$L = \{ (z a^* i^*)^n (t i^*)^n \mid n \geq 0 \}$ $\quad A = \{z, a, i, t\}$

1) Unambiguous grammar
2) LR grammar
3) LL "

$S \to z A I S t I \mid \varepsilon$

$A \to a A \mid \varepsilon$

$I \to i I \mid \varepsilon$

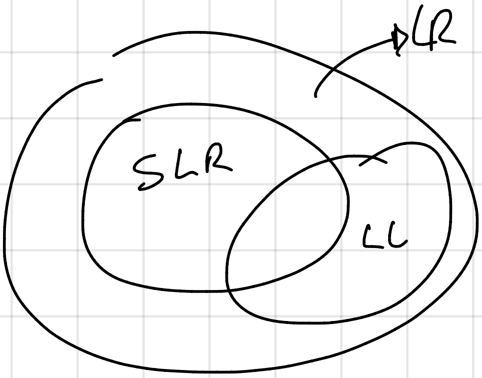
FIRST(S) = {z, ε}
FIRST(A) = {a, ε}
FIRST(I) = {i, ε}
FOLLOW(S) = {t, $}
FOLLOW(A) = {i, z, t}
FOLLOW(I) = {z, t, $}

LL(1) Table

	z	a	i	t	$
S	$S \to zAIStI$			$S \to \varepsilon$	$S \to \varepsilon$
A	$A \to \varepsilon$	$A \to aA$	$A \to \varepsilon$	$A \to \varepsilon$	
I	$I \to \varepsilon$		$I \to iI$	$I \to \varepsilon$	$I \to \varepsilon$

The grammar is LL(1) (3) ⇒ The grammar is also LR(1) (2)

⇒ The grammar is not ambiguous (1)



In addition, let's check if the grammar is SLR(1)
automaton of LR(0) items $\quad S' \to .S$ ← new initial symbol

$I_0 = S' \to . S$

$S \to rAIStI \mid \varepsilon$ (1, 2)

$A \to aA \mid \varepsilon$ (3, 4)

$I \to iI \mid \varepsilon$ (5, 6)

