Μεταγλωτιστές 2018

Προγραμματιστική άσκηση #2

Παρασκευή Μέμου

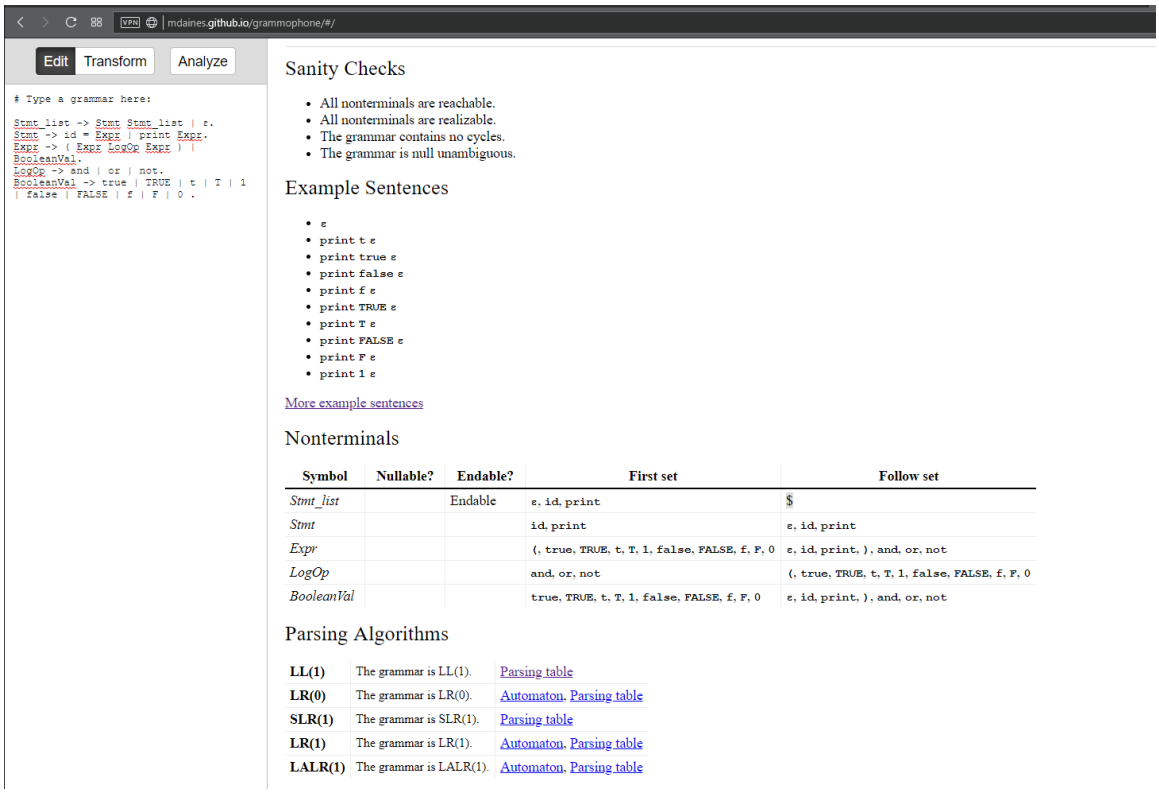
Π2013092

Η ανάπτυξη και η εκτέλεση της εργασίας πραγματοποιήθηκε σε περιβάλλον Windows 10 κάνοντας χρήση του module plex με τη βοήθεια του εργαλείου virtualenv.

Η γραμματική που σχεδιάστηκε είναι η παρακάτω:

```
Stmt_list -> Stmt Stmt_list | ε.

Stmt -> id = Expr | print Expr.

Expr -> ( Expr LogOp Expr ) | BooleanVal.

LogOp -> and | or | not.

BooleanVal -> true | TRUE | t | T | 1 | false | FALSE | f | F | 0 .
```

Με την έκφραση «LogOp» συμβολίζονται οι λογικοί τελεστές and, or, not, ενώ με την έκφραση BooleanVal οι λογικές τιμές. Η γραμματική είναι LL(1) όπως φαίνεται από την παρακάτω εικόνα. Το online εργαλείο που χρησιμοποιήθηκε για τον έλεγχο είναι το http://mdaines.github.io/grammophone



Τα FIRST και FOLLOW sets παρέχονται από το online εργαλείο και παρουσιάζονται στην παρακάτω εικόνα σε πιο κοντινό screenshot για να είναι πιο ευανάγνωστα. Το id στον κώδικα Python μεταφράστηκε περαιτέρω σε string που μπορεί να περιέχει οποιοδήποτε συνδυασμό γραμμάτων και ψηφίων ώστε να δημιουργούνται ονόματα μεταβλητών.

