Computer Science III

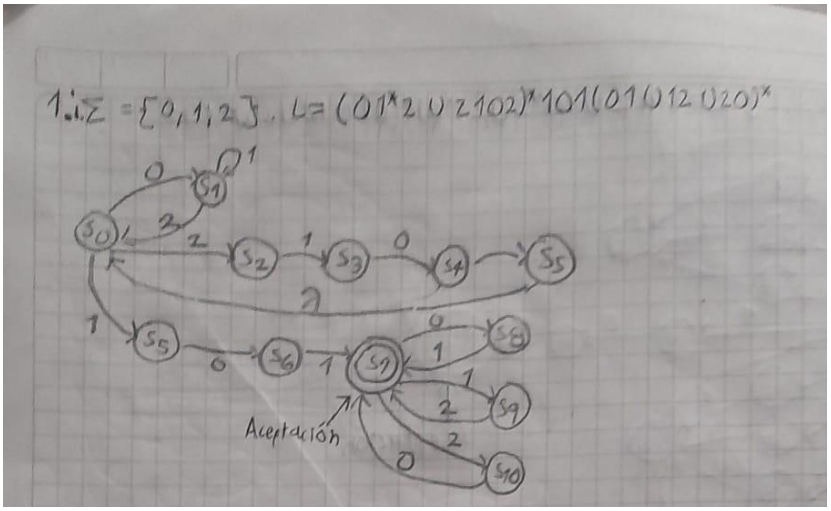
2024-III

WorkShop No. 1 — The Old Times

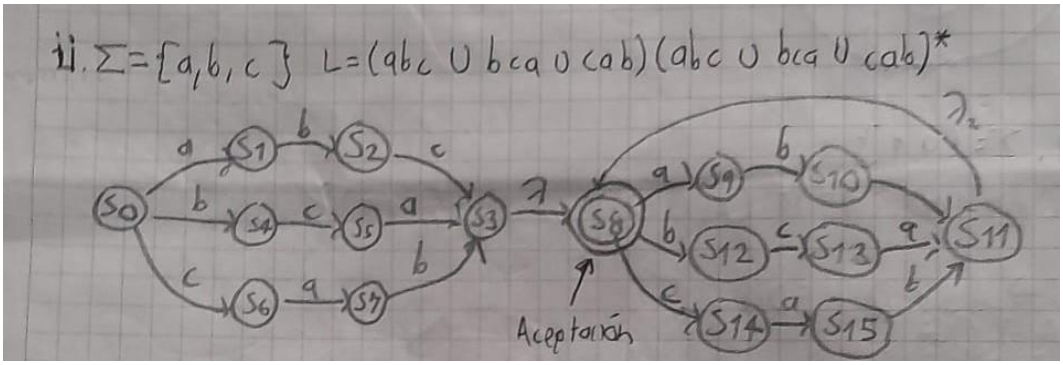
Miguel Angel Panqueva Pulido - 20201020174

1. For each of the following languages, define the corresponding **finite-state machine**:

- (i) Σ = {0, 1, 2}. *L* = (01*2 ∪ 2102)*101(01 ∪ 12 ∪ 20)*.