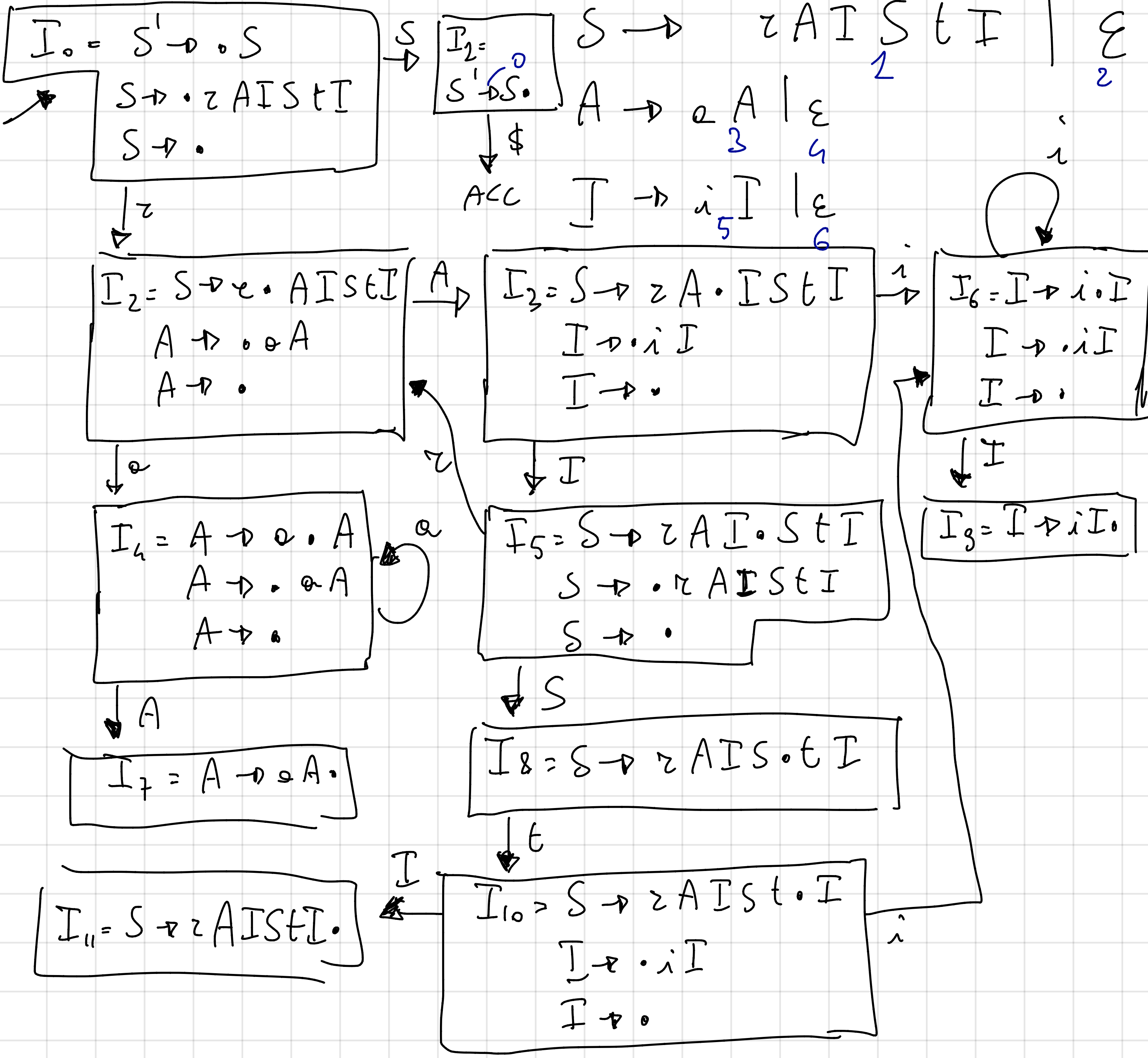


Table for the deterministic bottom-up LR parser

	r	a	i	t	$	S	A	I
0	s2			r2	r2	1		
1					ACC			
2	r4	s4	r4	r4			3	
3	r6		s6	r6	r6			5
4	r4	s4	r4	r4			7	
5	s2			r2	r2	8		
6	r6		s6	r6	r6			9
7	r3		r3	r3				
8				s10				
9	r5			r5	r5			
10	r6		s6	r6	r6			11
11				r1	r1			

The table is not multiply defined so the grammar is

SLR(1) $\Rightarrow$ LR(1)

$S \rightarrow (L)$ [1]

$L \rightarrow T , L$ [2]

$L \rightarrow T$ [3]

$T \rightarrow [\underline{label} , S]$ [4]

$T \rightarrow \underline{label}$ [5]

(S forest)
(L is a list of tree
T for tree

The forest cannot be empty

~~( )~~

2) check that the grammar is LL(1)

FIRST (S) = { (}
FIRST (L) = { label , [}
FIRST (T) = { label , [}
FOLLOW (S) = { \$,] }
FOLLOW (L) = {) }
FOLLOW (T) = { , ,) }

comma

	(	)	[	]	,	label	\$
S	1						
L			2 3			2 3	
T							

We need to factorise

1 $S \rightarrow (L)$

2 $L \rightarrow T L'$

3 $L' \rightarrow , L$

4 $L' \rightarrow \varepsilon$

5 $T \rightarrow [\underline{label} , S]$

6 $T \rightarrow \underline{label}$

FIRST(S) = { (}
FIRST (L) = { label , [}
FIRST (L') = { , , ε }
FIRST (T) = { label , [}
FOLLOW (S) = { \$,] }
FOLLOW (L) = {) }
FOLLOW (L') = {) }
FOLLOW (T) = { , ,) }

	(	)	[	]	,	label	\$
S	1						
L			2			2	
L'		4			3		
T			5			6	

1	$S \to (L)$	$S.t = L.t$
2	$L \to T L'$	$L.t = Add(T.t, L'.t)$
3	$L' \to , L$	$L'.t = L.t$
4	$L' \to \varepsilon$	$L'.t = MKE();$
5	$T \to [\underline{label} , S]$	$T.t = MKT(MKT(label.lexeme), S.t))$
6	$T \to \underline{label}$	$T.t = MKT(\underline{label}.lexeme)$

S, L, L', T There is an attribute t (tree) that holds the pointer to the structure representing the tree