Ejercicio 1

Escriba un archivo calc.y que implemente la gramática e imprima el valor de la expresión:

exp → exp opsuma term | term
opsuma → + | -
term → term opmult factor | factor
opmult → *
factor → (exp) | numero

Sugerencia: Referencia código en la pág. 228 del libro

Ver carpeta Pregunta1

Grammar

```
    0 $accept: command $end

    1 command: exp

    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'
```

Terminals, with rules where they appear

```
$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
```

Grammar

```
    0 $accept: command $end

    1 command: exp
```

```
    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7
Grammar

    0 $accept: command $end

    1 command: exp

    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
```

'-' (45) 3
error (256)
NUMBER (258) 7

Nonterminals, with rules where they appear

$accept (9)
  on left: 0
command (10)
  on left: 1, on right: 0
exp (11)
Grammar

  0 $accept: command $end

  1 command: exp

  2 exp: exp '+' term
  3 | exp '-' term
  4 | term

  5 term: term '*' factor
  6 | factor

  7 factor: NUMBER
  8 | '(' exp ')'

Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7

Nonterminals, with rules where they appear

$accept (9)
  on left: 0
command (10)

```
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4, on right: 1 2 3 8
term (12)
    on left: 5 6, on right: 2 3 4 5
Grammar

    0 $accept: command $end

    1 command: exp

    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7


Nonterminals, with rules where they appear

$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4, on right: 1 2 3 8
term (12)
    on left: 5 6, on right: 2 3 4 5
factor (13)
Grammar
```

```
    0 $accept: command $end

    1 command: exp

    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7


Nonterminals, with rules where they appear

$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4, on right: 1 2 3 8
term (12)
    on left: 5 6, on right: 2 3 4 5
factor (13)
    on left: 7 8, on right: 5 6
Grammar

    0 $accept: command $end

    1 command: exp
```

```
    2 exp: exp '+' term
    3    | exp '-' term
    4    | term

    5 term: term '*' factor
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7


Nonterminals, with rules where they appear

$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4, on right: 1 2 3 8
term (12)
    on left: 5 6, on right: 2 3 4 5
factor (13)
    on left: 7 8, on right: 5 6
Grammar

    0 $accept: command $end

    1 command: exp
    term    go to state 14
    factor  go to state 6


state 11
```

```
    5 term: term '*' . factor

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    factor  go to state 15


state 12

    8 factor: '(' exp ')' .

    $default  reduce using rule 8 (factor)


state 13

    2 exp: exp '+' term .
    5 term: term . '*' factor

    '*'  shift, and go to state 11

    $default  reduce using rule 2 (exp)


state 14

    3 exp: exp '-' term .
    5 term: term . '*' factor

    '*'  shift, and go to state 11

    $default  reduce using rule 3 (exp)


state 15

    5 term: term '*' factor .
    6     | factor

    7 factor: NUMBER
    8       | '(' exp ')'


Terminals, with rules where they appear
```

$end (0) 0
'(' (40) 8
')' (41) 8
'*' (42) 5
'+' (43) 2
'-' (45) 3
error (256)
NUMBER (258) 7

Nonterminals, with rules where they appear

$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4, on right: 1 2 3 8
term (12)
    on left: 5 6, on right: 2 3 4 5
factor (13)
    on left: 7 8, on right: 5 6

state 0

    0 $accept: . command $end

    NUMBER shift, and go to state 1
    '('   shift, and go to state 2

    command go to state 3
    exp   go to state 4
    term   go to state 5
    factor   go to state 6

state 1

    7 factor: NUMBER .