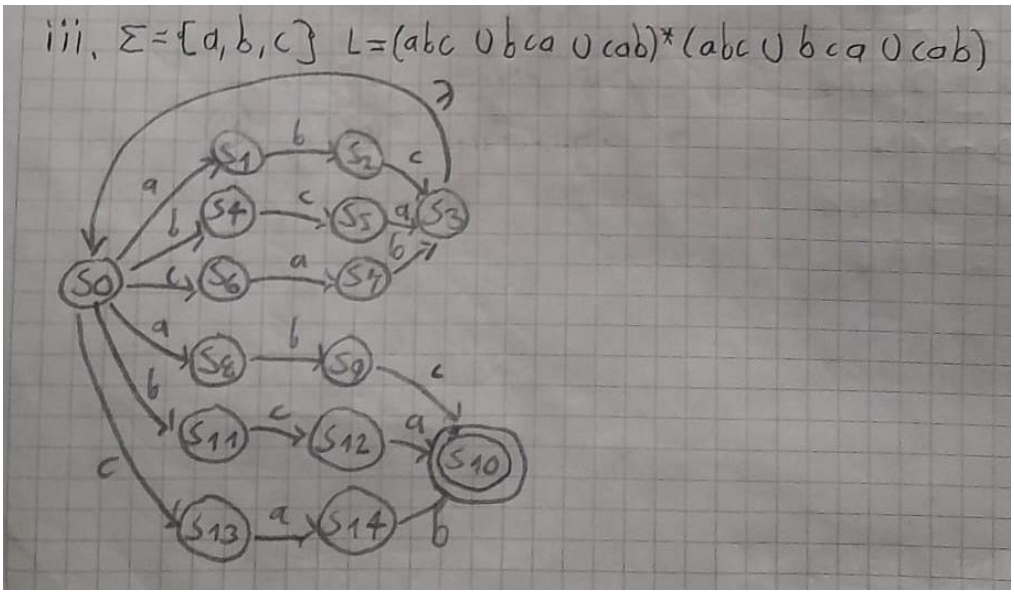


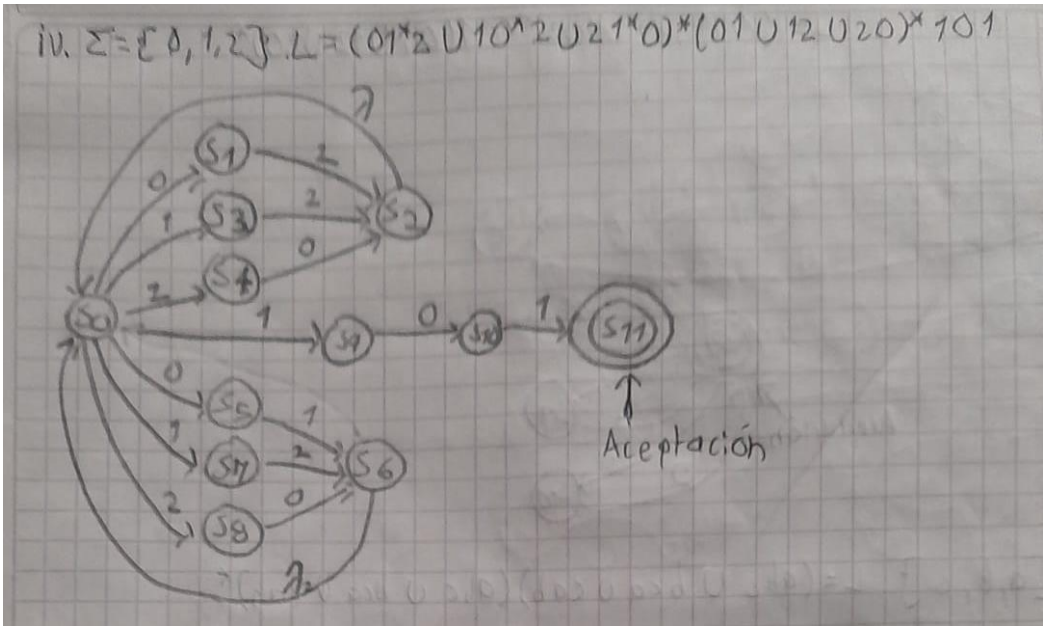
- (ii) Σ = {*a, b, c*}. *L* = (*abc* ∪ *bca* ∪ *cab*)(*abc* ∪ *bca* ∪ *cab*)*.



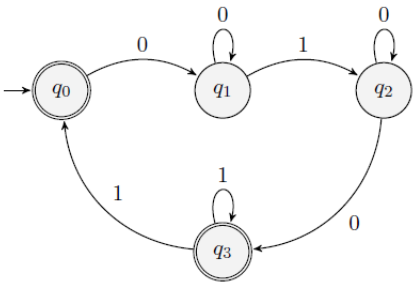
-

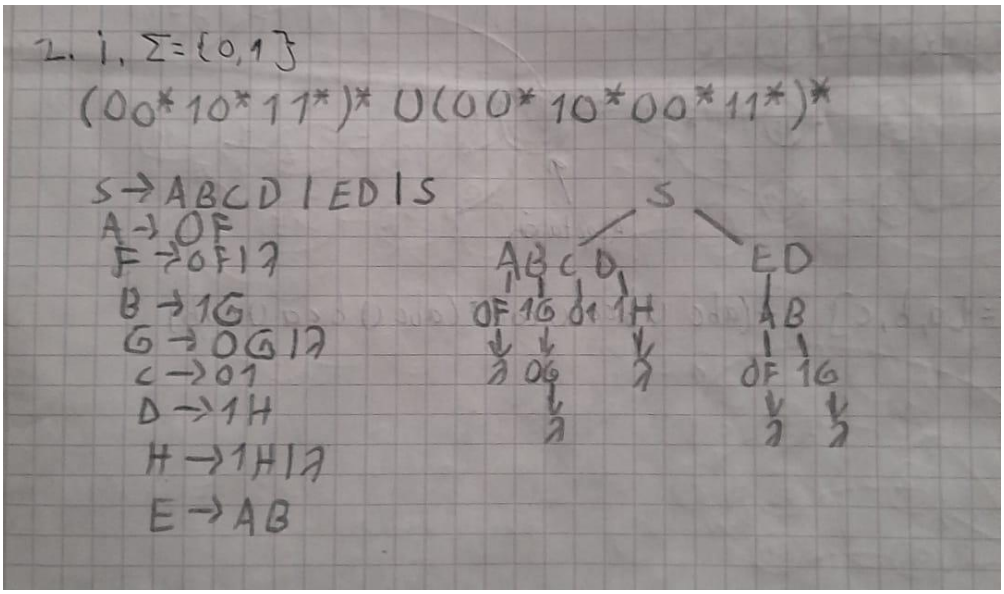
- (iii) Σ = {*a, b, c*}. *L* = (*abc* ∪ *bca* ∪ *cab*)*(*abc* ∪ *bca* ∪ *cab*).

