

Μεταγλωττιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: ΤΡΑΝΤΗ ΕΛΕΝΗ

ΑΜ: Π2016106

Κανόνες γραμματικής:

Grammar	
Stmt_list→	Stmt Stmt_list   .
Stmt→	id equal Expr   print Expr .
Expr→	Term Term_tail .
Term_tail→	xor Term Term_tail   .
Term→	Factor Factor_tail .
Factor_tail→	or Factor Factor_tail   .
Factor→	Atom Atom_tail .
Atom_tail→	and Atom Atom_tail   .
Factor→	(Expr)   id   number .

Η γραμματική της παραπάνω εικόνας είναι αυτή που χρησιμοποιήθηκε στην εργασία. Είναι αναδρομική κάτι που σημαίνει ότι υπάρχει σειρά προτεραιότητας: and, or, xor.

## Αποτελέσματα ελέγχου LL(1) συμβατότητας:

```
Grammar
Stmt_list → Stmt Stmt_list
           | .
Stmt → id equal Expr
      | print Expr .
Expr → Term Term_tail .
Term_tail → xor Term Term_tail
          | .
Term → Factor Factor_tail .
Factor_tail → or Factor Factor_tail
            | .
Factor → Atom Atom_tail .
Atom_tail → and Atom Atom_tail
          | .
Factor → (Expr)
        | id
        | number .
```

Some sentences generated by this grammar: {ε, print id, id equal id, print (Expr), print number, id equal number, id equal (Expr), id equal id or id, print (Expr) or id, print id or (Expr), id equal id or number, id equal id or (Expr), id equal number or id, id equal (Expr) or id, print (Expr) or (Expr), print (Expr) or number, id equal number or (Expr), id equal number or number, id equal (Expr) or number, id equal (Expr) or (Expr)}

- You have unrealizable nonterminals in your grammar. They are: Atom
- The nullable nonterminals are: Stmt\_list Term\_tail Factor\_tail Atom\_tail.
- The endable nonterminals are: Atom\_tail Atom Factor\_tail Factor Term\_tail Term Expr Stmt\_list Stmt.
- No cycles.

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	∅	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id number	id print	no	yes
Term_tail	xor	id print	yes	yes
Term	(Expr) id number	xor id print	no	yes
Factor_tail	or	xor id print	yes	yes
Factor	(Expr) id number	or xor id print	no	yes
Atom	∅	and or xor id print	no	yes
Atom_tail	and	or xor id print	yes	yes

The grammar is LL(1).

- attempt to [transform](#) the grammar (to LL(1))
- generate [LL\(1\)](#) parsing table
- generate [LR\(0\)/SLR\(1\)](#) automaton
- generate [LALR\(1\)](#) automaton

Η γραμματική είναι LL(1) όπως μας δείχνει και το αποτέλεσμα του παραπάνω εργαλείου.

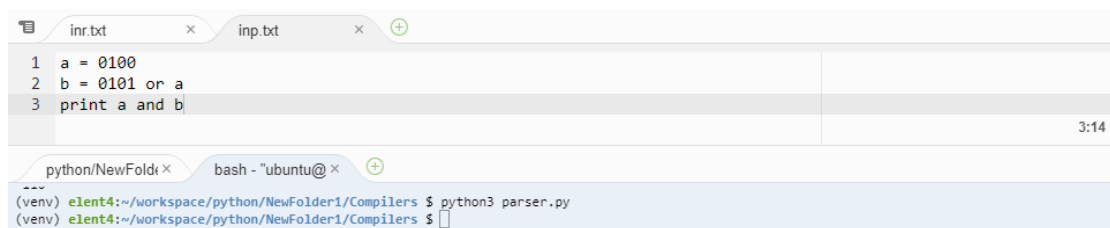
## Πίνακες με τα First και Follow sets:

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	$\emptyset$	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id number	id print	no	yes
Term_tail	xor	id print	yes	yes
Term	(Expr) id number	xor id print	no	yes
Factor_tail	or	xor id print	yes	yes
Factor	(Expr) id number	or xor id print	no	yes
Atom	$\emptyset$	and or xor id print	no	yes
Atom_tail	and	or xor id print	yes	yes

The grammar is LL(1).

## Αποτελέσματα εξόδου:

Έγκυρη είσοδος και έξοδος του parser.py.



```
1 a = 0100
2 b = 0101 or a
3 print a and b
```

```
(venv) elent4:~/workspace/python/NewFolder1/Compilers $ python3 parser.py
(venv) elent4:~/workspace/python/NewFolder1/Compilers $
```

