Μεταγλωτιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Διονύσης Κάλφας
ΑΜ: Π2014112

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
            | .
Atom →        (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 print id, id equal id id equal id, id equal id print (Expr), id equal (Expr) print id, id equal id print number, id equal id id equal (Expr), id equal (Expr) id equal id, id equal id id equal number, id equal (Expr) print (Expr), id equal (Expr) print number, id equal (Expr) id equal number, id equal (Expr) id equal (Expr), id equal number id equal (Expr)}

- You have unrealizable nonterminals in your grammar. They are: XOR OR AND
- 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.

Η σειρά προτεραιότητας των τελεστών πράξεων σε επίπεδο bits (από την υψηλότερη προς τη χαμηλότερη) είναι: and, or, xor.

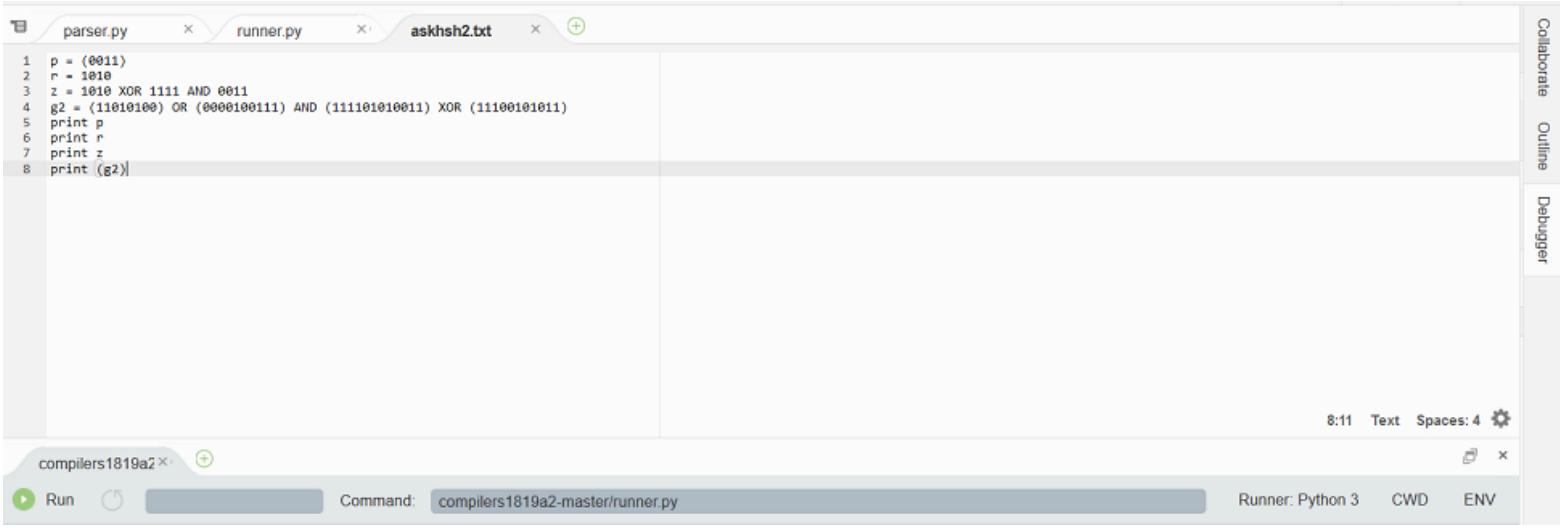
Για συντακτικό αναδρομικό αναλυτή Top-Down,ο οποίος αναγνωρίζει την ανάθεση μεταβλητών και εκτύπωση δυαδικών αριθμητικών εκφράσεων που περιέχουν δυαδικές αριθμητικες τιμές,αναγνωριστικά ονόματα μεταβλητών,συνδυασμούς δυαδικής λογικής σε επίπεδο bit : AND, OR, XOR.