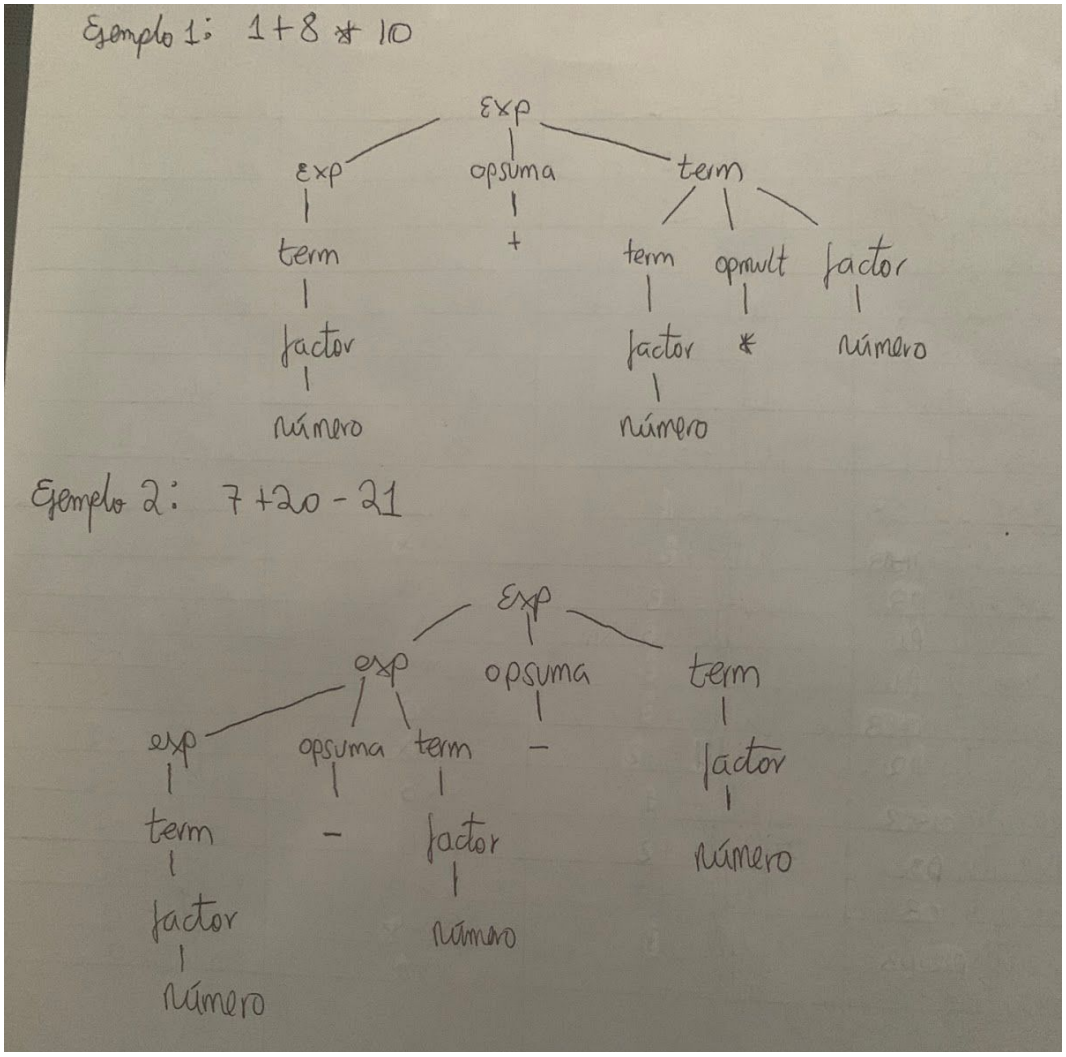
    term   go to state 14
    factor go to state 6

state 11

```
    5 term: term '*' . factor

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    factor  go to state 15


state 12

    8 factor: '(' exp ')' .

    $default  reduce using rule 8 (factor)


state 13

    2 exp: exp '+' term .
    5 term: term . '*' factor

    '*'  shift, and go to state 11

    $default  reduce using rule 2 (exp)


state 14

    3 exp: exp '-' term .
    5 term: term . '*' factor

    '*'  shift, and go to state 11

    $default  reduce using rule 3 (exp)


state 15

    5 term: term '*' factor .

    $default  reduce using rule 5 (term)
```

No es ambigua

Tiene precedencia por la multiplicación, osea va a realizar operaciones como una calculadora lo haría (semánticamente bien).



Ejercicio 2

Escriba un archivo calc2.y que implemente la gramática e imprima el valor de la expresión:

exp → numero | exp + exp | exp - exp | exp * exp | (exp)

- Genere el archivo y.output
- ¿Es ambigua?
- ¿Qué precedencia tiene? ¿Cómo lo implementa yacc?

Ver carpeta Pregunta2

Si es ambigua, conflicto desplazamiento/reducción

yacc lo implementa de la siguiente manera:

State 11 conflicts: 3 shift/reduce
State 12 conflicts: 3 shift/reduce
State 13 conflicts: 3 shift/reduce

Grammar

0 $accept: command $end

1 command: exp

2 exp: NUMBER
3 | exp '+' exp
4 | exp '-' exp
5 | exp '*' exp
6 | '(' exp ')'

Terminals, with rules where they appear

$end (0) 0
'(' (40) 6
')' (41) 6

'*' (42) 5
'+' (43) 3
'-' (45) 4
error (256)
NUMBER (258) 2

Nonterminals, with rules where they appear

$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4 5 6, on right: 1 3 4 5 6

state 0

    0 $accept: . command $end

    NUMBER shift, and go to state 1
    '('   shift, and go to state 2

    command go to state 3
    exp   go to state 4

state 1

    2 exp: NUMBER .

