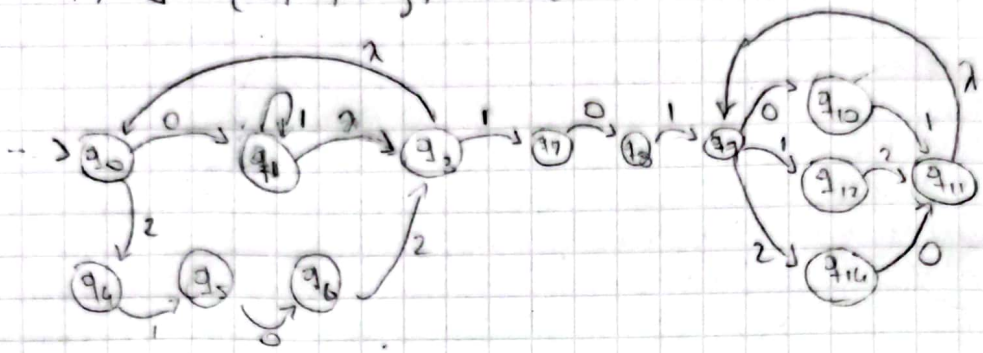
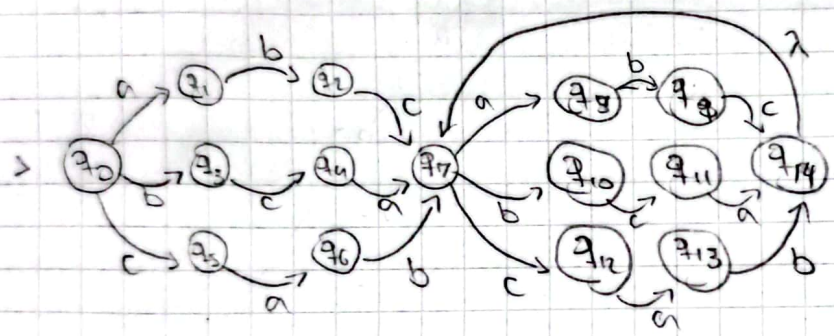
Workshop H1

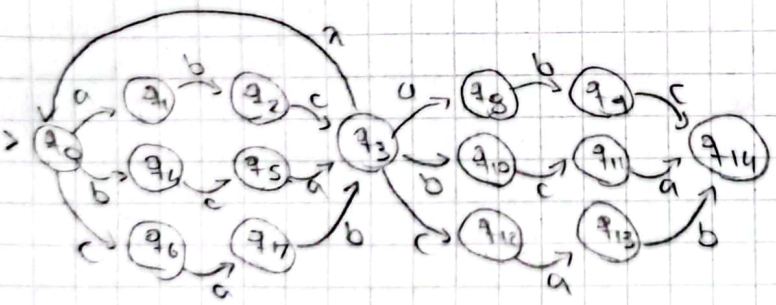
1. i) $\Sigma = \{0, 1, 2\}$, $L = (01^* \cup 2102)^* 101 (01 \cup 12 \cup 20)^*$



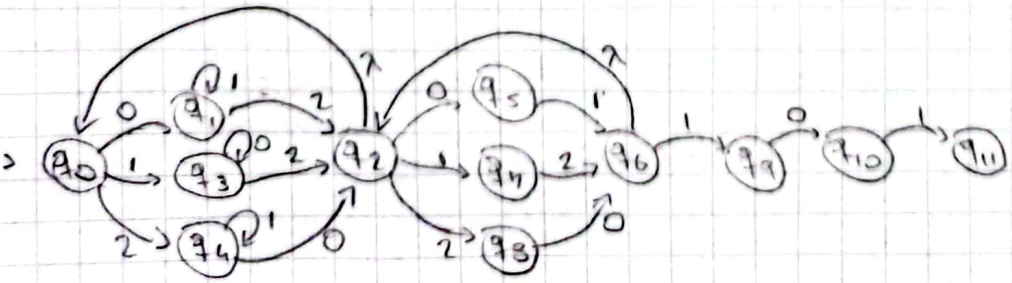
ii) $\Sigma = \{a, b, c\}$. $L = (abc \cup bca \cup cab)(abc \cup bca \cup cab)^*$



