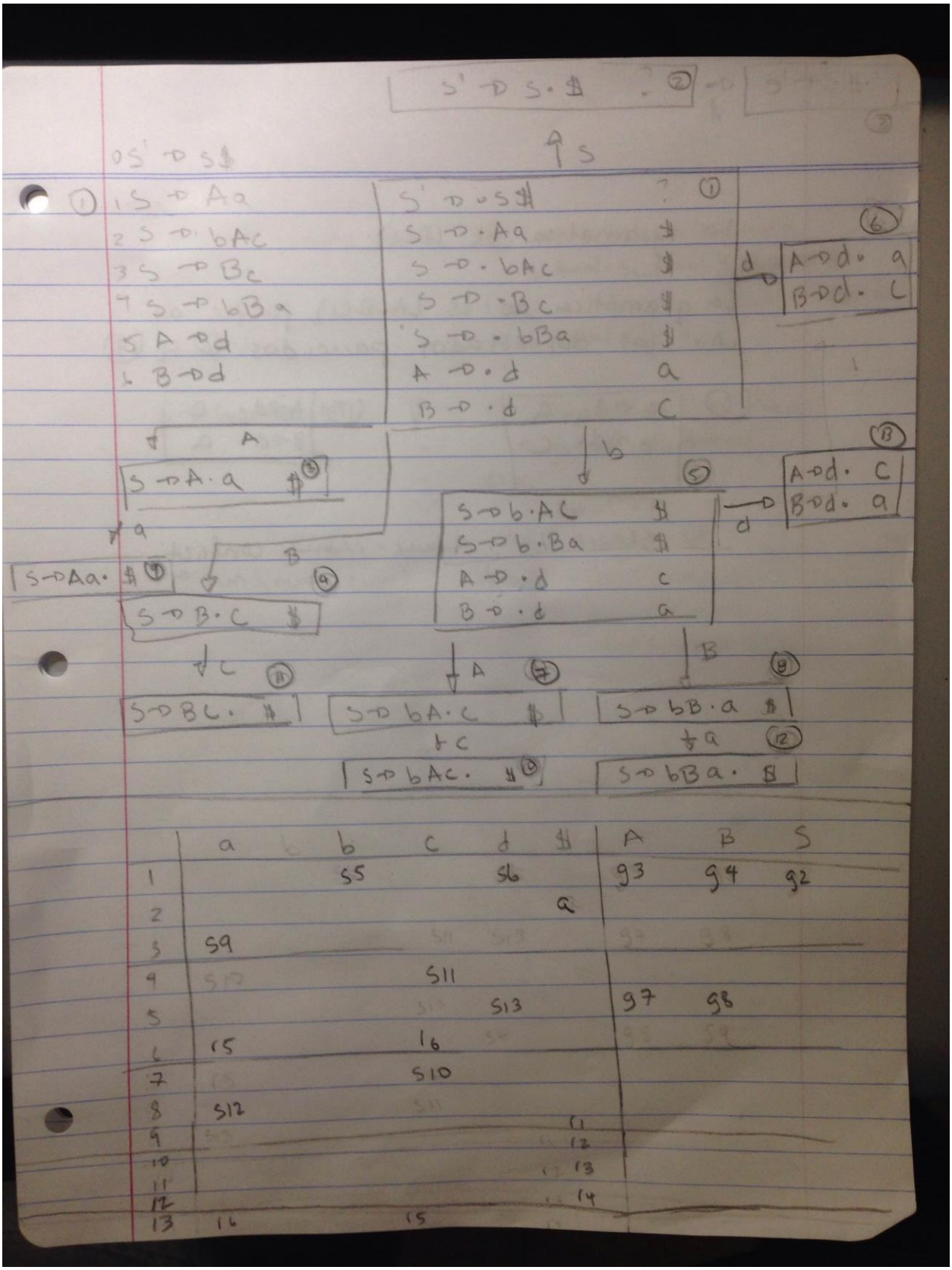
COMPILADORES I

Tarea No. 4

Horacio Galdamez
Cuenta No. 11241022

Ejercicio 1


S' → S · $
0 S' → S$
1 S → Aa
2 S → bAc
3 S → Bc
4 S → bBa
5 A → d
6 B → d
S' → · S $
S → · Aa $
S → · bAc $
S → · Bc $
S → · bBa $
A → · d a
B → · d c
S → A · a $
S → Aa · $
S → B · c $
S → bA · c $
S → bAc · $
S → bB · a $
S → bBa · $
A → d · a
B → d · c
A → d · c
B → d · a
a b c d $ A B S
1 s5 s6 g3 g4 g2
2 a
3 s9
4 s11
5 s13 g7 g8
6 r5 16
7 s10
8 s12
9 r1
10 r2
11 r3
12 r4
13 r6 r5

∴ La gramática es LR(1)

∴ La gramática no es LALR(1) porque al unir los dos estados parecidos: (6) y (13)

(6)

