EJEMPLO 1

1 3 1 2 2 → 9

+ + + +

1 → 1

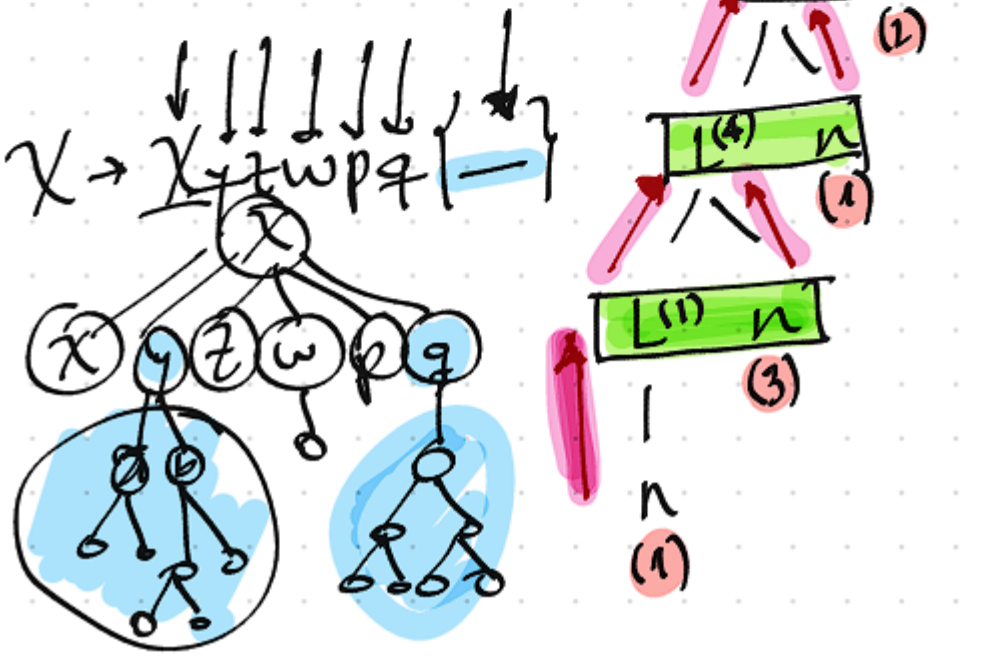
3 4 → 7

Regla Semántica

$S \rightarrow L$ {print (L.valor)}

$L \rightarrow L_1 n$ {L.valor = L_1.valor + val(n)}

$| n$ {L.valor = val(n)}



$S \to E$ { print(E.val) }

$E \to T$ { E'.val = T.val } E' { E.val = E'.val }

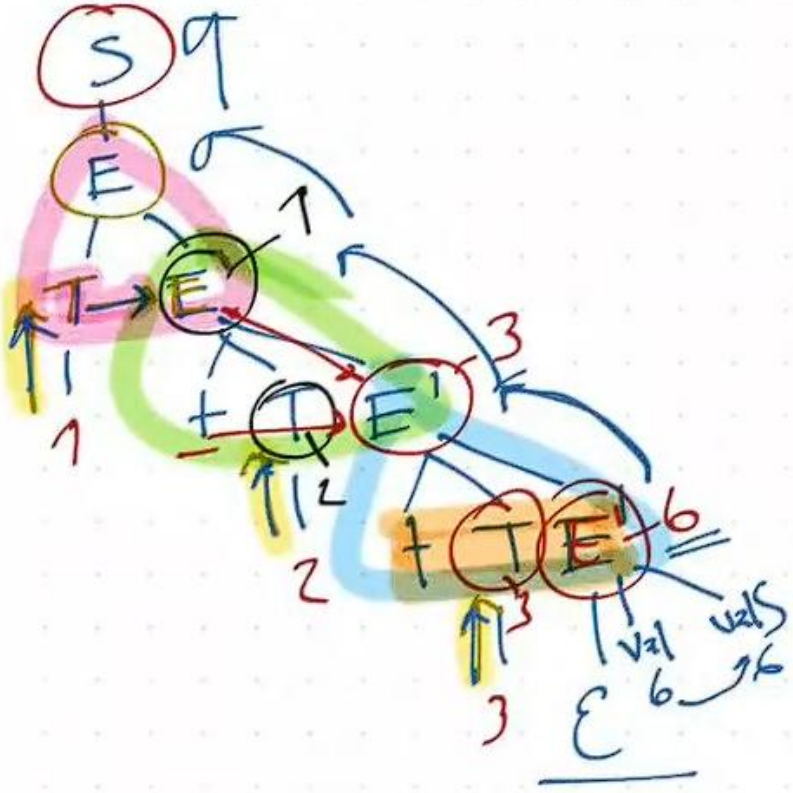
$E' \to +T$ { E_1'.val = E'.val + T.val } E' { E'.val = E_1'.val }

