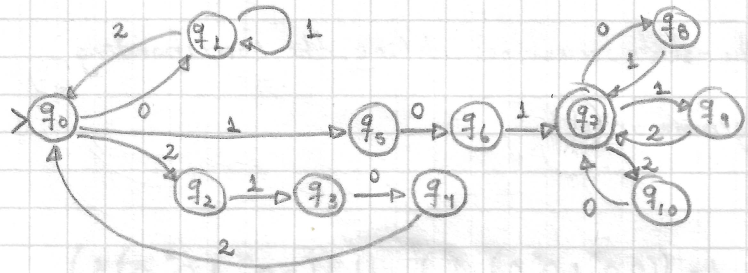
Computer Science III Omar Armando Neira Ordoñez 20192020110

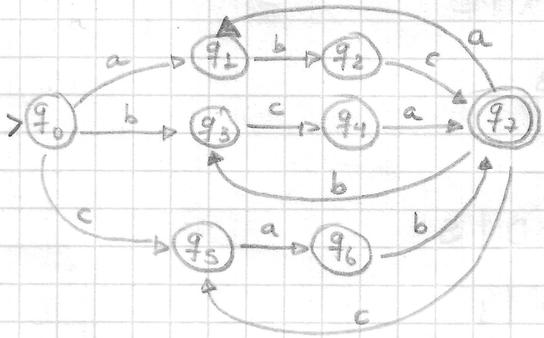
Workshop No. 1

1 For each of the following languajes, define the corresponding finite-state machine:

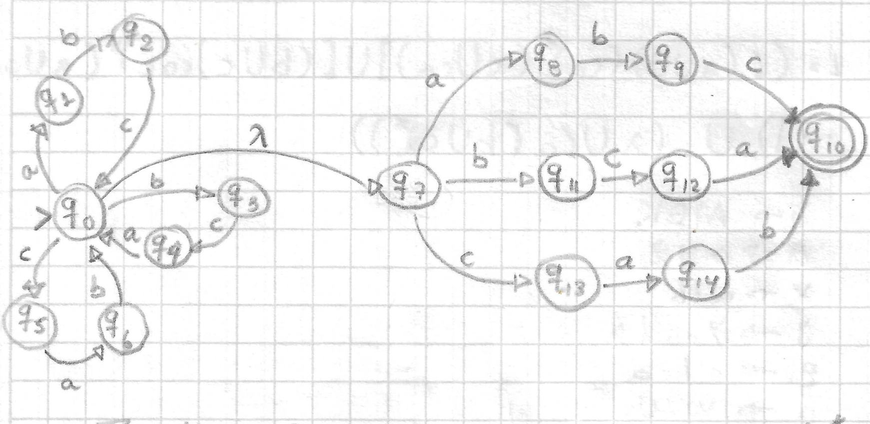
(i) $\Sigma = \{0,1,2\}$. $L = (01^*2 \cup 2102)^* 101 (01 \cup 12 \cup 20)^*$



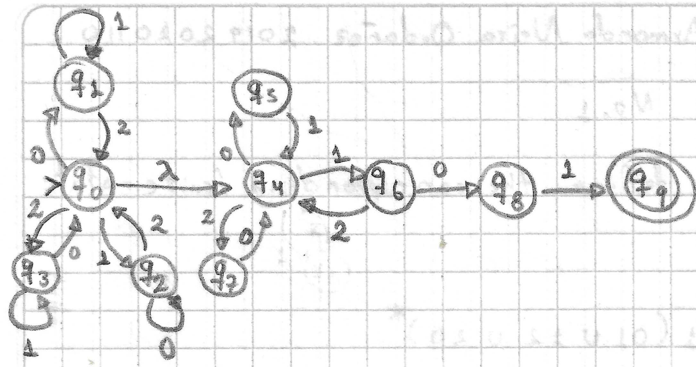
(ii) $\Sigma = \{a,b,c\}$. $L = (abc \cup bca \cup cab)(abc \cup bca \cup cab)^*$



(iii) $\Sigma = \{a,b,c\}$. $L = (abc \cup bca \cup cab)^* (abc \cup bca \cup cab)$