- 

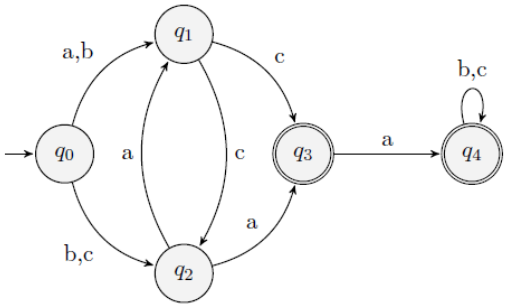
- (iv) Σ = {0, 1, 2}. *L* = (01*2 ∪ 10*2 ∪ 21*0)*(01 ∪ 12 ∪ 20)*101.
- 

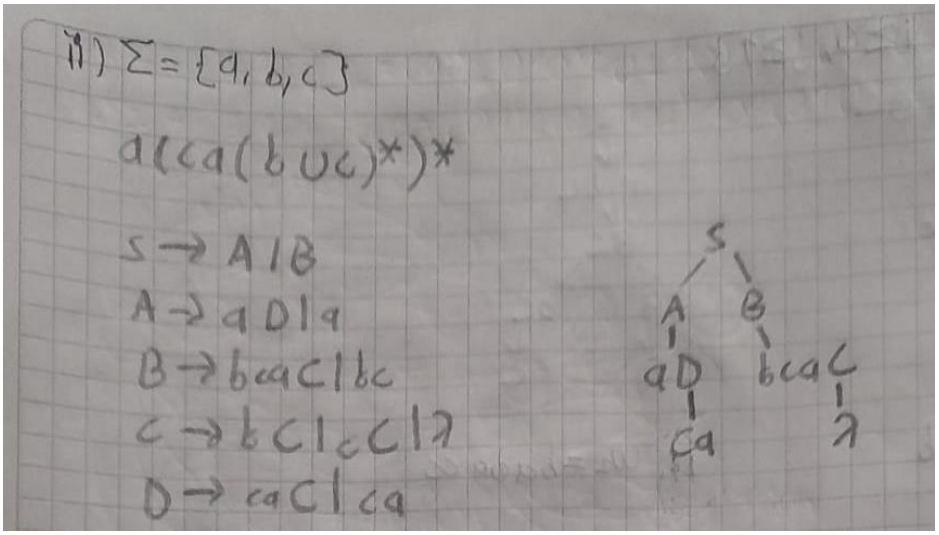
2. For each one of the following finite-state machines, define the corresponding regular expression and a generative grammar:
   - (i) Σ = {0, 1}.



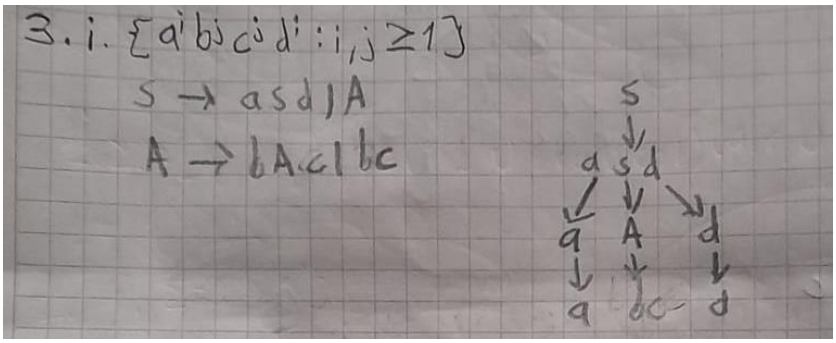


- (ii) Σ = {*a, b, c*}.