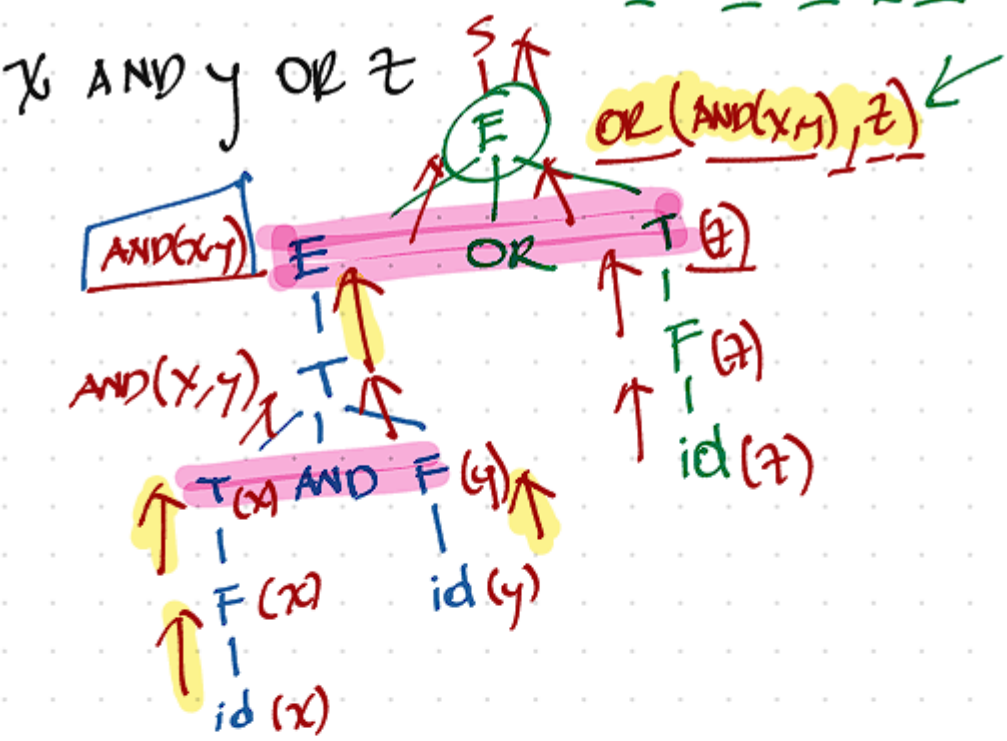
$\quad | \varepsilon$ { E'.val = E'.val }

$T \to$ num { T.val = num }

$\quad | (E)$

1+2+3 = 6





$S \rightarrow E$ {print(E.cadena)} Concat("OR(", E.cadena, ",", T.cadena, ")")

