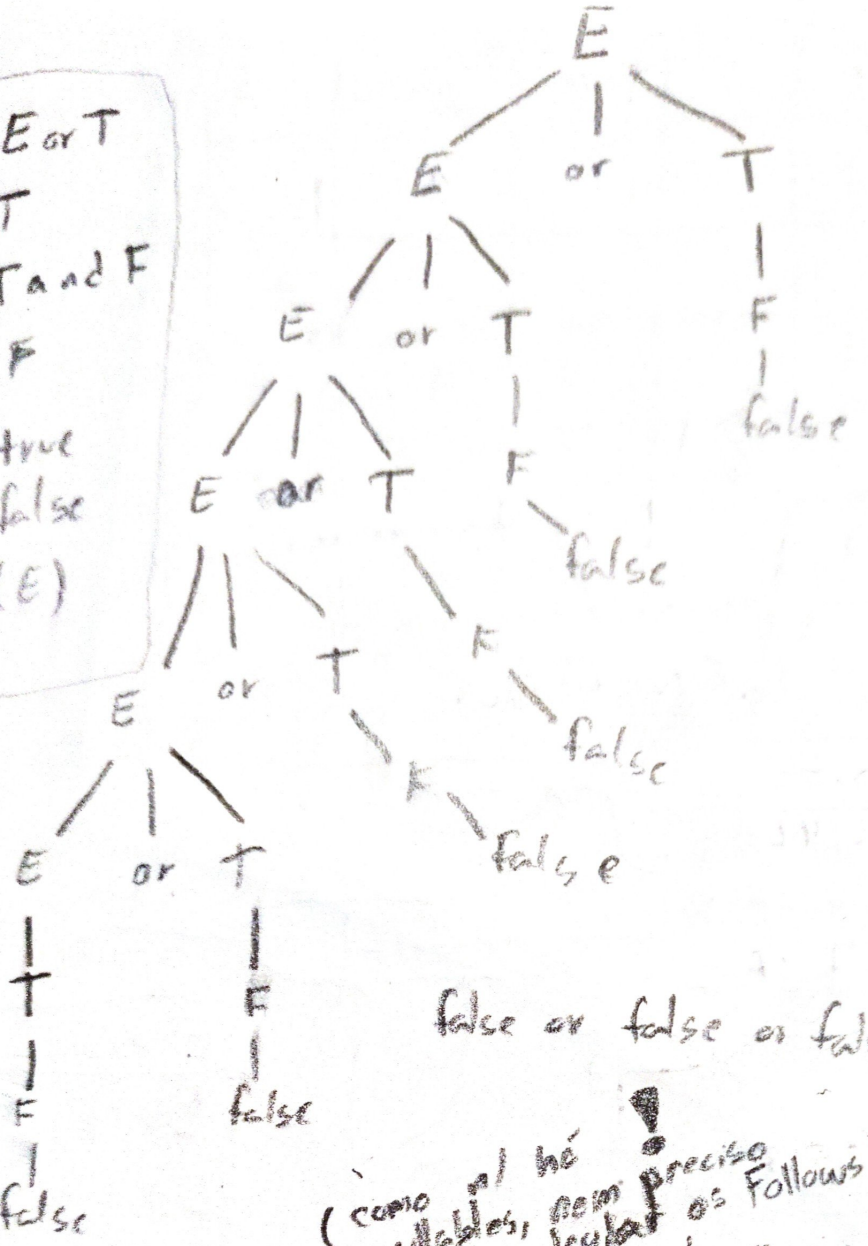
OCTAL →

c)



② a)

1- E → E or T
2- | T
3- T → T and F
4- | F
5- F → true
6- | false
7- | (E)



false or false or false or false or false or false

b) First(E) = First(T)
First(T) = First(F)
First(F) = {true, false, (}

(como não há nullables, não é preciso calcular os Follows)

	or	and	(	true	false	)	$
E			1,2	1,2	1,2		
T			3,4	3,4	3,4		
F			7	5	6		

A gramática não é LL(1) porque há conflitos em uma ou mais entradas da parse table

c)

1- E → TE'
2- E' → or TE'
3- | ε
4- T → FT'
5- T' → and F
6- | ε
7- F → true
8- | false
9- | (E)

Remover left-recursion

	Nullable	First	Follow
E	0	t, f, (	), $
E'	1	or, ε	), $
T	0	t, f, (	or,), $
T'	1	and, ε	or,), $
F	0	t, f, (	and, or,), $

First(F) = {t, f, (}
First(T') = {and, ε}
First(T) = First(F)
First(E') = {or, ε}
First(E) = First(T)
Follow(E) = {), $}
Follow(E') = Follow(E)