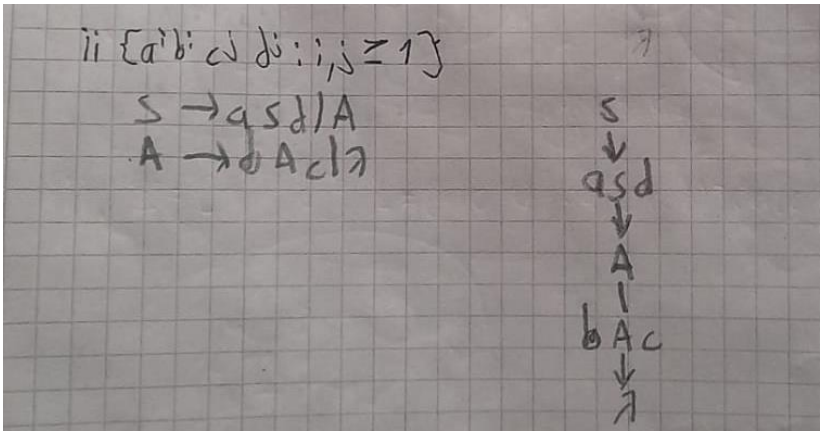




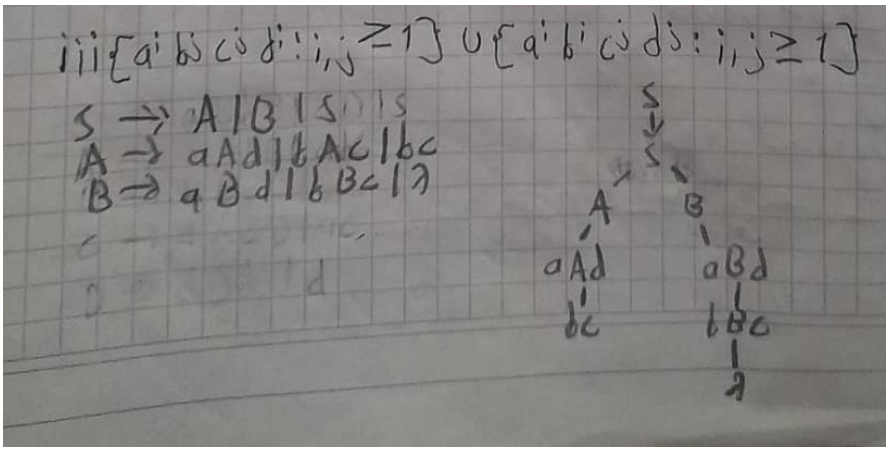
3. For each of the following regular expressions, define the corresponding generative grammar (all over the alphabet Σ = {a, b, c, d}):
   - (i) {*ai bj cj di* : *i, j* ≥ 1}.



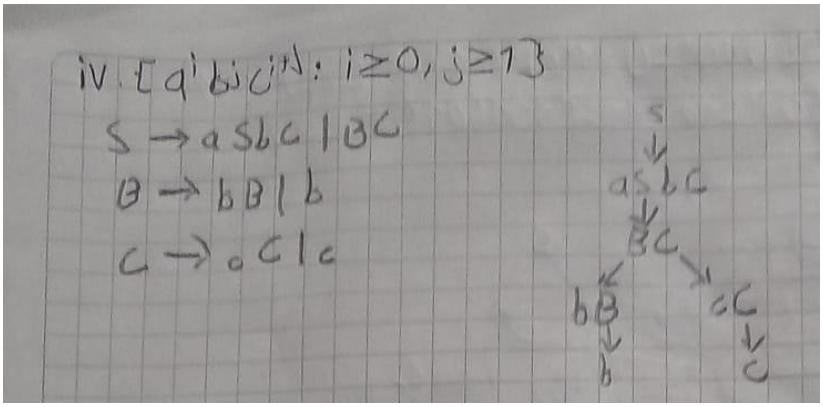
- (ii) {*ai bi cj dj* : *i, j* ≥ 1}.



- (iii) {*ai bj cj di* : *i, j* ≥ 1} ∪ {*ai bi cj dj* : *i, j* ≥ 1}.



- (iv) {*ai bj ci+j* : *i* ≥ 0, *j* ≥ 1}.