$E \rightarrow E$ OR T {E.cadena =

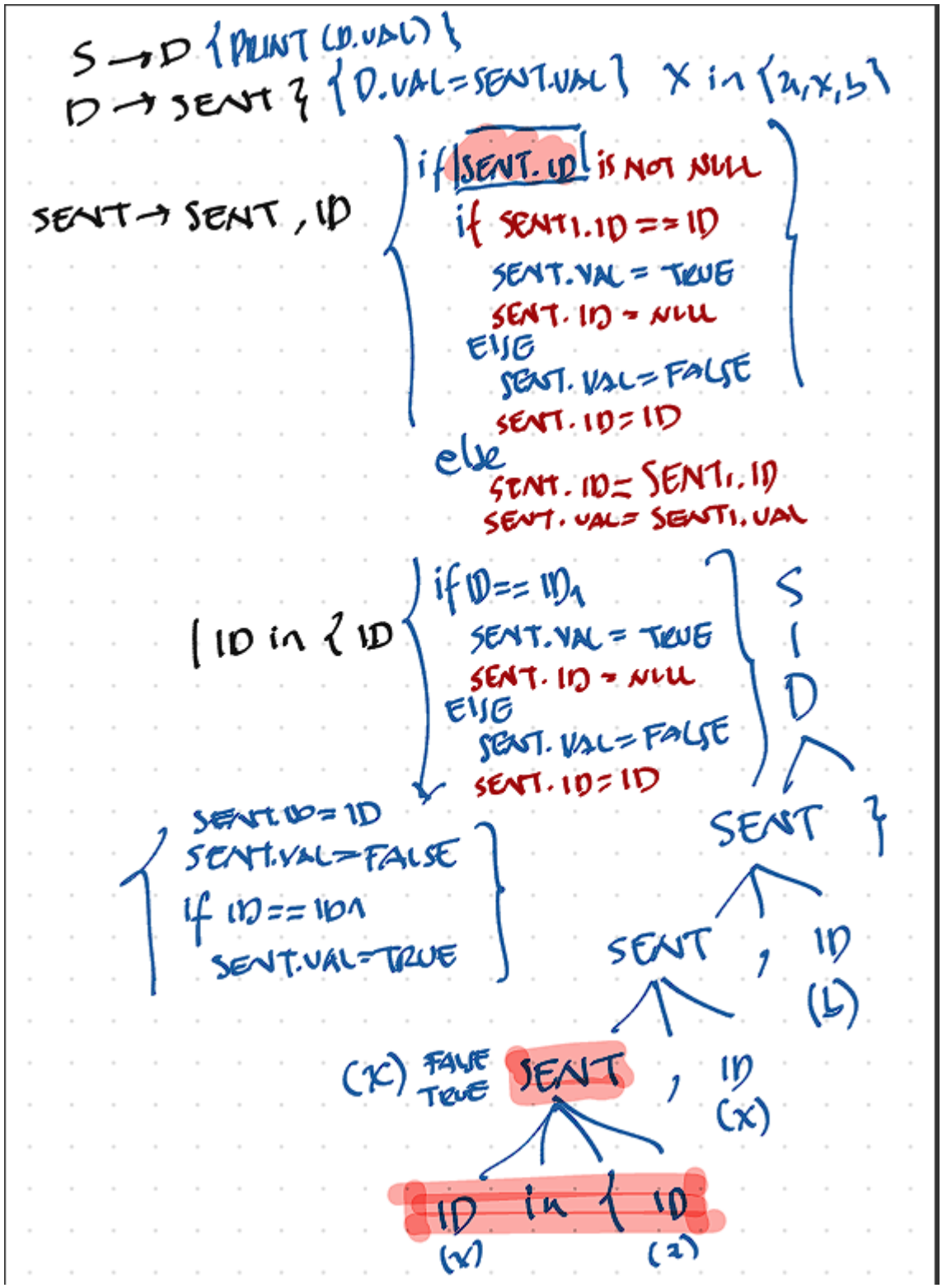
| T {E.cadena = T.cadena}

$T \rightarrow T$ AND F {T.cadena = Concat("AND(", T.cadena, ",", F.cadena, ")")}

| F {T.cadena = F.cadena}

$F \rightarrow$ id {F.cadena = id}

| (E) {F.cadena = E.cadena}


S → D { PRINT (D.VAL) }
D → SENT { D.VAL = SENT.VAL } X in {a, x, b}
SENT → SENT , ID
if SENT.ID is NOT NULL
if SENT1.ID == ID
SENT.VAL = TRUE
SENT.ID = NULL
ELSE
SENT.VAL = FALSE
SENT.ID = ID
else
SENT.ID = SENT1.ID
SENT.VAL = SENT1.VAL
| ID in { ID
if ID == ID1
SENT.VAL = TRUE
SENT.ID = NULL
ELSE
SENT.VAL = FALSE
SENT.ID = ID
SENT.ID = ID
SENT.VAL = FALSE
if ID == ID1
SENT.VAL = TRUE
S
I
D
SENT }
SENT
,
ID
(b)
(x) FALSE TRUE
SENT
,
ID
(x)
ID
in
{
ID
(x)
(a)

$S \rightarrow E$ { if E.op == "OR" print("OR(", E.val, ")") if E.op == "AND" print("AND(", E.val, ")") }