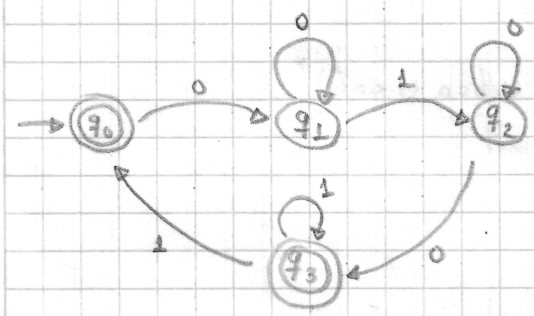


(iv) $\Sigma = \{0,1,2\}$. $L = (01^*2 \cup 10^*2 \cup 21^*0)^* (01 \cup 12 \cup 20)^* 101$



2. For each one of the following finite-state machines, define the corresponding regular expression and a generative grammar:

(i) $\Sigma = \{0, 1\}$



$L = ((00^*10^*0)\ 1^*)^* \cup ((00^*10^*0)1)$

$S \to 0A \mid \lambda \mid D$

$A \to S \mid 1B$

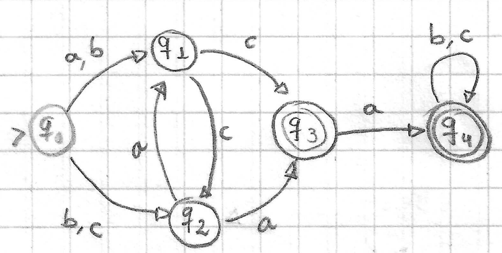
$B \to 0C$

$C \to 0B \mid 0D \mid S$

$D \to 1E \mid \lambda \mid 1$

$E \to 1D \mid S \mid \lambda$

(ii) $\Sigma = \{a, b, c\}$.



$L = ([(a \cup B)(ca)^*(c \cup ca)] \cup [(b \cup c)(ac)^*(a \cup ac) \ldots$

$\ldots]) \cup (\lambda \cup (a(b \cup c)^*))$

$S \to A \mid B \mid C$
$A \to xyz$
$x \to a \mid b$
$y \to yca \mid \lambda$
$z \to c \mid ca$
$B \to WVU$
$W \to b \mid c$
$V \to Vac \mid \lambda$
$U \to a \mid ac$
$C \to \lambda \mid aD$
$D \to DE \mid \lambda$
$E \to b \mid c$

5. As follows there is a context-free grammar to generate real numbers without sign, the alphabet is $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., +, -, E\}$:

<real> → <digits><decimal><exp>

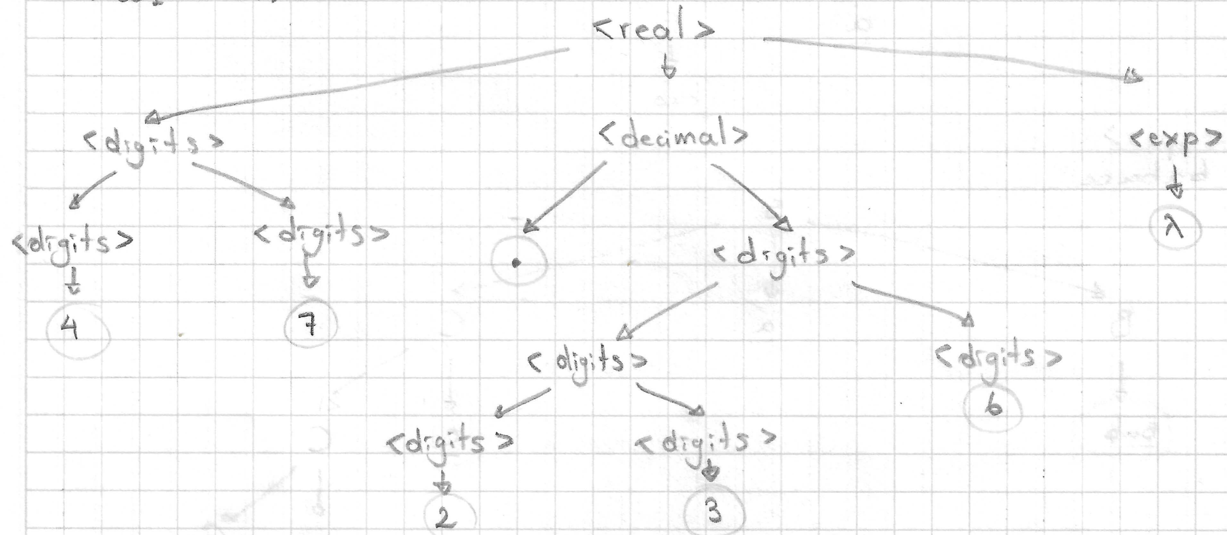
<digits> → <digits><digits> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<decimal> → . <digits> | λ Note: here i added the dot, to work with it