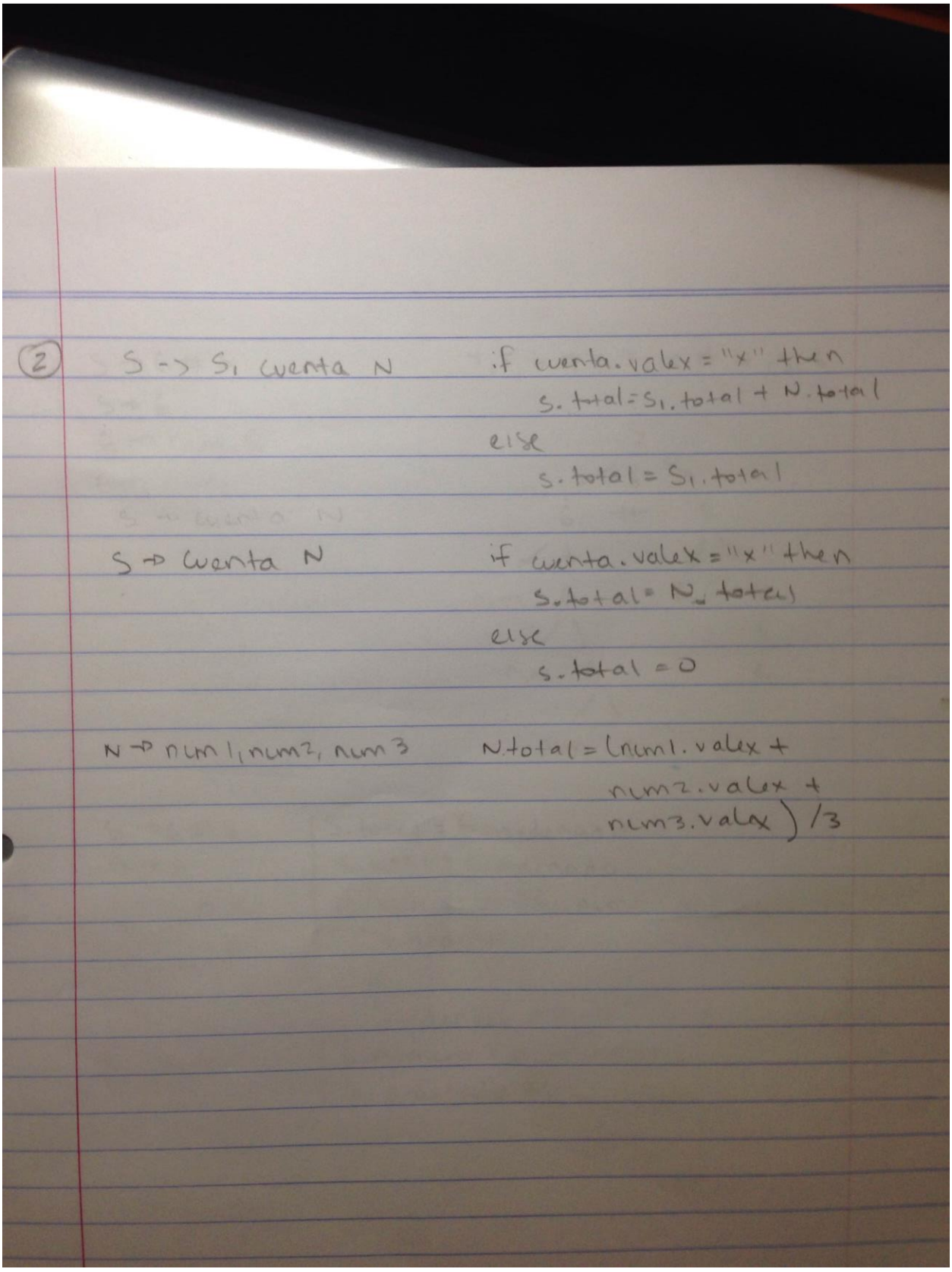
A → d · a
b → d · c

y (13)

A → d · c
B → d · a

Se genera un reduce-reduce conflict.

Ejercicio 2



(2) S -> S1 cuenta N if cuenta.valex = "x" then
S.total = S1.total + N.total
else
S.total = S1.total

S -> cuenta N if cuenta.valex = "x" then
S.total = N.total
else
S.total = 0

N -> num1, num2, num3 N.total = (num1.valex +
num2.valex +
num3.valx) /3

Ejercicio 3

(3)