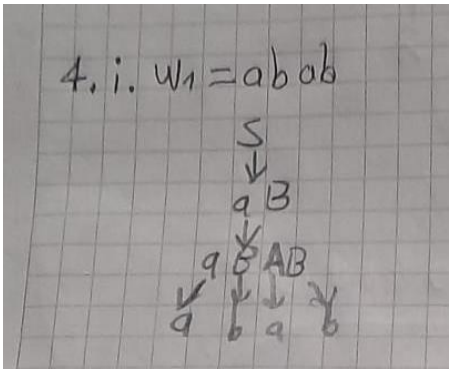


4. Be G a context-free grammar with the following productions:

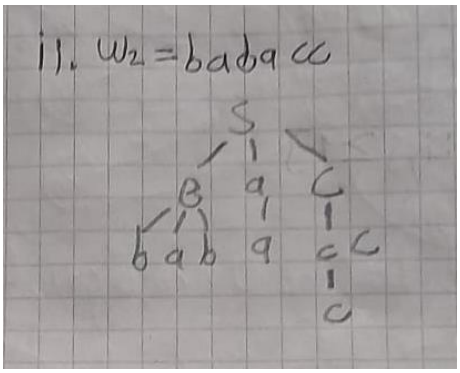
$$G = \begin{cases} S \to ABC \mid BaC \mid aB \\ A \to Aa \mid a \\ B \to BAB \mid bab \\ C \to cC \mid \lambda \end{cases}$$

Found derivation trees for the following strings:

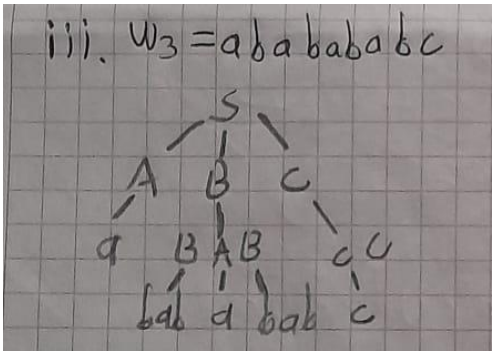
- (i) *w*1 = *abab*.



- (ii) *w*2 = *babacc*.



-
- (iii) *w*3 = *abababababc*.



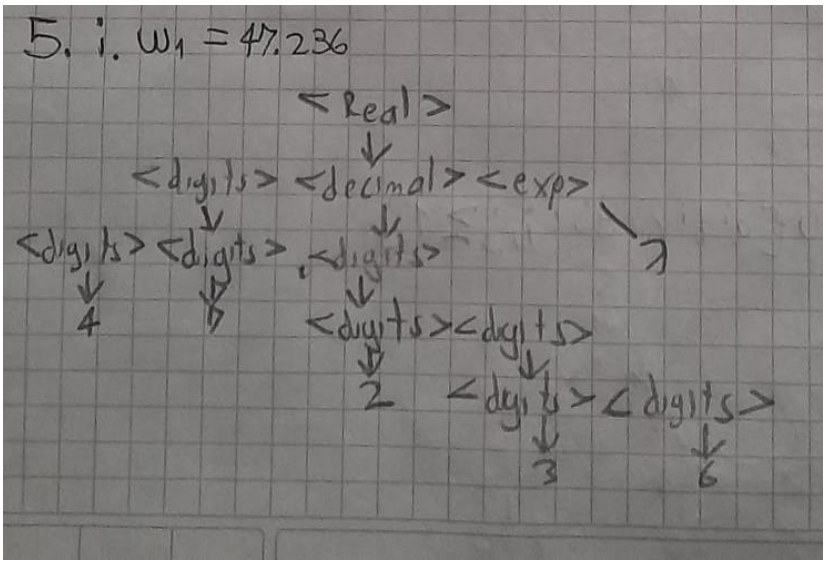
-

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

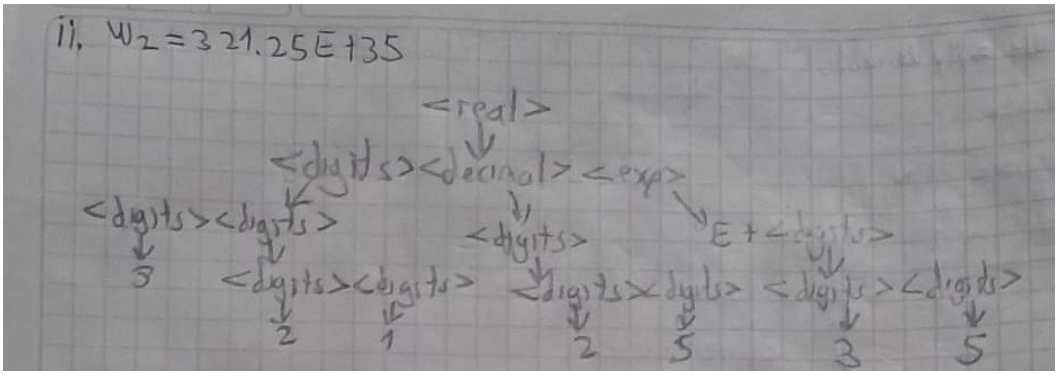
$$
\begin{array}{lcl}
< real > & \rightarrow & < digits >< decimal >< exp > \\
< digits > & \rightarrow & < digits >< digits > | \ 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
< decimal > & \rightarrow & < | digits > | \ \lambda \\
< exp > & \rightarrow & E < digits > | \ E+ < digits > | \ E- < digits > | \ \lambda
\end{array}
$$