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

$L = (00^* | 0^* 01^* | 1)^*$

$$G = \begin{cases} S \to A\,S \mid \lambda \\ A \to 0\,B \\ B \to 0\,B \mid 1\,C \\ C \to 0\,C \mid 0\,D \\ D \to 1\,D \mid 1 \end{cases}$$

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

$L = [((a \cup b)c(a \cup c)^*) \cup ((b \cup c)a(a \cup c)^*)]\, a\, (b \cup c)^*$

$$G = \begin{cases} S \to X\,a\,Y \\ X \to A\,c\,Z \mid B\,a\,Z \\ A \to a \mid b \\ B \to b \mid c \\ Z \to a\,Z \mid c\,Z \mid \lambda \\ Y \to b\,Y \mid c\,Y \mid \lambda \end{cases}$$

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

$$G = \begin{cases} S \to a\,S\,d \mid a\,A\,d \\ A \to b\,A\,c \mid \lambda \end{cases}$$

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

$$G = \begin{cases} S \to A\,C \\ A \to a\,A\,b \mid a\,b \\ C \to c\,C\,d \mid c\,d \mid \lambda \end{cases}$$

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

$$G = \begin{cases} S \to A \mid E\,C \\ A \to a\,A\,d \mid a\,B\,d \\ B \to b\,B\,c \mid \lambda \\ E \to a\,E\,b \mid a\,b \\ C \to c\,C\,d \mid c\,d \mid \lambda \end{cases}$$

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

$$G = \begin{cases} S \to A\ B \\ A \to aABc \mid \lambda \\ B \to bBc \mid bc \mid \lambda \end{cases}$$

a.

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

i) $w_1 = abab$

$S \to aB$

$\downarrow$

$abab$

ii) $w_2 = babacc$

S

$\downarrow$

BaC

$\downarrow$

$babaC$

$\downarrow$

$babacC$

$\downarrow$

$babaccC \to babacc\lambda$

iii) $w_3 = ababababc$

S

$\downarrow$

ABC

$\swarrow$

aBC

$\downarrow$

$aBABC$

$\downarrow$

$aBaBC$

$\downarrow$

$ababaBC \to ababababC$

$ababababc\lambda$

$ababababcC$

5. <real> -> <digits> <decimal> <exp>
<digits> -> <digit> <digits> | <digit>
<digit> -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<decimal> -> . <digits> | λ
<exp> -> E <digits> | E + <digits> | E - <digits> | λ

i) $w_1 = 47.236$

<real>
<digits> <decimal> <exp>
<digit> <digits> <decimal> <exp>
4 <digit> <decimal> <exp>
7 <decimal> <exp>
. <digits> <exp>
2 <digits> <exp>
3 <digits> <exp>
6 <exp>
λ

ii) $w_2 = 321.25E+35$

<real>
<digits> <decimal> <exp>
3 <digits> <decimal> <exp>
2 <digits> <decimal> <exp>
1 <decimal> <exp>
. <digits> <exp>
2 <digits> <exp>
5 <exp>
E + <digits>
3 <digits>
5

iii) $w_3 = 0.8E9$

<real>
<digits> <decimal> <exp>
0 <decimal> <exp>
. <digits> <exp>
8 <exp>
E <digits>
9

iv) $w_4 = 0.8E+9$

<real>
<digits> <decimal> <exp>
0 <decimal> <exp>
. <digits> <exp>
8 <exp>
E + <digits>
9