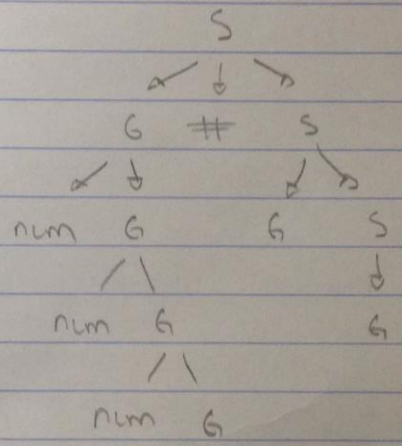
S → G # S

S → G

G → num, G

G → num

4, 5, 6 # 5, 7, 10 # 10, 5, 6

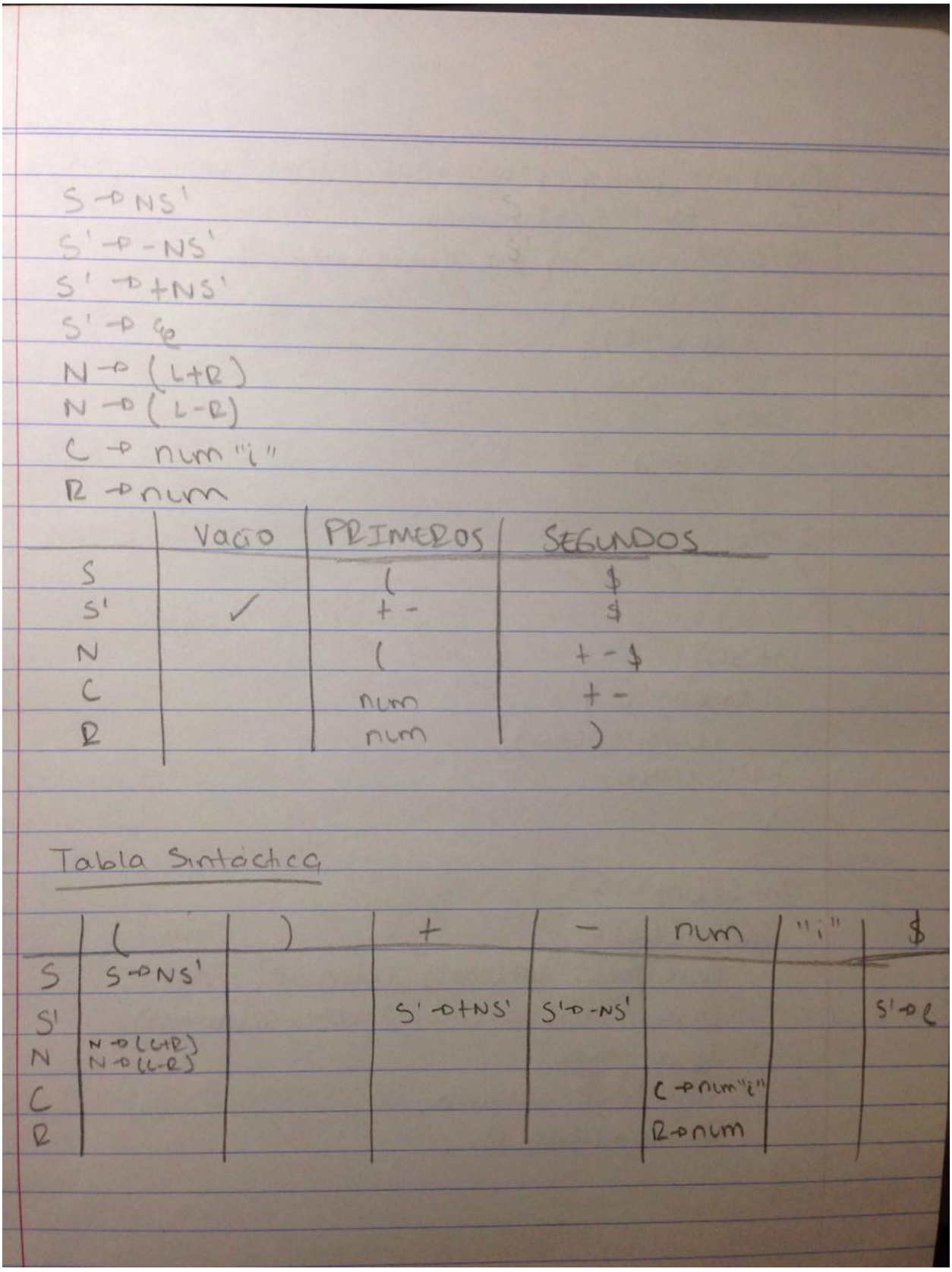


S → G # S1	S.total = G.ordenado + S1.total
S → G	S.total = G.ordenado

G → num, G1	if (num.valex ≥ G1.numero and G1.ordenado = 1)
	G.ordenado = 1
	else
	G.ordenado = 0

G → num	G.numero = num.valex ; G.ordenado = 1

Ejercicio 4



S → NS'
S' → -NS'
S' → +NS'
S' → ε
N → (L+R)
N → (L-R)
C → num "¡"
R → num

	Vacío	PRIMEROS	SEGUNDOS
S		(	$
S'	✓	+ -	$
N		(	+ - $
C		num	+ -
R		num	)

Tabla Sintáctica

	(	)	+	-	num	"¡"	$
S	S → NS'						
S'			S' → +NS'	S' → -NS'			S' → ε
N	N → (L+R) N → (L-R)						
C					C → num "¡"		
R					R → num		

```
Final int parAq, parder, mas, menos, num=1,2,3,4,5;
Token tok = getToken();
Void advance () { tok. getToken(); }
Void eat (int t) {
   if (tok==t) {
      advance();
   else
      error();
   }
}


int S(){
 if (tok == "(")
    return SP(N());
 } else error();
 }

Int SP(int h) {
 Switch (tok) {
    Case mas: eat(mas); return SP(h+N());
    Case menos: eat(menos); return SP(h-N());
    Case parder;
    Case dolar: return h;
    Default: error();
}
```

```
int N(){
  if (tok == parizq)
    int x=C();
  Switch(tok){
    case+: return x+R();
    case-: return x-R();
    Default: error();
  }
}

Int C(){
  eat(num);
  eat(i);
  return num.val;
}

int R(){
  eat(num);
  return num.val;
}
```