Μεταγλωτιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Αντρέας Γεωργίου

ΑΜ: Π2016167

Κανόνες Γραμματικής

Οι κανόνες τις γραμματικής μας έχουν δοθεί στο εργαστήριο, αυτούς πρέπει να εφαρμόσουμε στην εργασία #2 με την μέθοδο της αναδρομικής κατάβασης. Αυτή η γραμματική δέχεται συμβολοσειρές με bits (0,1) και λογικές πράξεις. Η σειρά προτεραιότητας των τελεστών πράξεων σε επίπεδο bits είναι από την υψηλότερη προς την χαμηλότερη δηλαδή and, or , xor.

Γι' αυτό αναγκάστηκα πρέπει να έχουμε 3 επίπεδα κανόνων. Το and στο term_tail, το or στο factor_tail, και τέλος το xor στο atom_tail για να κρατήσουμε την σειρά προτεραιότητας.

Έλεγχος LL(1) για συμβατότητα για την γραμματική

Grammar

```
Stmt_list →   Stmt Stmt_list
            | .
Stmt →        id equal Exp
            | print Exp.
Exp →         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→         lparen Exp rparen
            | id
            | num.
```

Some sentences generated by this grammar: {ε, print id, print num, id equal id, id equal num, print id and id, print id and num, print num and id, print num and num, id equal id and id, id equal num and id, id equal id and num, id equal num and num, id equal id and id and id, id equal id and id and num, id equal id and num and id, id equal num and id and id, id equal id and num and num, id equal num and id and num, id equal num and num and id}

- 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.

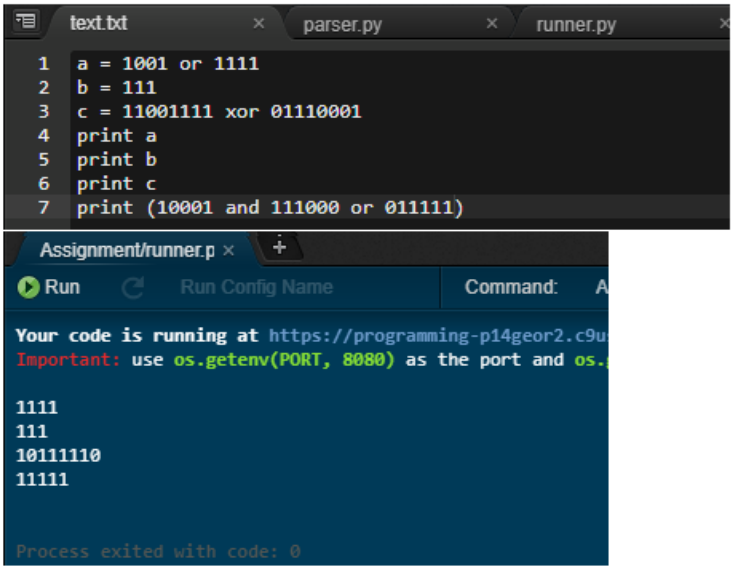
Αποτελέσματα Ελέγχου

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	∅	yes	yes
Stmt	id print	id print	no	yes
Term_tail	xor	rparen id print	yes	yes
Term	lparen id num	rparen xor id print	no	yes
Factor_tail	or	rparen xor id print	yes	yes
Factor	lparen id num	rparen or xor id print	no	yes
Atom_tail	and	rparen or xor id print	yes	yes
Atom	lparen id num	rparen and or xor id print	no	yes
Exp	lparen id num	rparen id print	no	yes

