

Μεταγλωτιστές 2018

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Πιπλικάτσης Αθανάσιος

ΑΜ: Π2015038

Οι κανόνες της γραμματικής είναι:

η γραμματική δεν έχει αριστερή αναδρομή ούτε κοινούς παράγοντες στους κανόνες επιτρέπονται κενές παραγωγές (μέσω του ϵ)

Stmt_list	->	Stmt Stmt_list ϵ
Stmt	->	id = Expr print Expr
Expr	->	Term Term_tail
Term_tail	->	Not Term Term_tail ϵ
Term	->	Factor Factor_tail
Factor_tail	->	AndOr Factor Factor_tail ϵ
Factor	->	(Expr) id truefalse
Not	->	not ϵ
AndOr	->	and or

To First και Follow set αντίστοιχα:

$\text{First}(\text{Stmt_list}) = \text{First}(\text{Stmt}) \cup \{\epsilon\} = \{\text{id}, \text{print}, \epsilon\}$
 $\text{First}(\text{Stmt}) = \{\text{id}, \text{print}\}$
 $\text{First}(\text{Expr}) = \text{First}(\text{Term}) = \{(\text{id}, \text{truefalse})\}$
 $\text{First}(\text{Term_tail}) = \text{First}(\text{Not}) \cup \{\epsilon\} = \{\text{not}, \epsilon\}$
 $\text{First}(\text{Term}) = \text{First}(\text{Factor}) = \{(\text{id}, \text{truefalse})\}$
 $\text{First}(\text{Factor_tail}) = \text{First}(\text{AndOr}) \cup \{\epsilon\} = \{\text{and}, \text{or}, \epsilon\}$
 $\text{First}(\text{Factor}) = \{(\text{id}, \text{truefalse})\}$
 $\text{First}(\text{Not}) = \{\text{not}, \epsilon\}$
 $\text{First}(\text{AndOr}) = \{\text{and}, \text{or}\}$

$\text{Follow}(\text{Stmt_list}) = \{\$ \}$
 $\text{Follow}(\text{Stmt}) = \text{Follow}(\text{Stmt_list}) = \{\$ \}$
 $\text{Follow}(\text{Expr}) = \text{Follow}(\text{Stmt}) = \{ \}$
 $\text{Follow}(\text{Term_tail}) = \text{Follow}(\text{Expr}) = \{ \}$
 $\text{Follow}(\text{Term}) = \text{First}(\text{Term_tail}) = \{\text{not}\}$
 $\text{Follow}(\text{Factor_tail}) = \text{Follow}(\text{Term}) = \{\text{not}\}$
 $\text{Follow}(\text{Factor}) = \text{First}(\text{Factor_tail}) = \{\text{and}, \text{or}\}$
 $\text{Follow}(\text{Expr}) = \{ \}$

Το parser.py αρχείο περιέχει την υλοποίηση των ζητούμενων με το module plex. Μπορεί να αναγνωρίσει μια λογική έκφραση (boolean) καθώς και τα σημεία που είναι λάθος. Παντα τερματίζει με error message (ακόμα και τα σωστά) υπονοώντας ότι θα μπορούσε να συνεχίσει παραπέρα η έκφραση. Μπορούμε να καταλάβουμε αν είναι valid μια έκφραση συγκρίνοντας το test.txt με όλα τα found "Tokens".

Δοκιμή του κώδικα με διάφορες τιμές:

```
thanos:plex$ cat test.txt
a = TrUe and 0thanos:plex$
thanos:plex$ python3 erg2.py
found identifier
found Equal Sign
found boolean Value
found operation
found boolean Value
Parser Error: Expected operation (not, and, or at line 1 char 15
thanos:plex$ cat "test.txt"
a = TrUe and 0thanos:plex$ clear
thanos:plex$ cat "test.txt"
a = TrUe and 0thanos:plex$ python3 erg2.py
found identifier
found Equal Sign
found boolean Value
found operation
found boolean Value
Parser Error: Expected operation (not, and, or at line 1 char 15
thanos:plex$ cat "test.txt"
foo = true = 1thanos:plex$ python3 erg2.py
found identifier
found Equal Sign
found boolean Value
Parser Error: Expected operation (not, and, or at line 1 char 12
thanos:plex$ cat "test.txt"
1a = False and Truethanos:plex$ python3 erg2.py
Parser Error: Expected identifier or print keyword at line 1 char 1
thanos:plex$ cat "test.txt"
false or truethanos:plex$ python3 erg2.py
Parser Error: Expected identifier or print keyword at line 1 char 1
thanos:plex$ cat "test.txt"
b      =      true and false and C or k thanos:plex$ python3 erg2.py
found identifier
found Equal Sign
found boolean Value
found operation
found boolean Value
found operation
found Identifier
found operation
found Identifier
Parser Error: Expected operation (not, and, or at line 1 char 42
```