Πίνακας με τα First και Follow sets για όλα τα μη τερματικά σύμβολα

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
XOR	∅	∅	no	no
Term_tail	∅	id print	yes	yes
Term	(Expr) id number	id print	no	yes
OR	∅	∅	no	no
Factor_tail	∅	id print	yes	yes
Factor	(Expr) id number	id print	no	yes
AND	∅	∅	no	no
Atom_tail	∅	id print	yes	yes
Atom	(Expr) id number	id print	no	yes

The grammar is LL(1).

Αποτελέσματα εξόδου για έγκυρες μορφές εισόδου

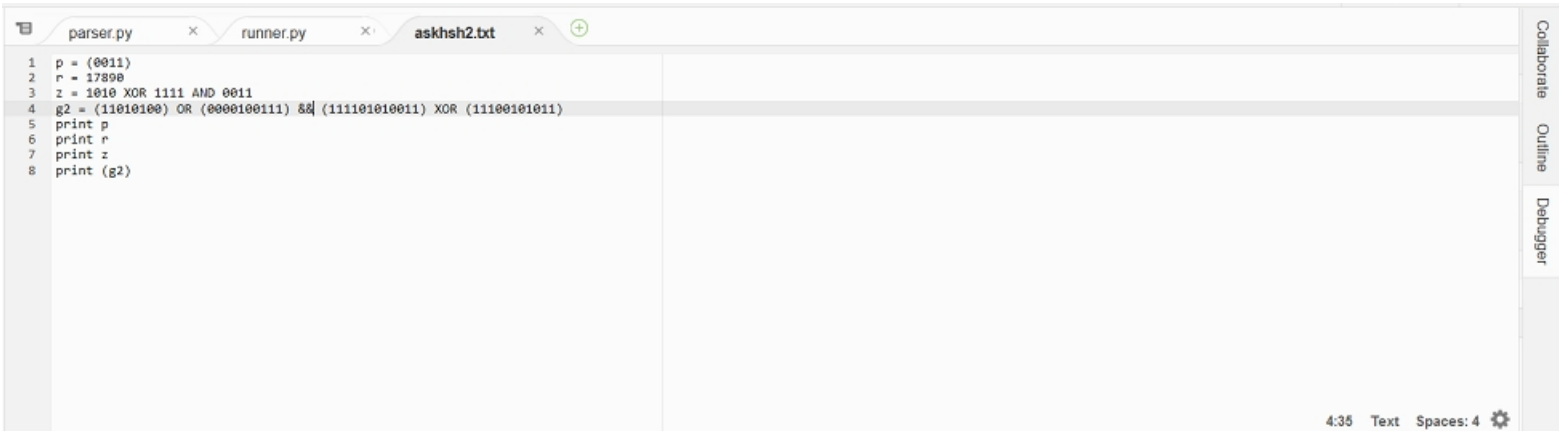



```
Important: use os.getenv(PORT, 8080) as the port and os.getenv(IP, 0.0.0.0) as the host in your scripts!

11
1010
1001
11111111100


Process exited with code: 0
```

Αποτελέσματα εξόδου για άκυρες μορφές εισόδου




```
  i=self.atom()
File "/home/ubuntu/workspace/compilers1819a2-master/runner.py", line 127, in atom
  self.match('bit_token')
File "/home/ubuntu/workspace/compilers1819a2-master/runner.py", line 40, in match
  self.la,self.text=self.next_token()
File "/home/ubuntu/workspace/compilers1819a2-master/runner.py", line 36, in next_token
  return self.scanner.read()
File "/usr/local/lib/python2.7/dist-packages/plex/scanners.py", line 94, in read
  self.text, action = self.scan_a_token()
File "/usr/local/lib/python2.7/dist-packages/plex/scanners.py", line 138, in scan_a_token
  raise errors.UnrecognizedInput(self, self.state_name)
plex.errors.UnrecognizedInput: '', line 2, char 5: Token not recognised in state ''
```