

**Μεταγλωτιστές 2019**  
**Προγραμματιστική Εργασία #2**

**Ονοματεπώνυμο: Διονύσιος Κοντοζίδης**  
**ΑΜ: Π2016178**

## Κανόνες Γραμματικής

| Grammar       |  |
|---------------|--|
| Stmt_list →   | Stmt Stmt_list<br>  .                  |
| Stmt →        | id equal Exp<br>  print Exp.           |
| Exp →         | Term Term_tail.                        |
| Term_tail →   | xor Term Term_tail<br>  .              |
| Term →        | Factor Factor_tail.                    |
| Factor_tail → | or Factor Factor_tail<br>  .           |
| Factor →      | Atom Atom_tail.                        |
| Atom_tail →   | and Atom Atom_tail<br>  .              |
| Atom →        | lparen Exp rparen<br>  id<br>  number. |

### *1.1 Σύνολο κανόνων γραμματικής*

Στην παραπάνω εικόνα είναι το σετ κανόνων γραμματικής που χρησιμοποιήθηκε για την υλοποίηση της εργασίας. Στην διαδικασία αυτή χρησιμοποιήθηκε η μέθοδος της αναδρομικής κατάβασης. Η γραμματική κατασκευάστηκε με τέτοιο τρόπο ώστε να δέχεται συμβολοσειρές με bits και λογικές πράξεις μεταξύ αυτών με προτεραιότητα:

- ❖ and
- ❖ or
- ❖ xor

Εξαιτίας αυτού στην γραμματική τοποθετήθηκε το and στο **term\_tail**, το or στο **factor\_tail** και το xor στο **atom\_tail** γιατί όσο πιο κάτω είναι ένας τελεστής στους κανόνες γραμματικής, τόσο πιο μεγάλη προτεραιότητα έχει.

## Αποτελέσματα Ελέγχου για LL(1) συμβατότητα

- All nonterminals are reachable and realizable.
- 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 Exp Stmt\_list Stmt.
- No cycles.

| nonterminal | first set        | follow set                 | nullable | endable |
|-------------|------------------|----------------------------|----------|---------|
| Stmt_list   | id print         | $\emptyset$                | yes      | yes     |
| Stmt        | id print         | id print                   | no       | yes     |
| Term_tail   | xor              | rparen id print            | yes      | yes     |
| Term        | lparen id number | rparen xor id print        | no       | yes     |
| Factor_tail | or               | rparen xor id print        | yes      | yes     |
| Factor      | lparen id number | rparen or xor id print     | no       | yes     |
| Atom_tail   | and              | rparen or xor id print     | yes      | yes     |
| Atom        | lparen id number | rparen and or xor id print | no       | yes     |
| Exp         | lparen id number | rparen id print            | no       | yes     |

The grammar is LL(1).

### 2.1 Αποτελέσματα για LL(1) συμβατότητα

Η γραμματική είναι LL(1) όπως φαίνεται και παραπάνω. Αυτό σημαίνει ότι το διάβασμα των κανόνων γίνεται από τα αριστερά στα δεξιά και ότι διαθέτετε μονοπάτι προς όλα τα μη τερματικά σύμβολα. Ακόμα, δεν υπάρχουν κυκλικές μεταβάσεις και συνεπώς συμφωνεί με την μέθοδο της αναδρομικής κατάβασης που προαναφέρθηκε.

**Πίνακας με FIRST και FOLLOW σελ της γραμματικής**

| <b>Μη<br/>τερματικά</b> | <b>FIRST</b> | <b>FOLLOW</b>         |
|-------------------------|--------------|-----------------------|
| Stmt_list               | id print     | ε                     |
| Stmt                    | id print     | id print              |
| Term_tail               | xor          | ) id print ε          |
| Term                    | ( id number  | ) xor id print        |
| Factor_tail             | or           | ) xor id print ε      |
| Factor                  | ( id number  | ) or xor id print     |
| Atom_tail               | and          | ) or xor id print ε   |
| Atom                    | ( id number  | ) and or xor id print |
| Exp                     | ( id number  | ) id print            |

Τα FIRST και FOLLOW σελ χρησιμοποιούνται ώστε ο συντακτικός αναλυτής να είναι σε θέση να εφαρμόσει τον σωστό κανόνα στην σωστή θέση.

## Αποτελέσματα Εξόδου

```
1 a = 1001
2 b = 101 xor 111
3 c = 1001100 and 0001001
4 print (10101 and 10111 or 11001 xor 00001)
5 print a
6 print 1 or 0 xor 1
7 print c
8 print b
```

Line 8, Column 8 Tab Size: 4

```
root@DESKTOP-OIKTPPO: /mnt/c/Users/User/Desktop/compilers
root@DESKTOP-OIKTPPO: /mnt/c/Users/User/Desktop/compil
11100
1001
0
1000
root@DESKTOP-OIKTPPO: /mnt/c/Users/User/Desktop/compil
```

### 4.1 Είσοδος και έξοδος μετά την εκτέλεση του **runner.py** με σωστή είσοδο

```
test.txt x
1 a = 1010101 ^ 1912
2 c = 010 | 011
```

Line 2, Column 14 Tab Size: 4

```
root@DESKTOP-OIKTPPO: /mnt/c/Users/User/Desktop/compilers# python3 runner.py
Traceback (most recent call last):
  File "runner.py", line 129, in <module>
    parser.parse(fp)
  File "runner.py", line 46, in parse
    self.stmt_list()
  File "runner.py", line 50, in stmt_list
    self.stmt()
  File "runner.py", line 61, in stmt
    e = self.expr()
  File "runner.py", line 71, in expr
    t = self.term()
  File "runner.py", line 84, in term
    t=self.factor()
  File "runner.py", line 98, in factor
    t=self.atom()
  File "runner.py", line 123, in atom
    self.match('BIT_TOKEN')
  File "runner.py", line 40, in match
    self.la,self.text=self.next_token()
  File "runner.py", line 36, in next_token
    return self.scanner.read()
  File "/usr/local/lib/python3.6/dist-packages/plex/scanners.py", line 94, in
read
    self.text, action = self.scan_a_token()
  File "/usr/local/lib/python3.6/dist-packages/plex/scanners.py", line 138, i
n scan_a_token
    raise errors.UnrecognizedInput(self, self.state_name)
plex.errors.UnrecognizedInput: '', line 1, char 12: Token not recognised in s
tate ''
```

### 4.2 Είσοδος και έξοδος μετά την εκτέλεση του προγράμματος **runner.py** με λάθος είσοδο

## Πηγές

Χρησιμοποιήθηκε το Context Free Grammar Tool για τον έλεγχο της συμβατότητας της γραμματικής και τον υπολογισμό των FIRST και FOLLOW set.

Link: <http://smlweb.cpsc.ucalgary.ca/start.html>