

Μεταγλωτιστές 2018

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Λάππα Πολυχρονία

AM: 2014103

Οι κανόνες γραμματικής που κατέληξα είναι οι εξής:

`<Program> -> Stmt_list#`

`Stmt_list -> Stmt Stmt_list | e`

`Stmt -> id assign Expr | print Expr`

`Expr -> Term Term_tail`

`Term_tail -> AndOrOp Term Term_tail | e`

`Term -> Factor Factor_tail`

`Factor_tail -> NotOp Factor Factor_tail | e`

`Factor -> (Expr) | id | Boolean`

`AndOrOp -> and | or`

`NotOp -> not`

`Boolean -> true | false | t | f | 0 | 1`

Η γραμματική επαληθεύτηκε με το εργαλείο που μας προτείνετε

(<http://smlweb.cpsc.ucalgary.ca/start.html>) και αποδείχτηκε ότι 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.

Πίνακες 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 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).

Στην εργασία κατάφερα να υλοποιήσω μόνο τον συντακτικό αναλυτή και ο κώδικας βρίσκεται στο αρχείο `parser.py`. Πρώτα ορίζεται η κλάση `ParseError` σε περίπτωση που γίνει κάποιο λάθος και στην συνέχεια ορίζεται η κλάση `MyParser`. Στην `create_scanner` πρόσθεσα τα εξής χαρακτηριστικά : `identifier= letter+ plex.Rep(letter|digit)`, το `=` ως `string`, τις παρενθέσεις, το `keyword print` καθώς και τους τελεστές `and,or,not`. Για τις λογικές τιμές χρησιμοποίησα το `NoCase constructor` καθώς αντιμετωπίζει κεφαλαία και μικρά γράμματα σαν ισοδύναμα. Έπειτα δημιούργησα το λεξικό που θεώρησα ότι είναι το κατάλληλο.