Define the derivation tree for the following strings:

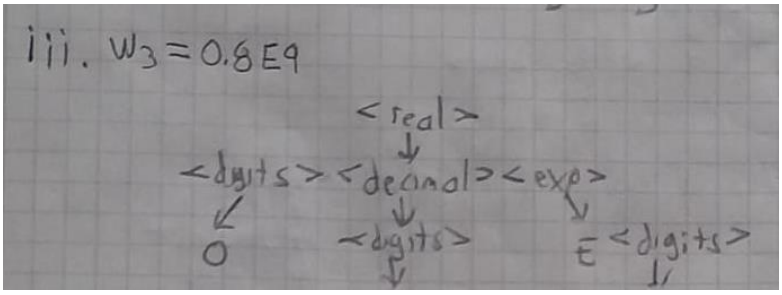
- (i) *w*1 = 47.236



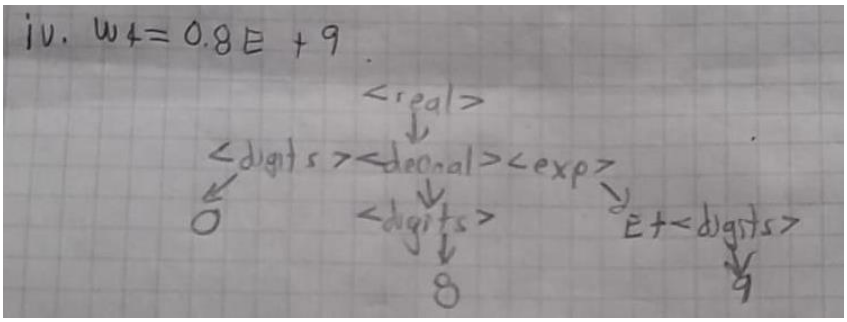
- (ii) *w*2 = 321.25*E* + 35



- (iii) w3 = 0.8E9



- (iv) w4 = 0.8E + 9

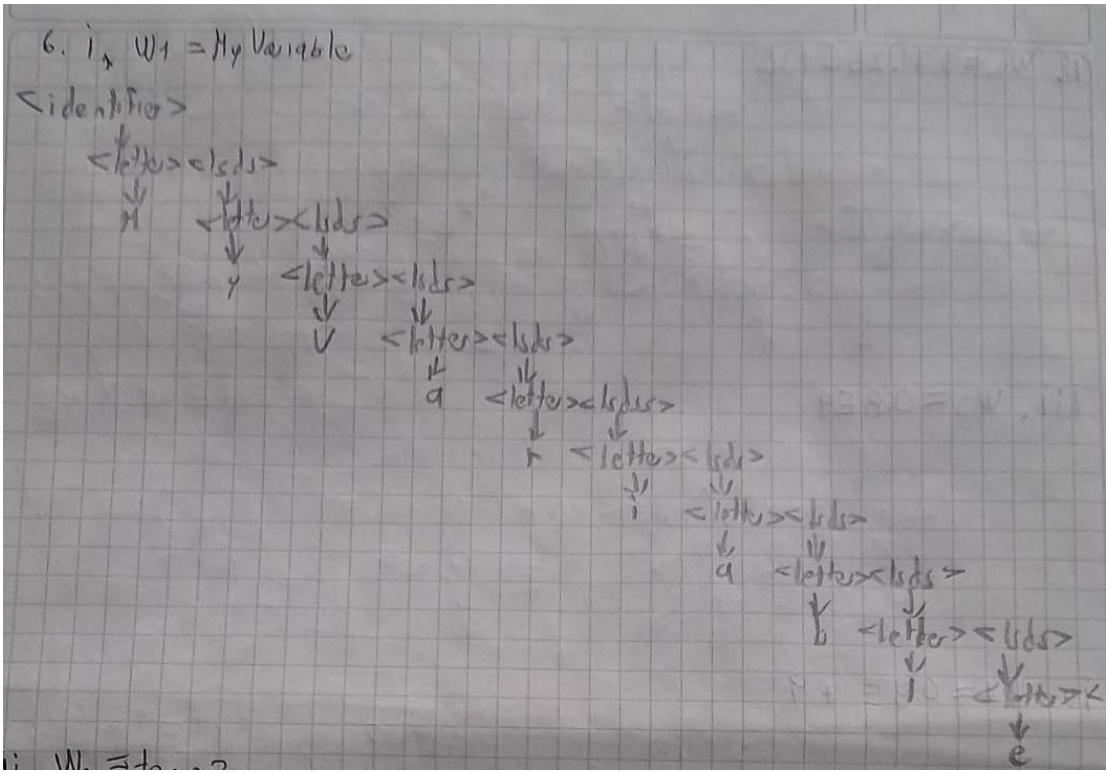


6. As follows there is a context-free grammar to generate identifiers, identifiers are strings of letters and digits, starting with a letter:

