

Αναφορά

Μεταγλωττιστές 2018

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο : Ιωάννης Αγγέλης

A.M: Π2015006

A) Η γραμματική η οποία μας ζητήθηκε να κατασκευάσουμε με τη μέθοδο της αναδρομικής κατάβαση είναι για να αναγνωρίζει την γραμματική BOOLEAN. Η γραμματική αυτή πρέπει να μπορεί να εκφράσει τις μορφές :

$\alpha = T$, $\beta = \alpha \text{ and } F$, $c = \text{not } a \dots$ κ.λπ. .

Η γραμματική θα πρέπει να τηρεί και την προτεραιότητα των τελεστών με της παρενθέσεις () μετά το not είναι μοναδιαίος τελεστής , το and ο οποίος είναι διάδικος τελεστής και στο τέλος ο τελεστής or ο οποίος είναι και αυτός διάδικος τελεστής.

B) Η γραμματική η οποία παράχθηκε και επιτρέπει την προτεραιότητα των τελεστών είναι η ακόλουθη :

GRAMMAR

Stmt_list **->** ***Stmt Stmt_list | ε .***

Stmt **->** ***id assign Expr | print Expr .***

Expr **->** ***ATerm ATerm_tail.***

ATerm_tail **->** ***OrOp ATerm ATerm_tail | ε.***

ATerm **->** ***BTerm BTerm_tail.***

BTerm_tail **->** ***AndOp BTerm BTerm_tail | ε.***

BTerm **->** ***Factor | Notop Factor .***

Factor **->** ***(Expr) | id | number | ε.***

OrOp **->** ***or .***

AndOp **->** ***and .***

Notop **->** ***not .***

B) Η παραπάνω γραμματική έχει ελεγχθεί και δεν διαθέτει αριστερή αναδρομή καθώς και με το εργαλείο το οποίο προτάθηκε [Grammar](#)

- All nonterminals are reachable and realizable.
- The nullable nonterminals are: Stmt_list ATerm_tail BTerm_tail.
- The endable nonterminals are: Factor BTerm_tail BTerm ATerm Expr 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
Expr	(Expr) id number not	id print	no	yes
ATerm_tail	or	id print	yes	yes
ATerm	(Expr) id number not	or id print	no	yes
BTerm_tail	and	or id print	yes	yes
BTerm	(Expr) id number not	and or id print	no	yes
Factor	(Expr) id number	and or id print	no	yes
OrOp	or	(Expr) id number not	no	no
AndOp	and	(Expr) id number not	no	no
Notop	not	(Expr) id number	no	no

The grammar is LL(1).

Το number περιέχει το T 'η t 'η true 'η TRUE 'η 1 'η F 'η f 'η false 'η FALSE 'η 0 για να μπορεί να διαβάζεται ευδιάκριτα η γραμματική

Γ)

Το αρχείο parser.py περιέχει ένα μέρος από τον κώδικα του runner.py . Ο parser χωρίζει το κείμενο το οποίο του δίνεται σε token τα οποία μπορεί να αναγνωρίσει. Για την υλοποίηση του χρησιμοποιήθηκε η μέθοδος τις αναδρομικής κατάβασης με την χρήση του module Plex της Python. Παρακάτω μερικές εκφράσεις που αναγνωρίζει.

```
jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 parser.py
jonh_aggelis:~/workspace/METAGLOTISTES_2 $
```

```

1 a = 1 or false
2 print a
3 b = true and false
4 print a and b
5 g = not 0
6 print g
7 k = a and (b or g )
8 print k
9 print k and b
10
```

Τώρα κάτι που δεν αναγνωρίζει

```
jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 parser.py
PERIMENO AND || OR
jonh_aggelis:~/workspace/METAGLOTISTES_2 $
```

```
Traceback (most recent call last):
  File "parser.py", line 208, in <module>
    parser.parse(fp)
  File "parser.py", line 199, in parse
    self.stmtlist()
  File "parser.py", line 72, in stmtlist
    self.stmt()
  File "parser.py", line 84, in stmt
    self.expr()
  File "parser.py", line 95, in expr
    self.ATermTail()
  File "parser.py", line 104, in ATermTail
    self.orop()
  File "parser.py", line 168, in orop
    self.match('OR')
  File "parser.py", line 189, in match
    self.la,self.val = self.next_token()
  File "parser.py", line 195, in next_token
    return self.scanner.read()
  File "/usr/local/lib/python3.4/dist-packages/plex/scanners.py", line 94,
in read
    self.text, action = self.scan_a_token()
  File "/usr/local/lib/python3.4/dist-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 ''
```

```
jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 parser.py
PERIMENO not
jonh_aggelis:~/workspace/METAGLOTISTES_2 $
```

Δ) Ο runner δεν αναγνωρίζει ότι και ο parser . Επείγεις έχει προσφερθεί μια global μεταβλητή μέσα στην κλάση για να μπορώ να ανεβάζω προς τα πάνω τα not. Ο κώδικας έχει δοθεί στα εργαστήρια του σε ένα προγενέστερο στάδιο και με της κατάλληλες μετατροπές τροποποιήθηκε για να μπορεί να αναγνωρίζει την παραπάνω γραμματική. Ο υπολογισμός των εκφράσεων και των ανέσεων γίνονται με την βοήθεια ανάδρομης . Παρακάτω ο υπολογισμός μερικών εκφράσεων.

Αποδεκτά :

```
jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 runner.py
TRUE
FALSE
TRUE
TRUE
FALSE
jonh_aggelis:~/workspace/METAGLOTISTES_2 $
```

Μη αποδεκτά :

```
self.stmtList()
File "runner.py", line 73, in stmtList
    self.stmt()
File "runner.py", line 86, in stmt
    self.st[varname] = self.expr()
File "runner.py", line 96, in expr
    e = self.ATerm()
File "runner.py", line 131, in ATerm
    Bt = self.BTerm()
File "runner.py", line 172, in BTerm
    f = self.factor()
File "runner.py", line 194, in factor
    raise RuntimeError('inisialize variable ' + varname )
main_.RunError: inisialize variable k

1 a = k and true
2 print a
3
4

jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 runner.py
PERIMENO AND || OR
jonh_aggelis:~/workspace/METAGLOTISTES_2 $

1 a = 1 and (false or true) not true
2 print a
3

jonh_aggelis:~/workspace/METAGLOTISTES_2 $ python3 runner.py
PERIMENO not
jonh_aggelis:~/workspace/METAGLOTISTES_2 $

1 a = 1 and or true
2 print a
3
```

Reference

[LL parser](#)

[Compiler Design](#)

[Context Free Grammars](#)