Follow(T) = First(E') \ ε ∪ Follow(E') = {or,), $}
Follow(T') = Follow(T) = {or,), $}
Follow(F) = First(T') \ ε ∪ Follow(T') = {and, or,), $}

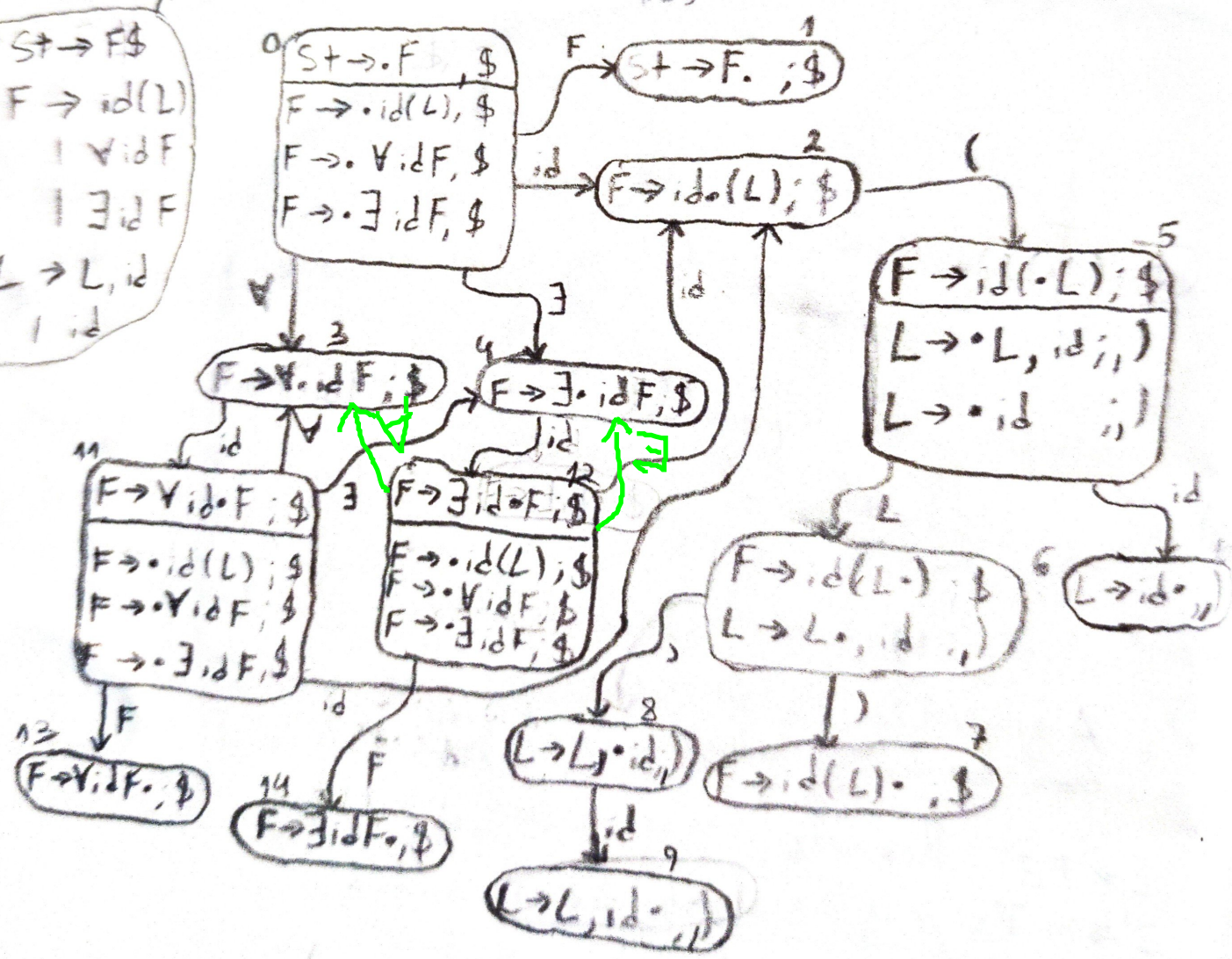
	and	or	(	)	true	false	$
E			1		1	1	
E'		2		3			3
T			4		4	4	
T'	5	6		6			6
F			9		7	8	