6. <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

i) w1 = MyVariable

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

ii) w2 = temp2

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

iii) w3 = string2Int

<identifier>
<letter><lsds>
s <letter><lsds>
t <letter><lsds>
r <letter><lsds>
i <letter><lsds>
n <letter><lsds>
g <digit><lsds>
2 <letter><lsds>
I <letter><lsds>
n <letter><lsds>
t <lsds>
↓
λ

iv) w4 = 2 Not A Variable

No se puede porque <identifier> -> <letter> <lsdo> obliga a empezar con un <letter>

7. i) Identifier

r'[a-zA-Z_][a-zA-Z0-9_]*'

ii) Integer Literal

r'[+-]?\d+'

iii) Floating Point Literal

r'[+-]?(\d+\.\d*|\.\d+)([eE][+-]?\d+)?'

iv) String Literal

r'"([^"\\]|\\.)*"'

v) Single-line Comment

r'//.*'

vi) Multi-line Comment

r'/*.*?*/'

vii) White space

r'\s+'

viii) Operators

r'==|!=|<=|>=|<|>|\+|-|*|/|='

ix) Keywords

r'\b(if|else|while|return)\b'

x) Hexadecimal Literal

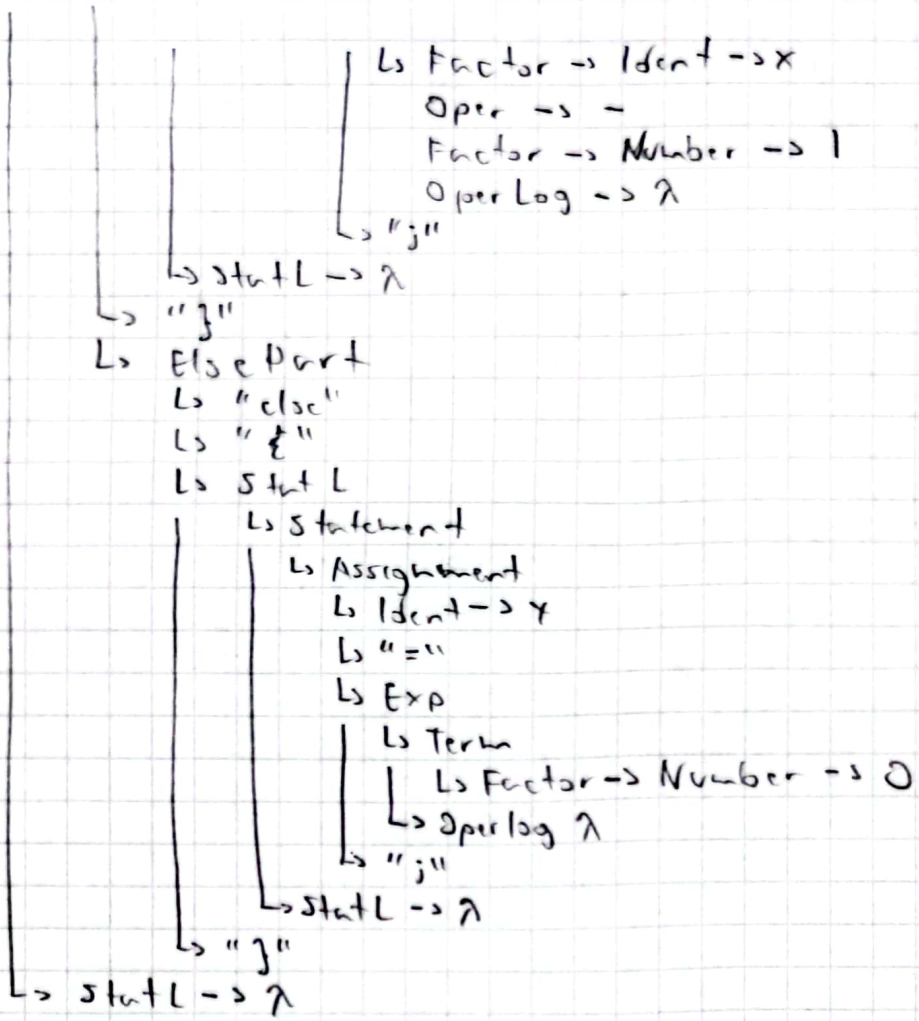
r'0[xX][0-9a-fA-F]+'