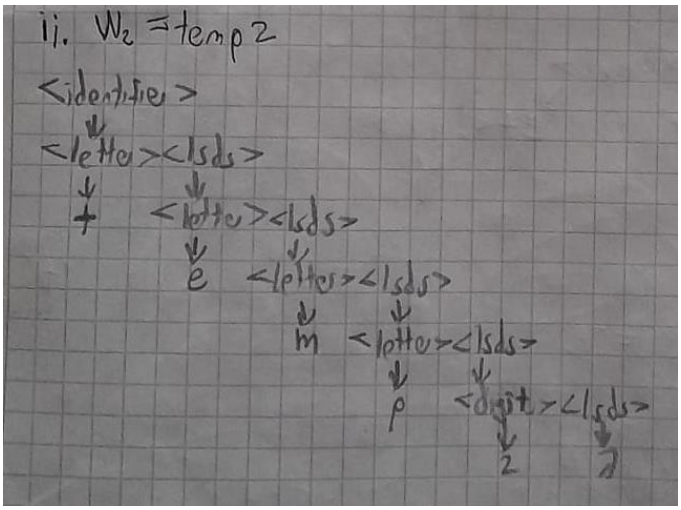
$$
\begin{array}{lcl}
< identifier > & \rightarrow & < letter >< lsds > \\
< lsds > & \rightarrow & < letter >< lsds > | < digit >< lsds > | \lambda \\
< letter > & \rightarrow & a \mid b \mid c \mid \ldots \mid x \mid y \mid z \mid A \mid B \mid C \mid \ldots \mid X \mid Y \mid Z \\
< digit > & \rightarrow & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{array}
$$

Define the derivation tree for the following names:

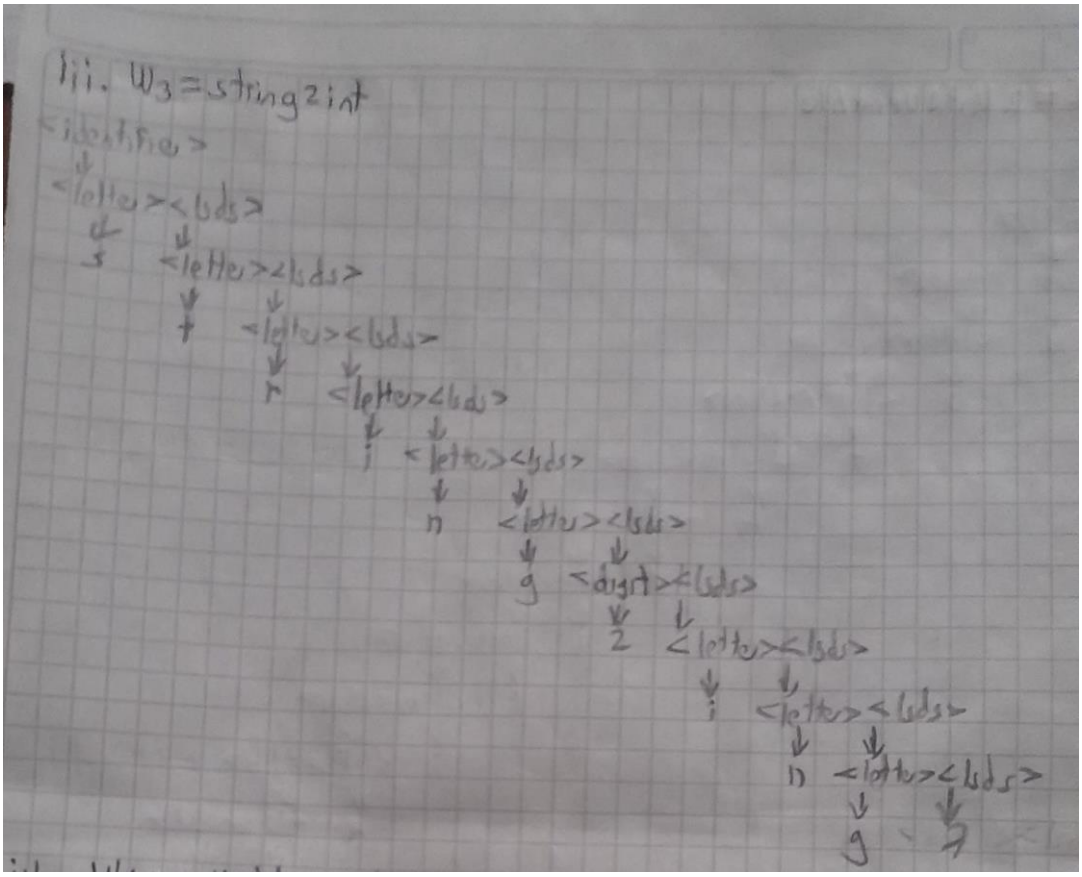
- (i) *w*1 = *MyVariable*



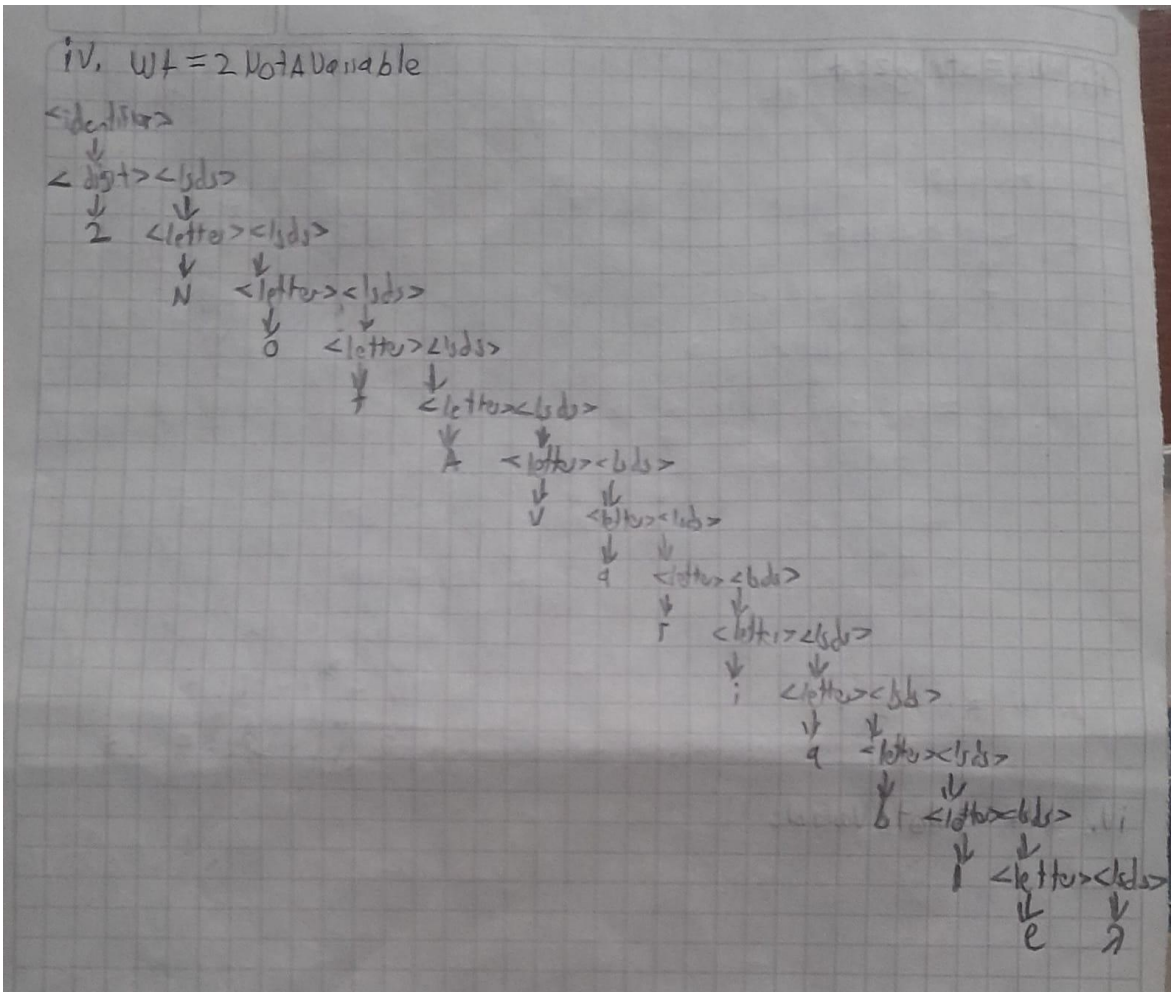
- (ii) *w*2 = *temp*2



- (iii) *w*3 = *string2int*



- (iv) *w*4 = 2*NotAVariable*


iv. Wt = 2 NotAVariable
<identifier>
<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><lsds>
b <letter><lsds>
l <letter><lsds>
e 2