    $default reduce using rule 2 (exp)

state 2

    6 exp: '(' . exp ')'

    NUMBER shift, and go to state 1
    '('   shift, and go to state 2

    exp go to state 5

state 3

```
    0 $accept: command . $end

    $end  shift, and go to state 6


state 4

    1 command: exp .
    3 exp: exp . '+' exp
    4    | exp . '-' exp
    5    | exp . '*' exp

    '+'  shift, and go to state 7
    '-'  shift, and go to state 8
    '*'  shift, and go to state 9

    $default  reduce using rule 1 (command)


state 5

    3 exp: exp . '+' exp
    4    | exp . '-' exp
    5    | exp . '*' exp
    6    | '(' exp . ')'

    '+'  shift, and go to state 7
    '-'  shift, and go to state 8
    '*'  shift, and go to state 9
    ')'  shift, and go to state 10


state 6

    0 $accept: command $end .

    $default  accept


state 7

    3 exp: exp '+' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2
```

```
    exp  go to state 11


state 8

    4 exp: exp '-' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    exp  go to state 12


state 9

    5 exp: exp '*' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    exp  go to state 13


state 10

    6 exp: '(' exp ')' .

    $default  reduce using rule 6 (exp)


state 11

    3 exp: exp . '+' exp
    3    | exp '+' exp .
    4    | exp . '-' exp
    5    | exp . '*' exp

    '+'  shift, and go to state 7
    '-'  shift, and go to state 8
    '*'  shift, and go to state 9

    '+'       [reduce using rule 3 (exp)]
    '-'       [reduce using rule 3 (exp)]
    '*'       [reduce using rule 3 (exp)]
    $default  reduce using rule 3 (exp)
```

```
state 12

    3 exp: exp . '+' exp
    4    | exp . '-' exp
    4    | exp '-' exp .
    5    | exp . '*' exp

    '+'  shift, and go to state 7
    '-'  shift, and go to state 8
    '*'  shift, and go to state 9

    '+'       [reduce using rule 4 (exp)]
    '-'       [reduce using rule 4 (exp)]
    '*'       [reduce using rule 4 (exp)]
    $default  reduce using rule 4 (exp)


state 13

    3 exp: exp . '+' exp
    4    | exp . '-' exp
    5    | exp . '*' exp
    5    | exp '*' exp .

    '+'  shift, and go to state 7
    '-'  shift, and go to state 8
    '*'  shift, and go to state 9

    '+'       [reduce using rule 5 (exp)]
    '-'       [reduce using rule 5 (exp)]
    '*'       [reduce using rule 5 (exp)]
    $default  reduce using rule 5 (exp)
```

Se eliminó la ambigüedad (ver archivo calc2_2.y) y yacc no díо esto:

Grammar

```
    0 $accept: command $end

    1 command: exp

    2 exp: NUMBER
    3    | NUMBER '+' exp
    4    | NUMBER '-' exp
    5    | NUMBER '*' exp
    6    | '(' exp ')'
```

Terminals, with rules where they appear

```
$end (0) 0
'(' (40) 6
')' (41) 6
'*' (42) 5
'+' (43) 3
'-' (45) 4
error (256)
NUMBER (258) 2 3 4 5
```

Nonterminals, with rules where they appear

```
$accept (9)
    on left: 0
command (10)
    on left: 1, on right: 0
exp (11)
    on left: 2 3 4 5 6, on right: 1 3 4 5 6
```

```
state 0

    0 $accept: . command $end

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    command  go to state 3
    exp      go to state 4
```

```
state 1

    2 exp: NUMBER .
    3    | NUMBER . '+' exp
    4    | NUMBER . '-' exp
    5    | NUMBER . '*' exp

    '+'  shift, and go to state 5
    '-'  shift, and go to state 6
    '*'  shift, and go to state 7

    $default  reduce using rule 2 (exp)


state 2

    6 exp: '(' . exp ')'

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    exp  go to state 8


state 3

    0 $accept: command . $end

    $end  shift, and go to state 9


state 4

    1 command: exp .

    $default  reduce using rule 1 (command)


state 5

    3 exp: NUMBER '+' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2
```

```
    exp  go to state 10


state 6

    4 exp: NUMBER '-' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    exp  go to state 11


state 7

    5 exp: NUMBER '*' . exp

    NUMBER  shift, and go to state 1
    '('     shift, and go to state 2

    exp  go to state 12


state 8

    6 exp: '(' exp . ')'

    ')'  shift, and go to state 13


state 9

    0 $accept: command $end .

    $default  accept


state 10

    3 exp: NUMBER '+' exp .

    $default  reduce using rule 3 (exp)


state 11
```

4 exp: NUMBER '-' exp .

$default reduce using rule 4 (exp)

state 12

5 exp: NUMBER '*' exp .

$default reduce using rule 5 (exp)

state 13

6 exp: '(' exp ')' .