$E \rightarrow E$ OR T {
if E1.op == "AND"
E1.val = Concat("AND(", E1.val, ")")
if T.op == "AND"
T.val = Concat("AND(", T.val, ")")
E.val = concat(E1.val, ";", T.val)
E.op = "OR"
}

| T {E.val = T.val; E.op = T.op}

$T \rightarrow T$ AND F {
if T1.op == "OR"
T1.val = concat("OR(", T1.val, ")")
if F.op == "OR"
F.val = concat("OR(", F.val, ")")
T.val = concat(T1.val, ";", F.val)
T.op = "AND"
}

| F {T.val = F.val; T.op = F.op}

$F \rightarrow$ id {F.val = id; F.op = "id"}

| (E) {F.val = E.val; F.op = E.op}

Ejemplo

x in {a, b, x}

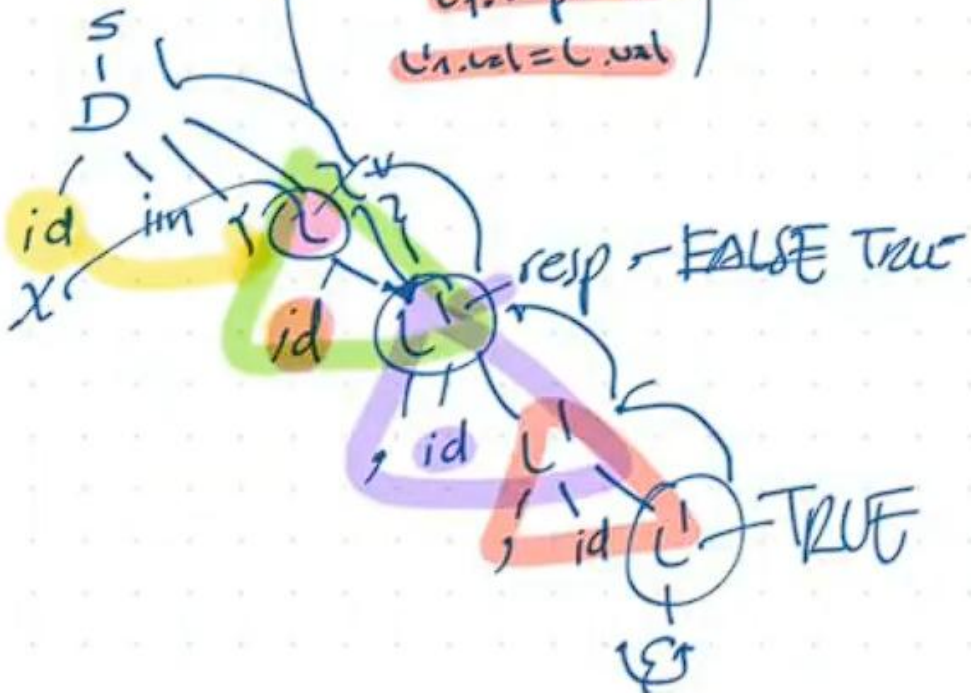
S → D { print(D.resp$) }

D → id in { { L.val = id } L } { D.resp$ = L.resp$ }

L → id { L'.resp = FALSE; if L.val == id then L'.resp = TRUE; L'.val = L.val } { L' { L.resp$ = L'.resp$ } }

L' → , id { if L'.res == FALSE { L'1.resp = FALSE; if L'.val == id then L'1.resp = TRUE; L'1.val = L.val } } L' { L'.resp$ = L'1.resp$ }