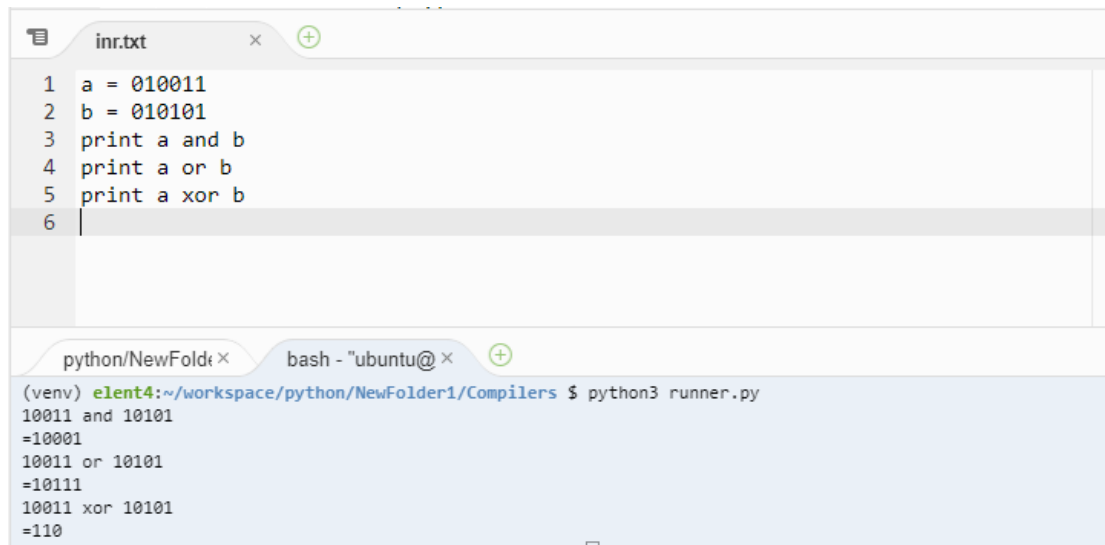
Άκυρη είσοδος και έξοδος του parser.py.



```
1 print xor 0
```

```
(venv) elent4:~/workspace/python/NewFolder1/Compilers $ python3 parser.py
Traceback (most recent call last):
  File "parser.py", line 152, in <module>
    parser.parse(fp)
  File "parser.py", line 54, in parse
    self.stmt_list()
  File "parser.py", line 58, in stmt_list
    self.stmt()
  File "parser.py", line 74, in stmt
    self.expr()
  File "parser.py", line 85, in expr
    raise ParseError("Expected ( or id or num or )")
__main__.ParseError: Expected ( or id or num or )
(venv) elent4:~/workspace/python/NewFolder1/Compilers $
```

Έγκυρη είσοδος και έξοδος του runner.py.



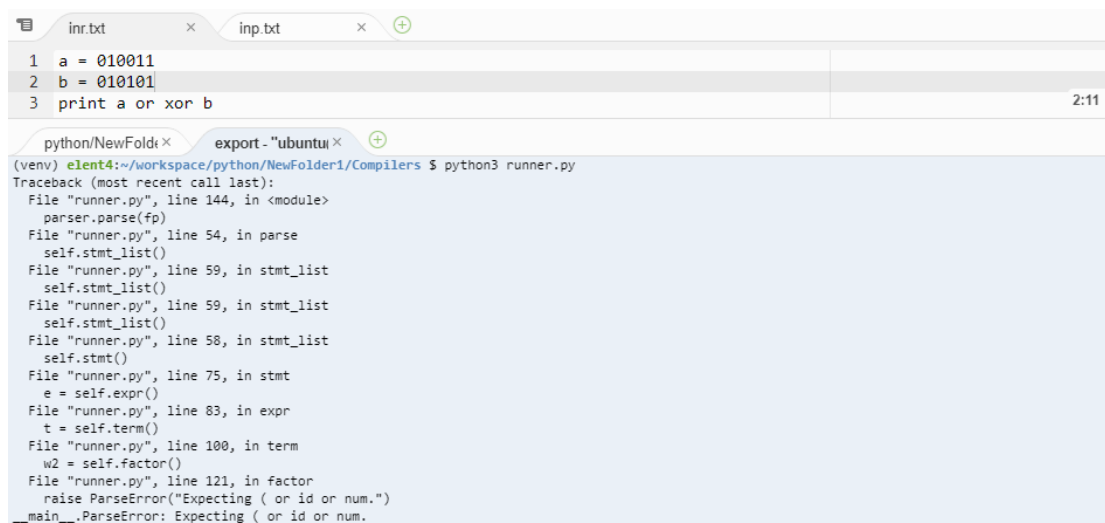
The screenshot shows a code editor with a file named `inr.txt` containing the following Python code:

```
1 a = 010011
2 b = 010101
3 print a and b
4 print a or b
5 print a xor b
6
```

Below the code editor, a terminal window shows the execution of the script using `python3 runner.py`. The output is as follows:

```
(venv) elent4:~/workspace/python/NewFolder1/Compilers $ python3 runner.py
10011 and 10101
=10001
10011 or 10101
=10111
10011 xor 10101
=110
```

Άκυρη είσοδος και έξοδος του runner.py.



The screenshot shows a code editor with a file named `inp.txt` containing the following Python code:

```
1 a = 010011
2 b = 010101
3 print a or xor b
```

Below the code editor, a terminal window shows the execution of the script using `python3 runner.py`. The output is a traceback error:

```
(venv) elent4:~/workspace/python/NewFolder1/Compilers $ python3 runner.py
Traceback (most recent call last):
  File "runner.py", line 144, in <module>
    parser.parse(fp)
  File "runner.py", line 54, in parse
    self.stmt_list()
  File "runner.py", line 59, in stmt_list
    self.stmt_list()
  File "runner.py", line 59, in stmt_list
    self.stmt_list()
  File "runner.py", line 58, in stmt_list
    self.stmt()
  File "runner.py", line 75, in stmt
    e = self.expr()
  File "runner.py", line 83, in expr
    t = self.term()
  File "runner.py", line 100, in term
    w2 = self.factor()
  File "runner.py", line 121, in factor
    raise ParseError("Expecting ( or id or num.")
__main__.ParseError: Expecting ( or id or num.
```

## Πηγές:

Cloud9: <https://ide.c9.io>