$default reduce using rule 6 (exp)

Su precedencia no es semánticamente correcta ya que en una operación que contenga + - * o () no se va a resolver como debería, se resolverá de acuerdo al orden en que se lleguen a las expresiones.

Ejercicio 3

Escriba un archivo cfg3.y que implemente la gramática :

S → A | B
A → a
B → a

- Genere el archivo y.output
- ¿Es ambigua?
- ¿Cómo lo implementa yacc?

Ver carpeta Pregunta3

Es ambigua, conflicto reducción/reducción

```
conflicts: 1 reduce/reduce
cfg3.y:22.5-25: warning: rule never reduced because of conflicts: B: JUANCITO
```

yacc lo implementa de la siguiente manera:

Rules never reduced

5 B: JUANCITO

State 1 conflicts: 1 reduce/reduce

Grammar

0 $accept: command $end

1 command: S

2 S: A
3 | B

4 A: JUANCITO

5 B: JUANCITO

Terminals, with rules where they appear

$end (0) 0
error (256)
JUANCITO (258) 4 5

Nonterminals, with rules where they appear

$accept (4)
    on left: 0
command (5)
    on left: 1, on right: 0
S (6)
    on left: 2 3, on right: 1
A (7)
    on left: 4, on right: 2
B (8)
    on left: 5, on right: 3

state 0

    0 $accept: . command $end
Rules never reduced

    5 B: JUANCITO

State 1 conflicts: 1 reduce/reduce

Grammar

    0 $accept: command $end

    1 command: S

    2 S: A
    3 | B

    4 A: JUANCITO

    5 B: JUANCITO

Terminals, with rules where they appear

$end (0) 0
error (256)
JUANCITO (258) 4 5

Nonterminals, with rules where they appear

$accept (4)
    on left: 0
command (5)
    on left: 1, on right: 0
S (6)
    on left: 2 3, on right: 1
A (7)
    on left: 4, on right: 2
B (8)
    on left: 5, on right: 3

state 0

    0 $accept: . command $end

    JUANCITO shift, and go to state 1
    command go to state 2
    S   go to state 3
    A   go to state 4
    B   go to state 5

state 1

    4 A: JUANCITO .
    5 B: JUANCITO .

    $end   reduce using rule 4 (A)
    $end   [reduce using rule 5 (B)]
    $default reduce using rule 4 (A)

state 2

0 $accept: command . $end

$end shift, and go to state 6

state 3

1 command: S .

$default reduce using rule 1 (command)

state 4

2 S: A .

$default reduce using rule 2 (S)

state 5

3 S: B .

$default reduce using rule 3 (S)

state 6

0 $accept: command $end .

$default accept

Se eliminó la ambigüedad (ver archivo cfg3_2.y) y yacc no dío esto:

Grammar

0 $accept: command $end

1 command: S

2 S: M

3 M: JUANCITO

Terminals, with rules where they appear

$end (0) 0
error (256)
JUANCITO (258) 3

Nonterminals, with rules where they appear

$accept (4)
on left: 0
command (5)
on left: 1, on right: 0
S (6)
on left: 2, on right: 1
M (7)
on left: 3, on right: 2

state 0

0 $accept: . command $end

JUANCITO shift, and go to state 1

command go to state 2
S go to state 3
M go to state 4

state 1

3 M: JUANCITO .

```
    $default  reduce using rule 3 (M)


state 2
Grammar

    0 $accept: command $end

    1 command: S

    2 S: M

    3 M: JUANCITO


Terminals, with rules where they appear

$end (0) 0
error (256)
JUANCITO (258) 3


Nonterminals, with rules where they appear

$accept (4)
    on left: 0
command (5)
    on left: 1, on right: 0
S (6)
    on left: 2, on right: 1
M (7)
    on left: 3, on right: 2


state 0

    0 $accept: . command $end

    JUANCITO  shift, and go to state 1

    command  go to state 2
    S        go to state 3
    M        go to state 4
```

state 1

    3 M: JUANCITO .

    $default reduce using rule 3 (M)

state 2

    0 $accept: command . $end

    $end shift, and go to state 5
$accept (4)
    on left: 0
command (5)
    on left: 1, on right: 0
S (6)
    on left: 2, on right: 1
M (7)
    on left: 3, on right: 2

state 0

    0 $accept: . command $end

    JUANCITO shift, and go to state 1

    command go to state 2
    S   go to state 3
    M   go to state 4

state 1

    3 M: JUANCITO .

    $default reduce using rule 3 (M)

state 2

    0 $accept: command . $end

    $end shift, and go to state 5

state 3

    1 command: S .

    $default reduce using rule 1 (command)

state 4

    2 S: M .

    $default reduce using rule 2 (S)

state 5

    0 $accept: command $end .

    $default accept