L' → ε { L'.resp$ = L'.resp }



(1,2) NOEE

$S \rightarrow MOV$ {PRINT("(", MOV.x, ",", MOV.y, ")")}

$MOV \rightarrow MOV\ DIR$ { MOV.x = MOV1.x + DIR.x ; MOV.y = MOV1.y + DIR.y }

$| (num, num)\ DIR$ { MOV.x = num + DIR.x ; MOV.y = num + DIR.y }

$DIR \rightarrow N$ {DIR.x = 0 DIR.y = 1}

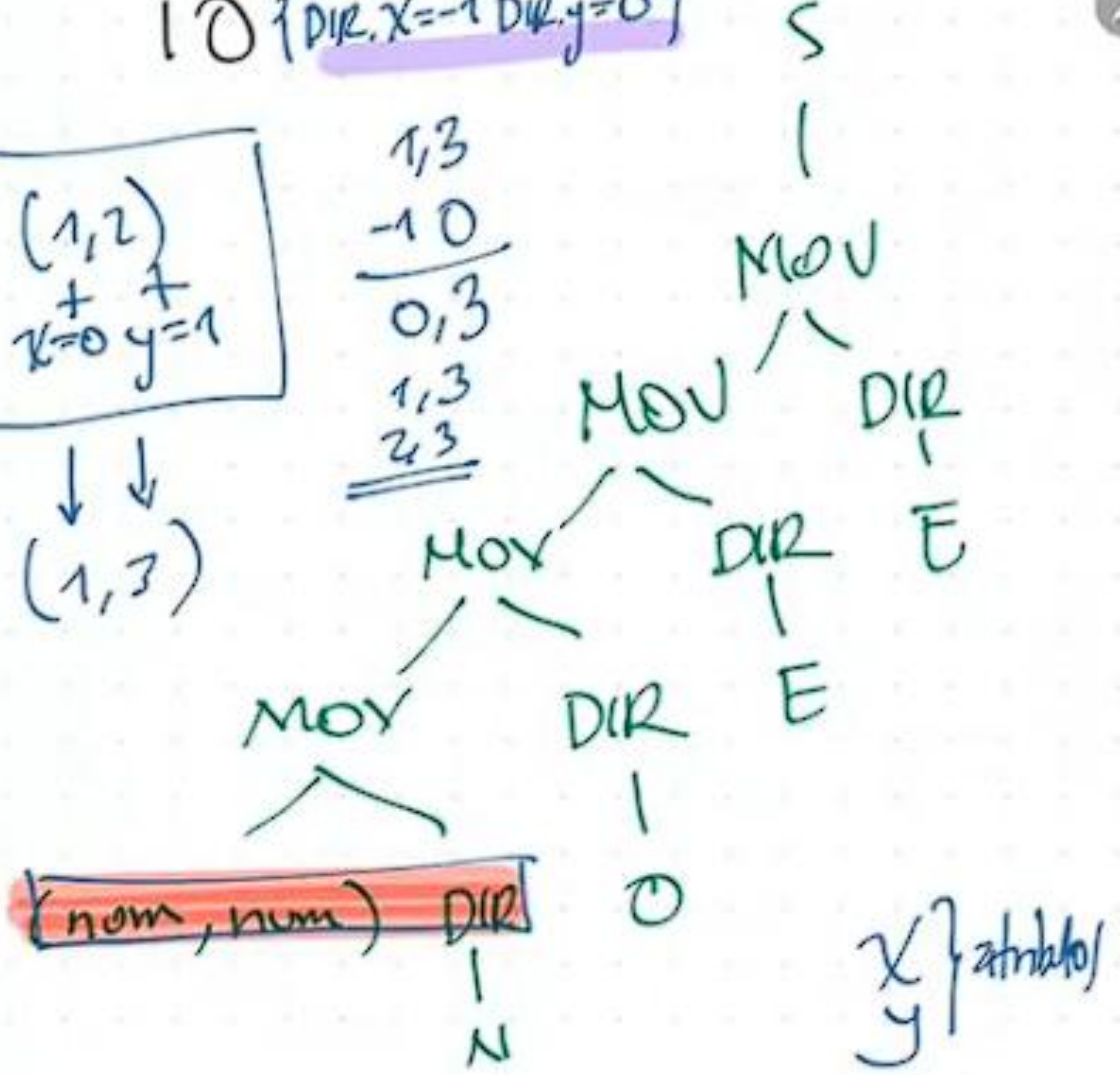
$| S$ {DIR.x = 0 DIR.y = -1}

$| E$ {DIR.x = 1 DIR.y = 0}

$| O$ {DIR.x = -1 DIR.y = 0}

(1,2) + x=0 y=1 + (1,3)

1,3
-1 0
0,3
1,3
2 3



ESQUEMA

(1,2) NOEE

S → MOV { PRINT("(", MOV.xs, ",", MOV.ys, ")") }

MOV → (num,num) DIR { MOV'.x = num + DIR.x ; MOV'.y = num + DIR.y } MOV' { MOV.xs = MOV'.xs ; MOV.ys = MOV'.ys }