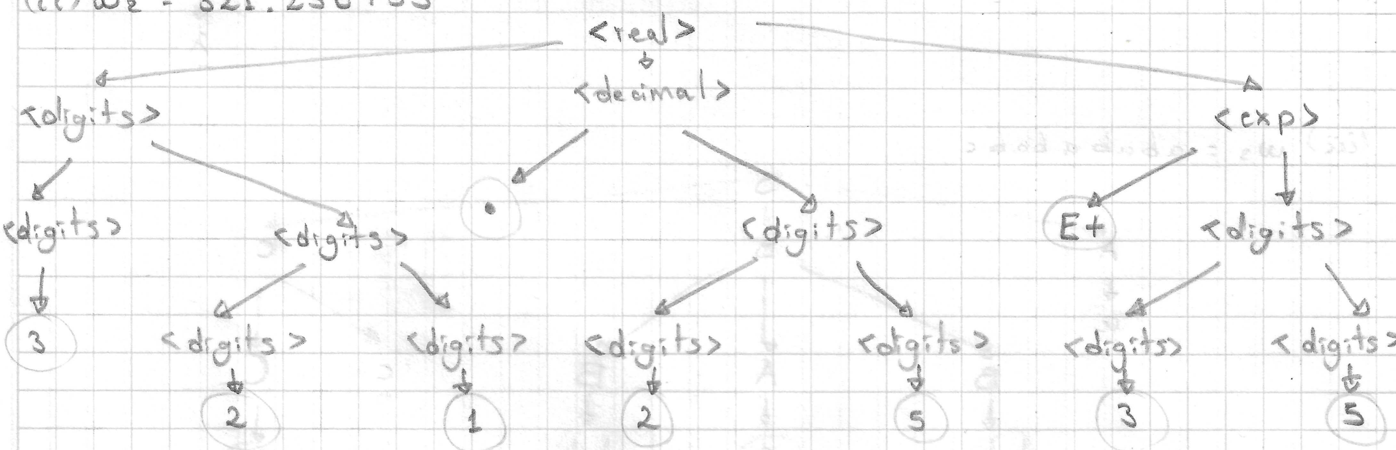
<exp> → E <digits> | E+ <digits> | E- <digits> | λ

Define the derivation tree for the following strings:

(i) $w_1 = 47.236$



(ii) $w_2 = 321.25E+35$

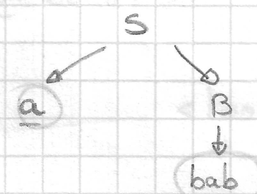


4. Be G a context-free grammar with the following productions:

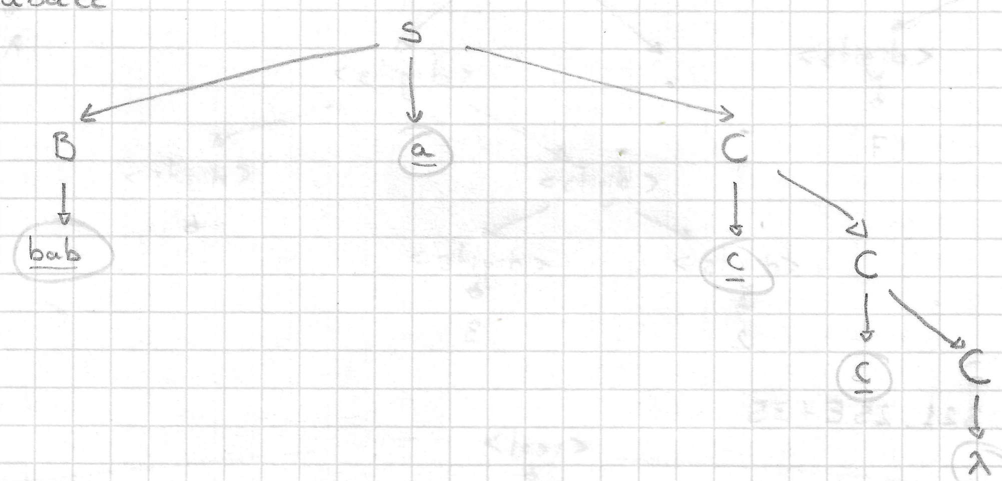
$$G = \begin{cases} S \to ABC \mid BaC \mid aB \\ A \to Aa \mid a \\ B \to BAB \mid bab \\ C \to cC \mid \lambda \end{cases}$$

Found derivation trees for the following strings:

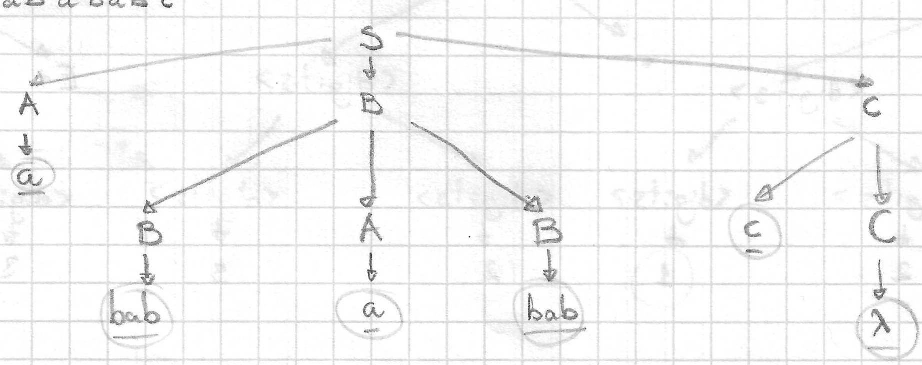
(i) $w_1 = abab$



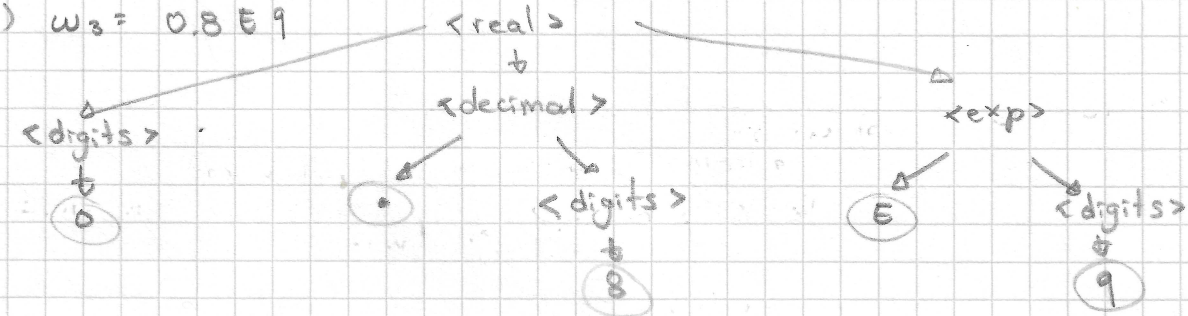
(ii) $w_2 = babacc$



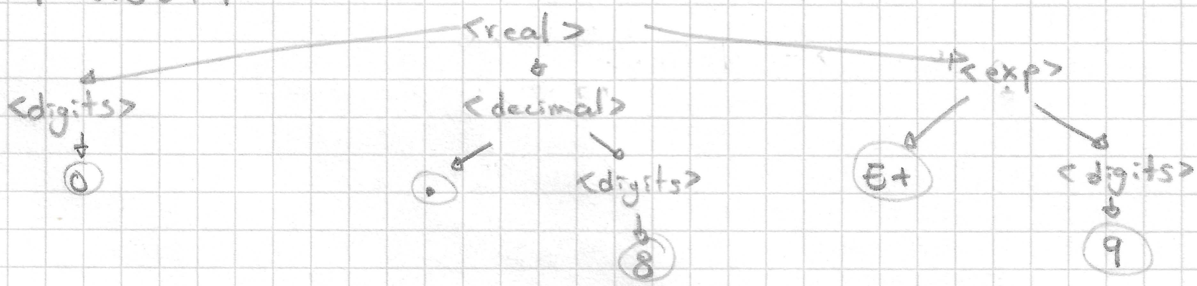
(iii) $w_3 = abababababc$



(iii) $w_3 = 0.8E9$



(iv) $w_4 = 0.8E+9$



6. As follows there is a contex-free grammar to generate identifiers, identifiers are strings of letters and digits, starting with a letter:

<identifier> → <letter><lsds>

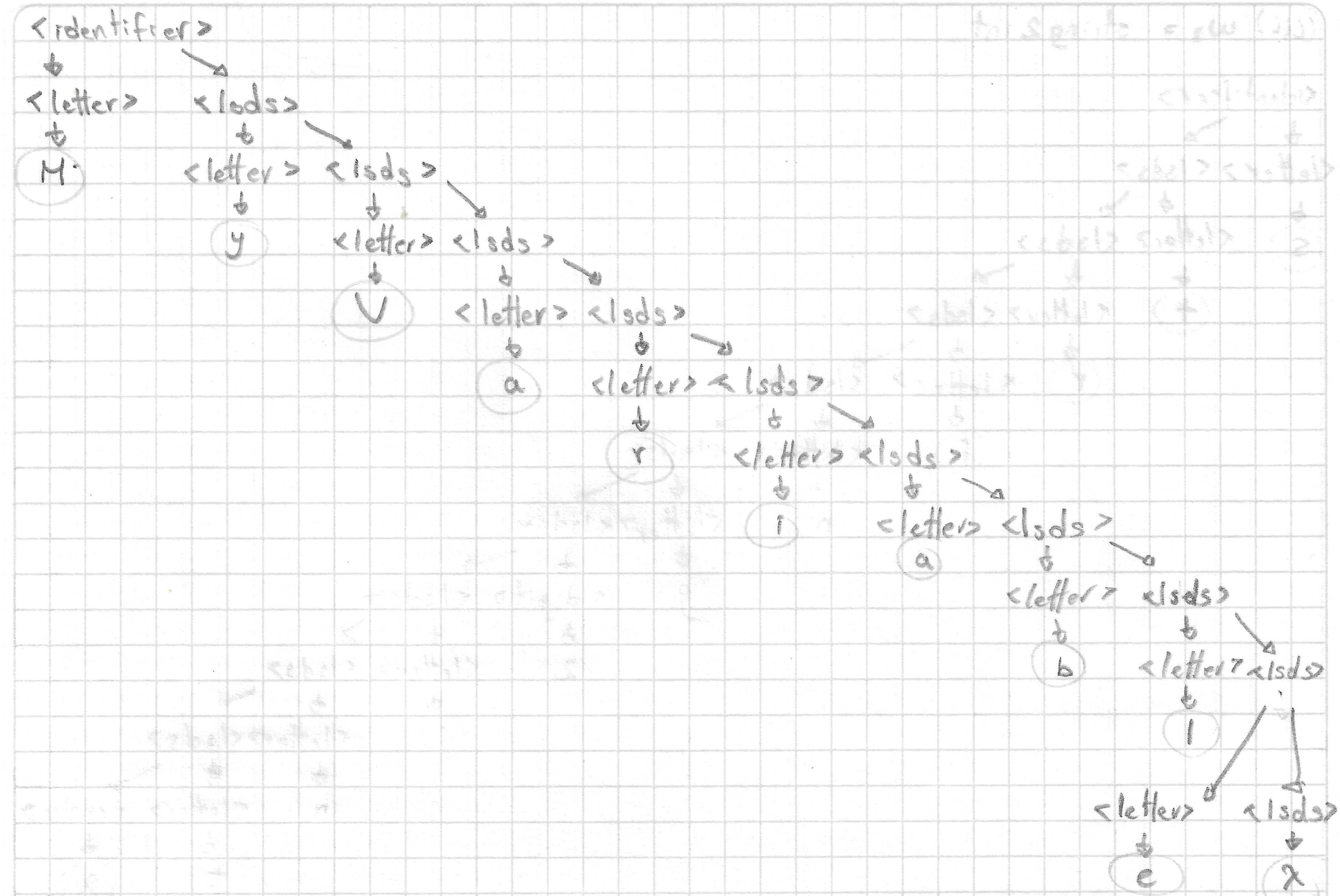
<lsds> → <letter><lsds> | <digit><lsds> | λ

