

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: {e, 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 αριθμητικές 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	\emptyset	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id number	id print	no	yes
XOR	\emptyset	\emptyset	no	no
Term_tail	\emptyset	id print	yes	yes
Term	(Expr) id number	id print	no	yes
OR	\emptyset	\emptyset	no	no
Factor_tail	\emptyset	id print	yes	yes
Factor	(Expr) id number	id print	no	yes
AND	\emptyset	\emptyset	no	no
Atom_tail	\emptyset	id print	yes	yes
Atom	(Expr) id number	id print	no	yes

The grammar is LL(1).

Αποτελέσματα εξόδου για έγκυρες μορφές εισόδου

The screenshot shows a code editor with three tabs: `parser.py`, `runner.py`, and `askhsh2.txt`. The `askhsh2.txt` tab is active, displaying the following Python code:

```
1 p = (0011)
2 r = 1010
3 z = 1010 XOR 1111 AND 0011
4 g2 = (11010100) OR (0000100111) AND (111101010011) XOR (11100101011)
5 print p
6 print r
7 print z
8 print (g2)|
```

On the right side of the editor, there are buttons for `Collaborate`, `Outline`, and `Debugger`. At the bottom right of the editor, the status bar shows `8:11 Text Spaces: 4` and a settings gear icon.

Below the editor, there is a terminal window titled `compilers1819a2`. It contains the following text:

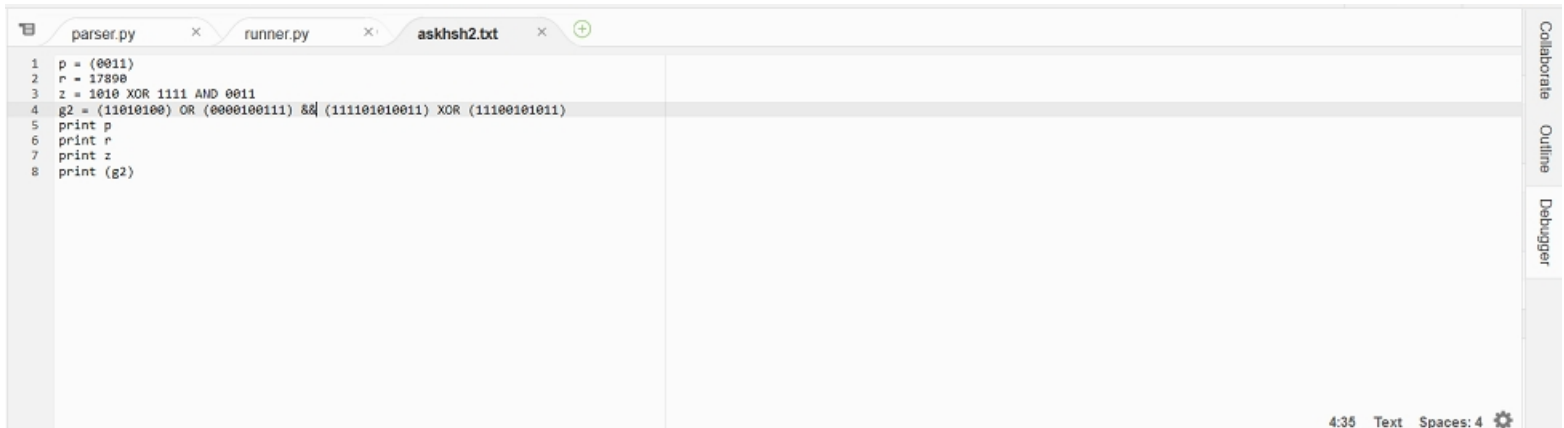
```
Run Command: compilers1819a2-master/runner.py Runner: Python 3 CWD ENV
```

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



The screenshot shows a code editor with three tabs: `parser.py`, `runner.py`, and `askhsh2.txt`. The `askhsh2.txt` tab is active, displaying the output of a Python script. The script in `parser.py` defines variables `p`, `r`, `z`, and `g2` based on binary values and bitwise operations, then prints them. The output in `askhsh2.txt` shows the execution of these operations, resulting in the values 17898, 1010, and 111010100.

```
1 p = (0011)
2 r = 17898
3 z = 1010 XOR 1111 AND 0011
4 g2 = (11010100) OR (0000100111) && (111101010011) XOR (11100101011)
5 print p
6 print r
7 print z
8 print (g2)
```

4:35 Text Spaces: 4

```
File "/home/ubuntu/workspace/compilers1819a2-master/runner.py", line 100, in next_token
    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 ''
```