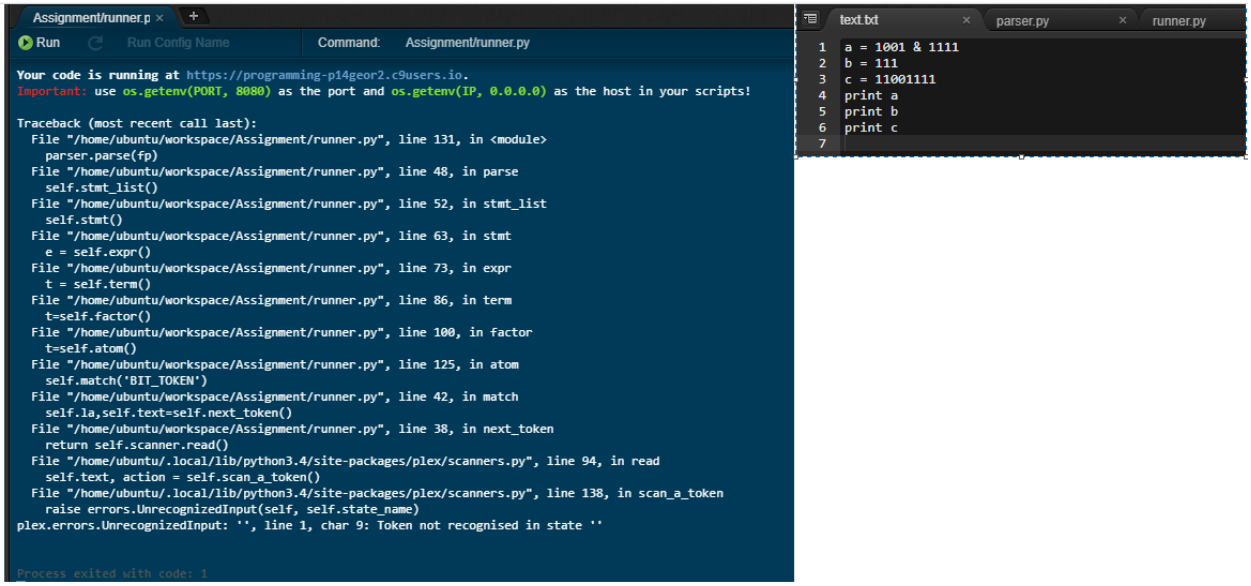
The grammar is LL(1).

Αποτελέσματα εξόδου

1) Έγκυρη μορφή εισόδου - εκτέλεση runner.py



2) Άκυρη μορφή εισόδου - εκτέλεση runner.py



```
Assignment/runner.p ×
Run   Run Config Name   Command: Assignment/runner.py

Your code is running at https://programming-p14geor2.c9users.io.
Important: use os.getenv(PORT, 8080) as the port and os.getenv(IP, 0.0.0.0) as the host in your scripts!

Traceback (most recent call last):
  File "/home/ubuntu/workspace/Assignment/runner.py", line 131, in <module>
    parser.parse(fp)
  File "/home/ubuntu/workspace/Assignment/runner.py", line 48, in parse
    self.stmt_list()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 52, in stmt_list
    self.stmt()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 63, in stmt
    e = self.expr()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 73, in expr
    t = self.term()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 86, in term
    t=self.factor()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 100, in factor
    t=self.atom()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 125, in atom
    self.match('BIT_TOKEN')
  File "/home/ubuntu/workspace/Assignment/runner.py", line 42, in match
    self.la,self.text=self.next_token()
  File "/home/ubuntu/workspace/Assignment/runner.py", line 38, in next_token
    return self.scanner.read()
  File "/home/ubuntu/.local/lib/python3.4/site-packages/plex/scanners.py", line 94, in read
    self.text, action = self.scan_a_token()
  File "/home/ubuntu/.local/lib/python3.4/site-packages/plex/scanners.py", line 138, in scan_a_token
    raise errors.UnrecognizedInput(self, self.state_name)
plex.errors.UnrecognizedInput: '', line 1, char 9: Token not recognised in state ''

Process exited with code: 1
```

```
text.txt × parser.py × runner.py
1 a = 1001 & 1111
2 b = 111
3 c = 11001111
4 print a
5 print b
6 print c
7
```