3. a) `x = 5 + 3 * 2;`

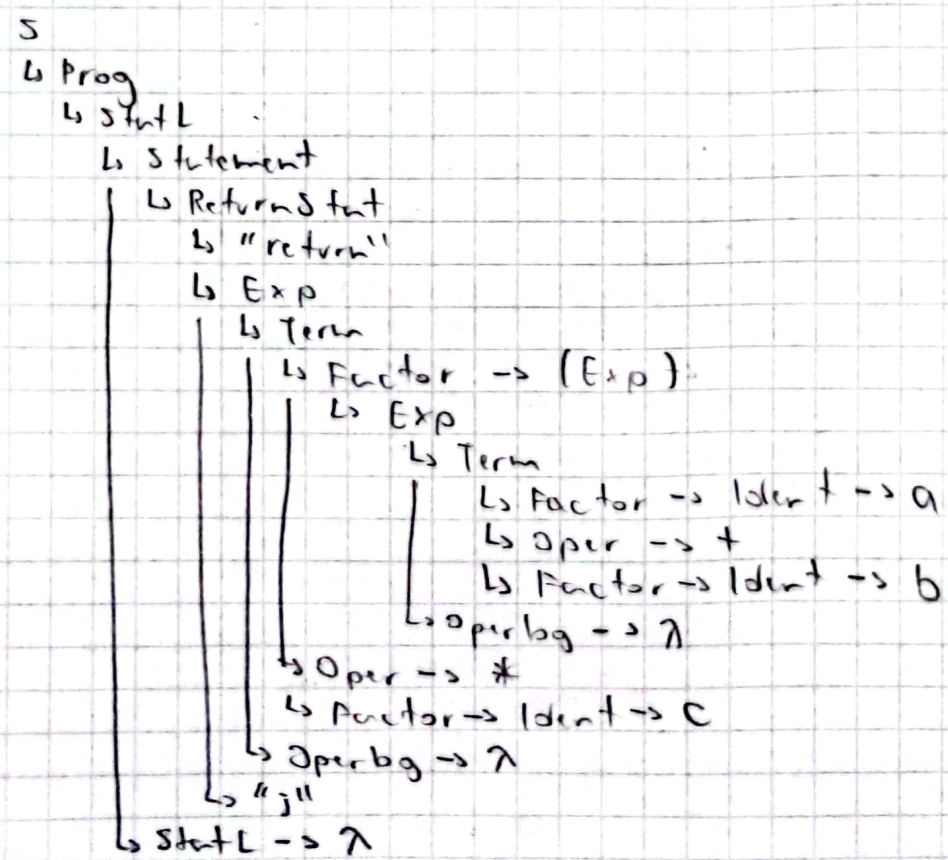
S
- Prog
  - StatL
    - Statement
      - Assignment
        - Ident -> x
        - "="
        - Exp
          - Term
            - Factor -> Number -> 5
            - Oper -> +
            - Factor
              - Number -> 3
              - Oper -> *
              - Number -> 2
          - OperLog -> λ
        - ";"
      - StatL -> λ

b) `if(x>0){y=x-1;}else{y=0;}`

S
- Prog
  - StatL
    - Statement
      - IfStat
        - "if"
        - "("
        - Exp
          - Term
            - Factor -> Ident -> x
            - Oper -> >
            - Factor -> Number -> 0
          - OperLog -> λ
        - ")"
        - "{"
        - StatL
          - Statement
            - Assignment
              - Ident -> y
              - "="
              - Exp
                - Term
                  - 1



g) return (a+b) * c j



c) while (x < 10) { x = x + 1 ; }