A gramática é LL(1) porque não há conflitos na parse-table

③

First(F) = {id, ∀, ∃}, First(L) = {id}

1- S' → F$
2- F → id(L)
3- | ∀ id F
4- | ∃ id F
5- L → L, id
6- | id



ACTION

	id	(	)	∀	∃	,	$	L	F
0	s2			s3	s4				61
1							Acc		
2		s5							
3	s11								
4	s12								
5	s10							96	
6			s7			s8			
7							r2		
8	s9								
9				r5			r5		
10				r6			r6		
11	s2				s3	s4			913
12	s2				s3	s4			914
13							r3		
14							r4		

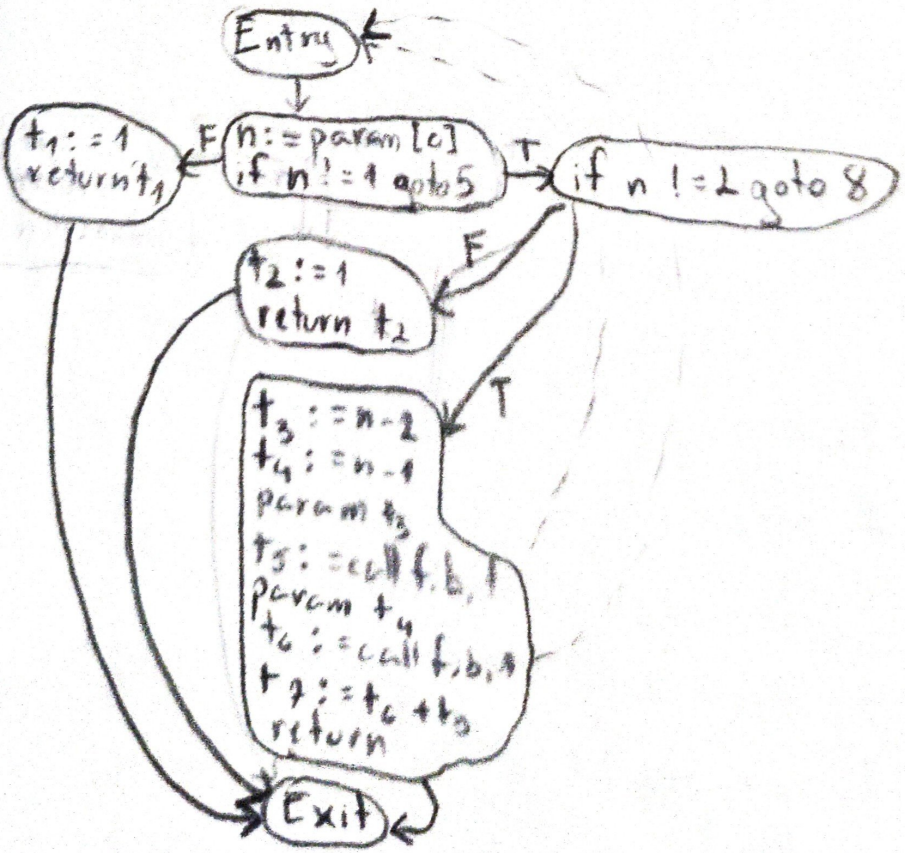
Stack	Current Head	Action
$	∀ id ∃ id id (id) $	s3
$ ∀3 $	id ∃ id id (id) $	s11
$ ∀2 id11 $	∃ id id (id) $	s4
$ ∀2 id11 ∃4 $	id id (id) $	s12
$ ∀2 id11 ∃4 id12 $	id (id) $	s2
$ ∀2 id11 ∃4 id12 id2 $	(id) $	s5
$ ∀2 id11 ∃4 id12 id2 (5 $	id) $	s10
$ ∀2 id11 ∃4 id12 id2 (5 id10 $	) $	r6
$ ∀2 id11 ∃4 id12 id2 (5 L6 $	) $	s7
$ ∀2 id11 ∃4 id12 id2 (5 L6)8 $	$	r2
$ ∀2 id11 ∃4 id12 F14 $	$	r4
$ ∀2 id11 F13 $	$	r3
$ F1 $	$	ACC

sete

a) (4) goto ← Instrução para chamar a main!

```
fib:
1- n := param[0]
2- if n != 1 goto 5      ← Também dá para usar labels
3- t1 := 1
4- return t1
5- if n != 2 goto 8
6- t2 := 1
7- return t2
8- t3 := n - 2
9- t4 := n - 1
10- param t3
11- t5 := call fib, 1    ← #arguments
12- param t4
13- t6 := call fib, 1
14- t7 := t6 + t5
15- return t7
main:
16- t8 := 4
17- param t8
19- t9 := call fib, 1
20- return t9
```

b) Leaders = {1, 3, 5, 6, 8}



c) 4