<letter> → a | b | c | ... | x | y | z | A | B | C | ... | X | Y | Z

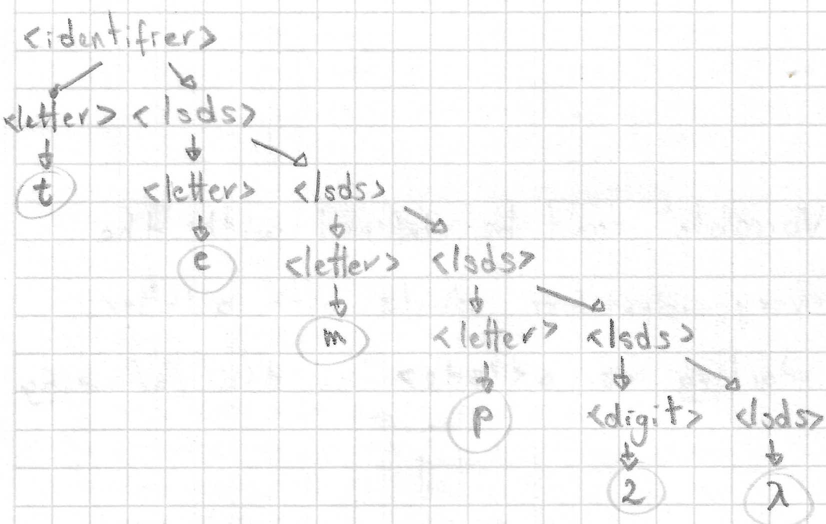
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Define the derivation tree for the following names:

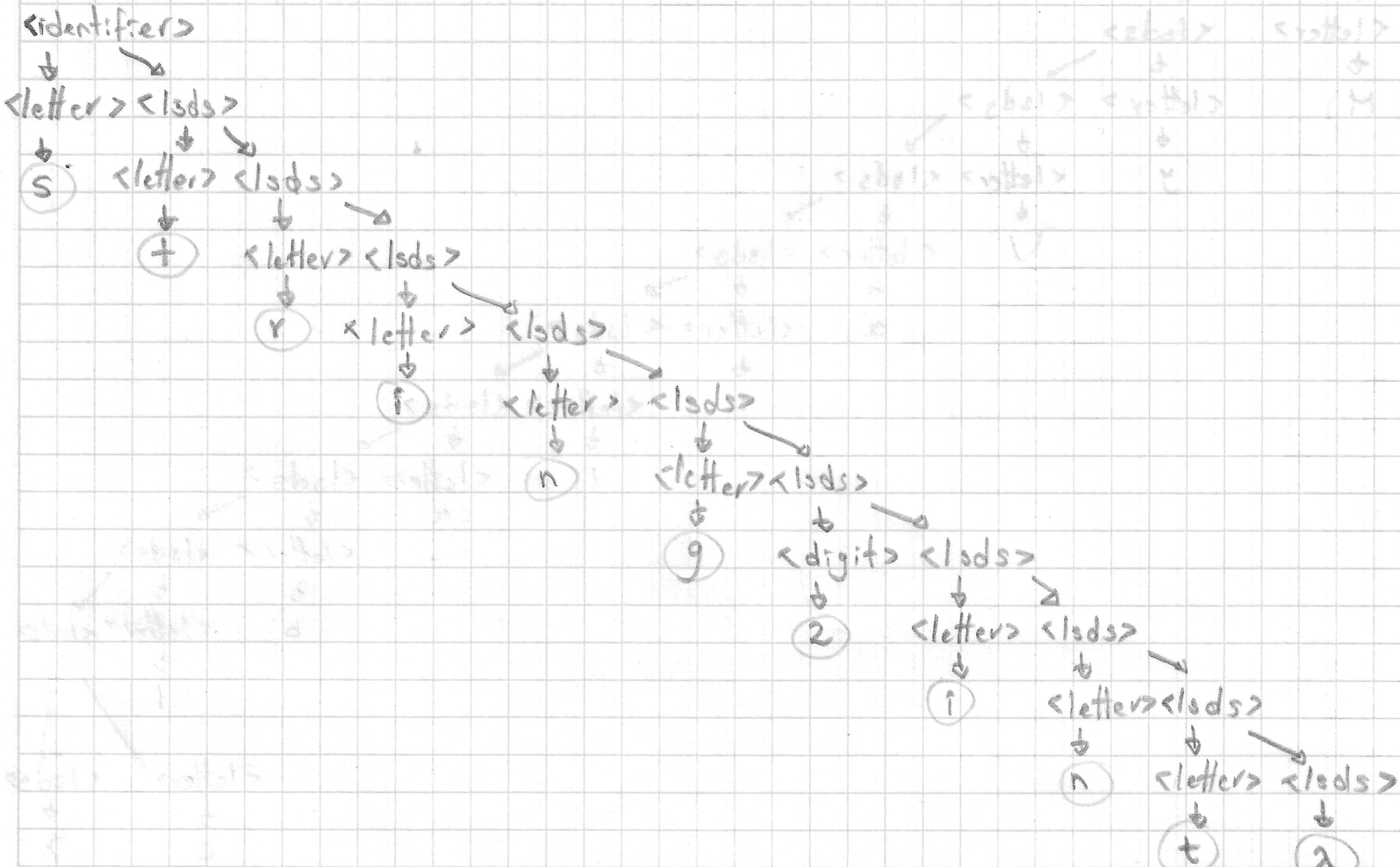
(i) w_1 = MyVariable


<identifier>
<letter> <lsds>
M y V a r i a b l e λ

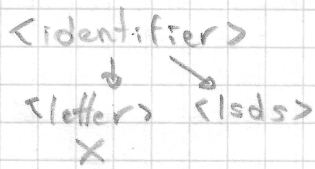
(ii) $w_2 = temp2$


<identifier>
<letter> <lsds>
<digit>
t e m p 2 λ

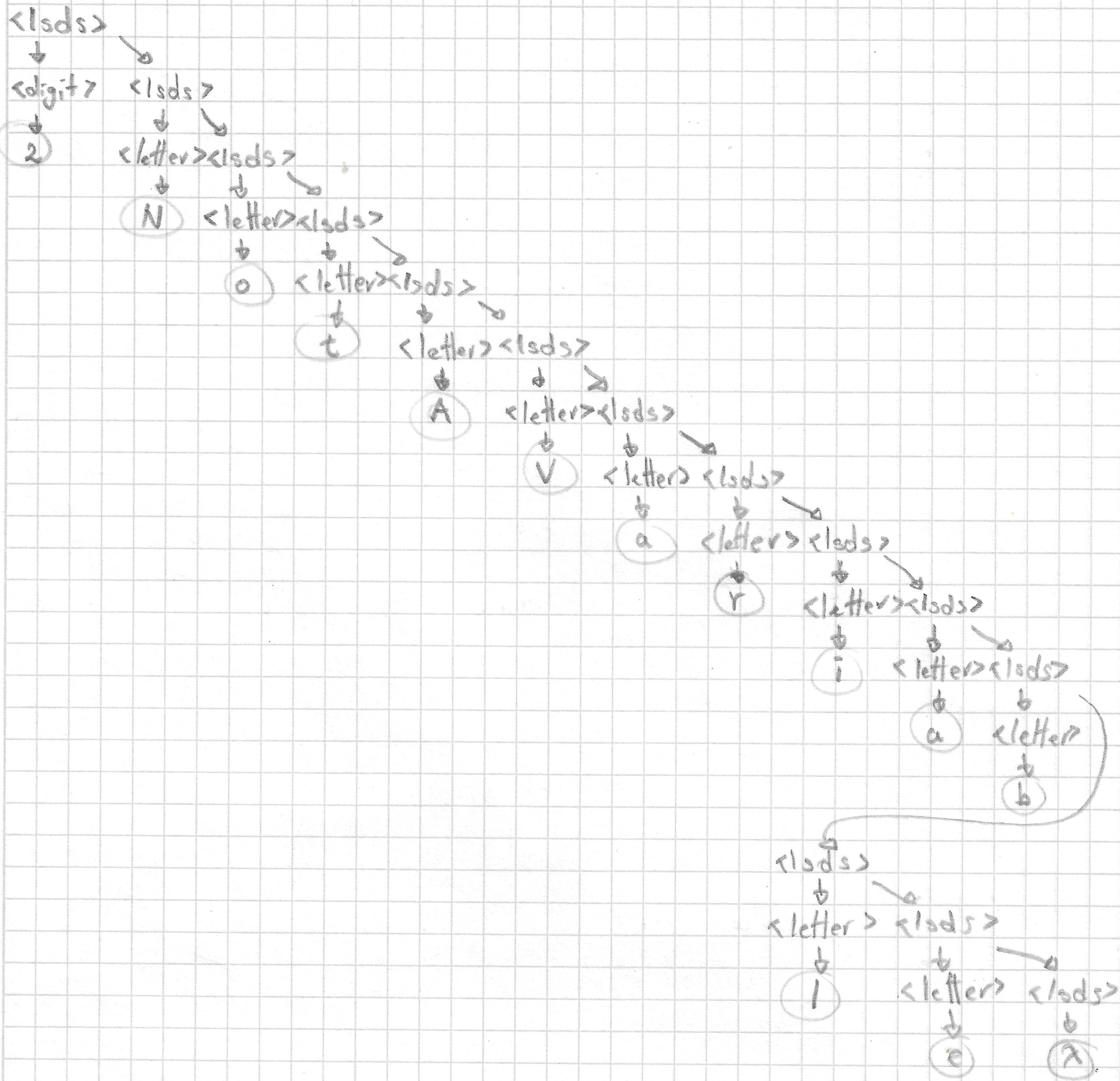
(iii) $w_3 =$ string2int



(iv) $w_4 =$ 2NotAVariable



The word w_4, "Not A Variable" can't be defined with the given grammar, because every identifier starts with a letter, but it can be defined starting as a <lsds> as the following tree


<lsds>
<digit>
<lsds>
2
<letter><lsds>
N
<letter><lsds>
o
<letter><lsds>
t
<letter><lsds>
A
<letter><lsds>
V
<letter><lsds>
a
<letter><lsds>
r
<letter><lsds>
i
<letter><lsds>
a
<letter>
b
<lsds>
<letter><lsds>
l
<letter><lsds>
e
λ

3. For each of the following regular expressions, define the corresponding generative grammar (all over the alphabet $\Sigma = \{a, b, c, d\}$):

(i) $\{ a^i b^j c^j d^i : i,j \geq 1 \}$.

$S \to ABCD$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to cC \mid c$

$D \to dD \mid d$

(ii) $\{ a^i b^i c^j d^j : i,j \geq 1 \}$

$S \to ABCD$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to cC \mid c$

$D \to dD \mid d$

(iii) $\{ a^i b^j c^j d^i : i,j \geq 1 \} \cup \{ a^i b^i c^j d^j : i,j \geq 1 \}$

Este es la union de ambos de los anteriores y se vería

$S \to ABCD \mid A'B'C'D'$

donde $A'B'C'D'$ son las del caso ii

(iv) $\{ a^i b^j c^{i+j} : i \geq 0, j \geq 1 \}$

$S \to ABC$

$A \to aA \mid \lambda$

$B \to bB \mid \lambda$

$C \to cC \mid c$