TD3. Sémantique et TD2. Grammaire Attribuées

消除左递归. LL(1)型文法. AST

<1.1>

5 4 b

(1) S → I

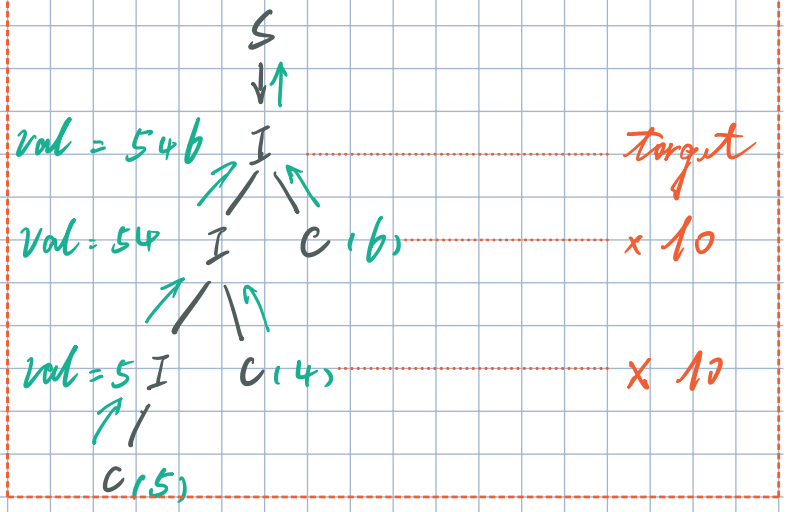
(2) I → c (terminal)

(3) I → I c

(1) S.val = I.val

(2) I.val = c.txt

(3) I.val = 10 × I.val + c.txt



*考试时需要以上三个部分

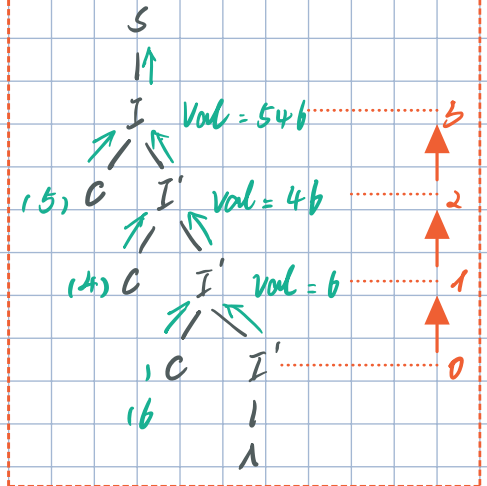
<1.5> 将上述文法 消除左递归 → LL(1)型文法

S - Attribute

L - Attribute

LL(1)

S → I	#1
I → cI'	#2
I' → cI'	#3 +1
I → ^	#4 +2



Rules: +1 I'.profonde = I'.profonde + 1

+2 I'.profonde = 0

#1 S.val = I.val

#2 I.val = c.txt × 10^(I'.profond) + I'.val

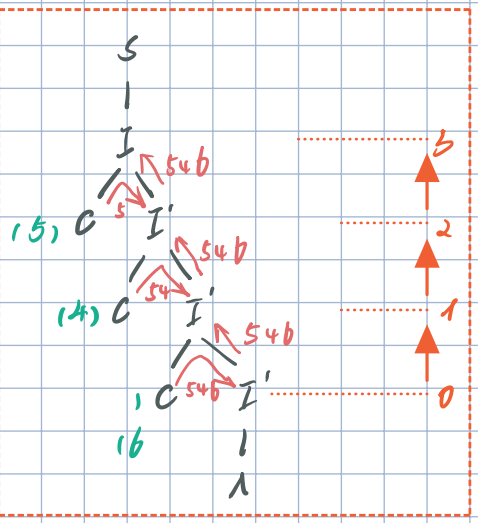
#3 I'.val = c.txt × 10^(I'.profond) + I'.val

#4 I'.val = 0

另一种形式

LL(1)

$S \to I$	#1	
$I \to cI'$	#2	+1
$I' \to cI'$	#3	+2
$I \to \wedge$	#4	



Rules:

+1 $I'.hval = c.tat$

+2 $I'.hval = I'.hval \times 10 + c.tat$

#1 $S.sval = I.sval$

#2 $I.sval = I'.sval$

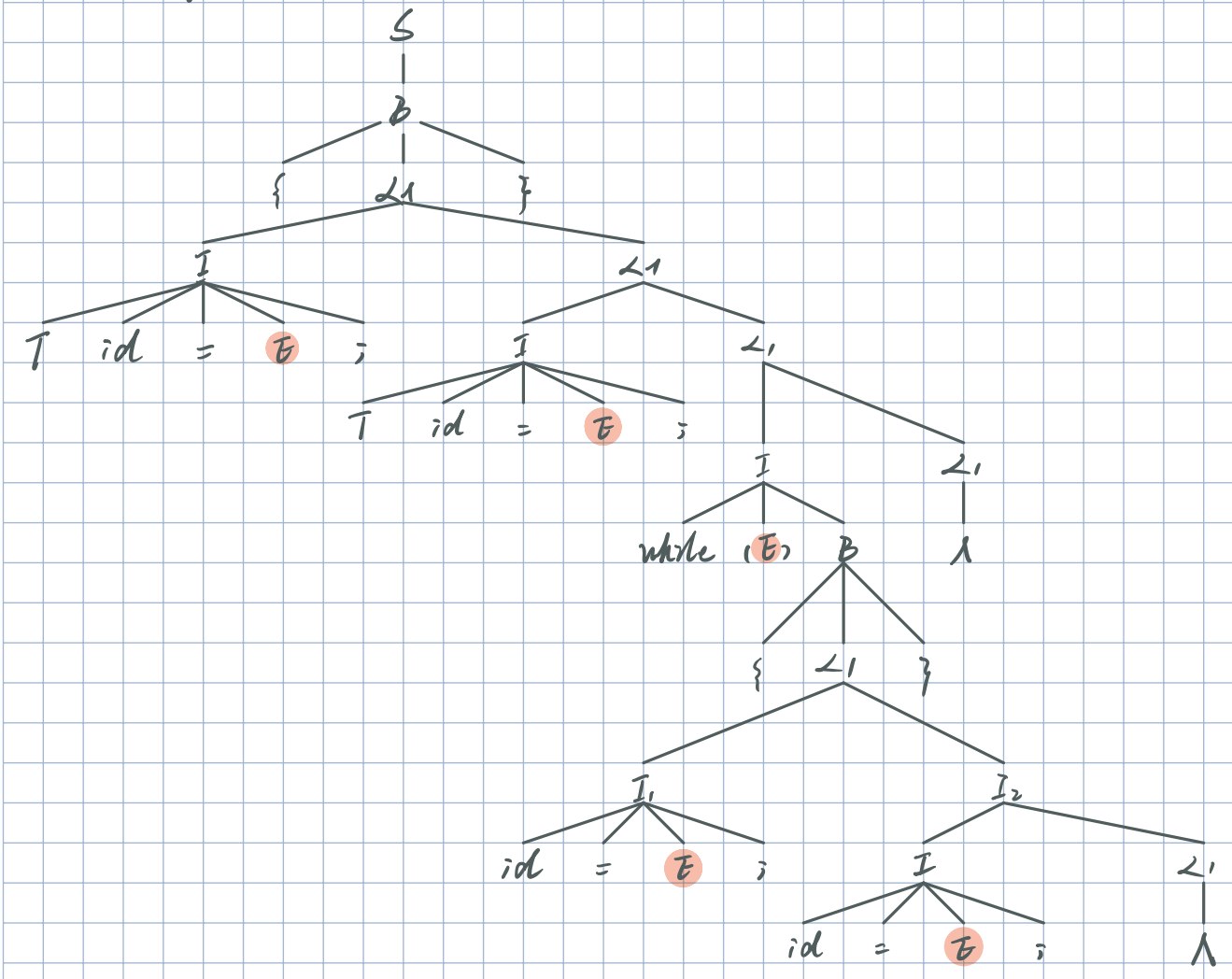
#3 $I'.sval = I'.hval$

#4 $I'.sval = I'.hval$

hval = 在分析树中从左向右的赋值

sval = 在分析树中从子节点向父节点赋值

<2.1>



Grammar LL(1):

$S \rightarrow B$

$L_1 \rightarrow \{ L_1 \} \mid I\ L_1 \mid \lambda$

$I \rightarrow T\ id = E \mid id = E \mid if\ (E)\ B\ else\ B \mid if\ (E)\ B$
$\mid while\ (E)\ B$

$T \rightarrow \ldots <typage>$

Rules:

#1 S.ast = B.ast

#2 B.ast = new createBlock(L1.ast)

#3 L1.ast = L1.ast add(I.ast)

#4 L1.ast = new Vector<Instruction>();

#5 I.ast = new VariableDeclaration(id.txt, T.ast, E.ast)

#6 I.ast = new Assignment(id.txt, E.ast)

#7 I.ast = new Conditional(E.ast, B.ast, B.ast)

#8 I.ast = new Conditional(E.ast, B.ast, null)

#9 I.ast = new Repetition(E.ast, B.ast)