

## Μεταγλωτιστές 2018

### Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Παναγιώτα Καραντζή

ΑΜ: Π2015130

## ΚΑΝΟΝΕΣ ΓΡΑΜΜΑΤΙΚΗΣ

Οι κανόνες της γραμματικής που κατέληξα φαίνονται παρακάτω. Συμβουλευτήκα το σχετικό κώδικα στο github καθώς και τις σημειώσεις από ασκήσεις που έγιναν στο εργαστήριο.

```
<Program> → Stmt_list #  
Stmt_list → Stmt Stmt_list | ε  
Stmt → id = Expr | print Expr  
Expr → Term Term_tail  
Term_tail → AndOrOp Term Term_tail | ε  
Term → Factor Factor_tail  
Factor_tail → NotOp Factor Factor_tail | ε  
Factor → (Expr) | id | Boolean  
AndOrOp → and | or  
NotOp → not  
Boolean → true | false | t | f | 0 | 1
```

Η γραμματική επαληθεύτηκε με το εργαλείο που προτάθηκε (<http://smlweb.cpsc.ucalgary.ca/start.html>) και είναι LL(1).

Grammar	
Stmt_list →	Stmt Stmt_list  .
Stmt →	id assign Expr   print Expr. .
Expr →	Term Term_tail. .
Term_tail →	AndOrOp Term Term_tail  .
Term →	Factor Factor_tail. .
Factor_tail →	NotOp Factor Factor_tail  .
Factor →	(Expr)   id   Boolean. .
NotOp →	not. .
AndOrOp →	and. .
Boolean →	true   false   t   f   0   1. .

Παρακάτω φαίνονται τα FIRST και FOLLOW sets που προέκυψαν.

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	∅	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id true false t f 0 1	id print	no	yes
Term_tail	and	id print	yes	yes
Term	(Expr) id true false t f 0 1	and id print	no	yes
Factor_tail	not	and id print	yes	yes
Factor	(Expr) id true false t f 0 1	not and id print	no	yes
NotOp	not	(Expr) id true false t f 0 1	no	no
AndOrOp	and	(Expr) id true false t f 0 1	no	no
Boolean	true false t f 0 1	not and id print	no	yes

The grammar is LL(1).

## ΚΩΔΙΚΑΣ

Έκανα μόνο το συντακτικό αναλυτή – αναγνωριστή της γλώσσας και ο κώδικας βρίσκεται στο αρχείο parser.py. Χρησιμοποιείται το module plex για την υλοποίησή του.

Αρχικά, ορίζεται η κλάση ParseError για την περίπτωση του λάθους. Έπειτα ορίζεται η κλάση MyParser η οποία περιέχει τη συνάρτηση create\_scanner.

Στη συνάρτηση create\_scanner προστέθηκαν για τα αναγνωριστικά: `identifier = letter+ plex.Rep(letter|digit)` όπου το Rep αντιστοιχεί μηδέν ή περισσότερες επαναλήψεις ενός γράμματος ή ψηφίου, το '=' ως string, οι παρενθέσεις , το keyword 'print' και οι τελεστές and,or,not ως string.

Για τις λογικές τιμές χρησιμοποίησα ο NoCase επειδή ζητήθηκε οι τιμές να είναι ανεξάρτητες από κεφαλαία-πεζά και ο NoCase constructor αντιμετωπίζει τα κεφαλαία και τα μικρά γράμματα ως ισοδύναμα (case-insensitive). Στη συνέχεια, δημιουργήθηκε το κατάλληλο λεξικό και ο scanner προσπελαύνει το αρχείο και δημιουργούνται οι self.la και self.val. Η match ελέγχει για την ύπαρξη token. Έπειτα, όρισα τις κατάλληλες συναρτήσεις σύμφωνα με τη γραμματική μου και τα first-follow sets.

Τέλος, δημιουργείται ένα αντικείμενο της κλάσης myParser.

Πηγές που χρησιμοποιήθηκαν:

<https://pythonhosted.org/plex/tutorial.html>

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

<https://gist.github.com/mixstef/946fce67f49f147991719bfa4d0101fa>