MOV' → DIR { MOV'.x = MOV'.x + DIR.x ; MOV'.y = MOV'.y + DIR.y } MOV' { MOV'.xs = MOV'.xs ; MOV'.ys = MOV'.ys }

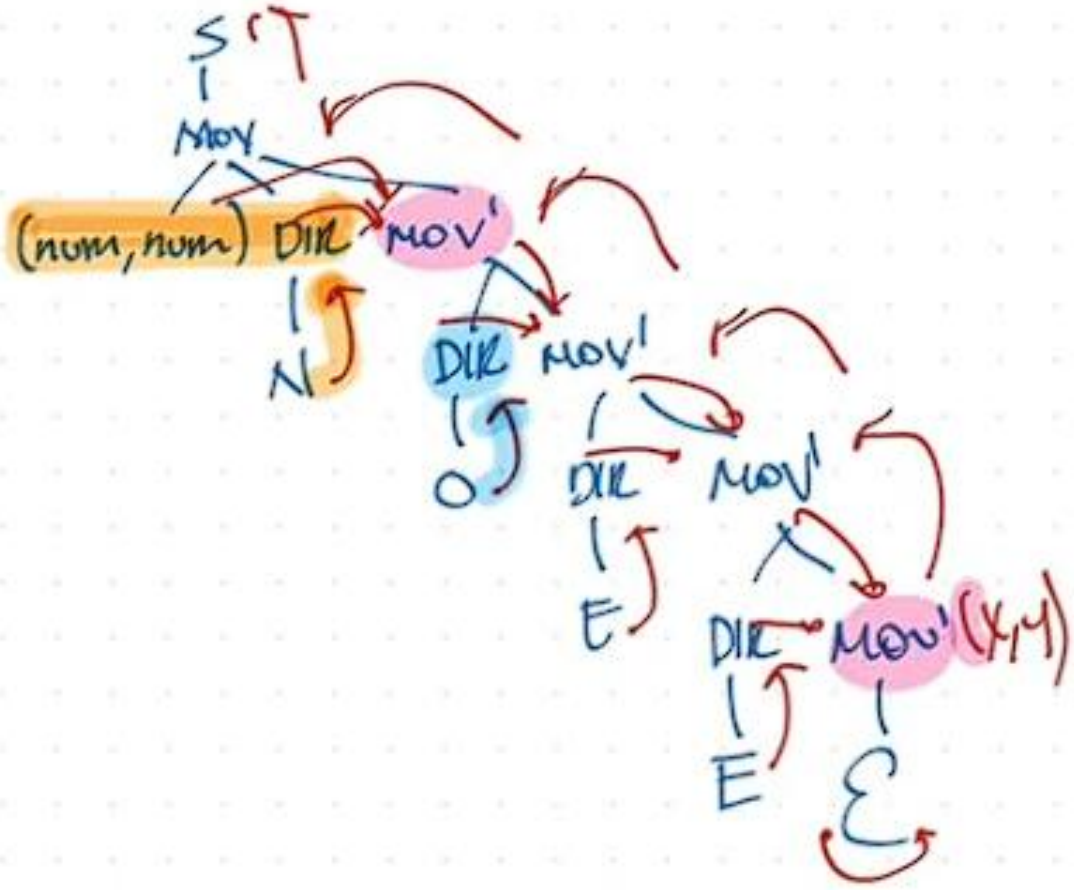
| ε { MOV'.xs = MOV'.x ; MOV'.ys = MOV'.y }

DIR → N { DIR.x = 0 DIR.y = 1 }

| S { DIR.x = 0 DIR.y = -1 }

| E { DIR.x = 1 DIR.y = ϕ }

| O { DIR.x = -1 DIR.y = ϕ }



ESQUEMA 2

(1,2) NOEE

$S \rightarrow D$

$D \rightarrow (num, num)\ MOV\ \{\ \_\_\_\ \}$ ②

$MOV \rightarrow DIR\ MOV'$ $\{\ \_\_\_\ \}$

$MOV' \rightarrow DIR\ MOV'$ $\{\ \_\_\_\ \}$

$| \varepsilon$ $\{\ \_\_\_\ \}$

$DIR \rightarrow N$ ①

$| S$

$| E$

$| 0$