Symbol	Nullable?	Endable?	First set	Follow set
Stmt_list		Endable	ε, id, print	$
Stmt			id, print	ε, id, print
Expr			(, true, TRUE, t, T, 1, false, FALSE, f, F, 0	ε, id, print,), and, or, not
LogOp			and, or, not	(, true, TRUE, t, T, 1, false, FALSE, f, F, 0
BooleanVal			true, TRUE, t, T, 1, false, FALSE, f, F, 0	ε, id, print,), and, or, not

Δίνοντας σαν είσοδο το αρχείο που φαίνεται στο ακόλουθο screenshot

recursive-descent-parsing.txt

```
1  shouldBeTrue1 = (TRUE and t) or (False and F)
2  print shouldBeTrue1
3  shouldBeTrue2 = (1 or (0 and 1))
4  print shouldBeTrue2
5  shouldBeFalse1 = 0 and 1
6  print shouldBeFalse1
7  shouldBeFalse2 = not 1
8  print shouldBeFalse2
9  shouldBeFalse3 = (((true and false) or 0 ) and 0 )
10 print shouldBeFalse3
11 thisshouldntwork = (true and
12
```

Το πρόγραμμα μας παράγει την ακόλουθη έξοδο ανιχνεύοντας σωστά ότι η έκφραση στην τελευταία γραμμή δεν είναι σωστή δεν κλείνειη παρένθεση.

```
val in logOp: or
val in expr: 0
val in bool: 0
val in expr: )
val in expr: and
val in logOp: and
val in expr: 0
val in bool: 0
val in expr: )
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse3
val in stmt: thisshouldntwork
val in expr: =
val in assignOp: =
val in expr: (
val in expr: true
val in bool: true
val in expr: and
val in logOp: and
val in expr:
None

Parser Error: in expr: Valid expression expected at line 12 char 1

(venv)  0:21:27.51
C:\Project\Python\CompilersVirtualEnv>
```

Αντίστοιχα έχουμε ανίχνευση σφάλματος στα παρακάτω προγράμματα

recursive-descent-parsing.txt

```
1  shouldBeTrue1 = (TRUE and t) or (False and F)
2  print shouldBeTrue1
3  shouldBeTrue2 = (1 or (0 and 1))
4  print shouldBeTrue2
5  shouldBeFalse1 = 0 and 1
6  print shouldBeFalse1
7  shouldBeFalse2 = not 1
8  print shouldBeFalse2
9  shouldBeFalse3 = (((true and false) or 0 ) and 0 )
10 print shouldBeFalse3
11 thisshouldntwork = true or )
12
```

```
val in expr: 0
val in bool: 0
val in expr: )
val in expr: and
val in logOp: and
val in expr: 0
val in bool: 0
val in expr: )
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse3
val in stmt: thisshouldntwork
val in expr: =
val in assignOp: =
val in expr: true
val in bool: true
val in expr: or
val in logOp: or
val in expr: )
val in expr:
None

Parser Error: in expr: Valid expression expected at line 12 char 1
```

recursive-descent-parsing.txt

```
1  shouldBeTrue1 = (TRUE and t) or (False and F)
2  print shouldBeTrue1
3  shouldBeTrue2 = (1 or (0 and 1))
4  print shouldBeTrue2
5  shouldBeFalse1 = 0 and 1
6  print shouldBeFalse1
7  shouldBeFalse2 = not 1
8  print shouldBeFalse2
9  shouldBeFalse3 = (((true and false) or 0 ) and 0 )
10 print shouldBeFalse3
11 print
12 thisshouldntworkalso =
13 thisshouldntwork = (true and
14
```

```
val in logOp: and
val in expr: false
val in bool: false
val in expr: )
val in expr: or
val in logOp: or
val in expr: 0
val in bool: 0
val in expr: )
val in expr: and
val in logOp: and
val in expr: 0
val in bool: 0
val in expr: )
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse3
val in stmt: print
-----
match exception
PRINT_COMMAND
print
-----
Parser Error: found PRINT_COMMAND instead of string at line 11 char 1
```

recursive-descent-parsing.txt

```
1  shouldBeTrue1 = (TRUE and t) or (False and F)
2  print shouldBeTrue1
3  shouldBeTrue2 = (1 or (0 and 1))
4  print shouldBeTrue2
5  shouldBeFalse1 = 0 and 1
6  print shouldBeFalse1
7  shouldBeFalse2 = not 1
8  print shouldBeFalse2
9  shouldBeFalse3 = (((true and false) or 0 ) and 0 )
10 print shouldBeFalse3
11 thisshouldntworkalso =
12 print
13 thisshouldntwork = (true and
14
```

```
val in expr: (
val in expr: (
val in expr: true
val in bool: true
val in expr: and
val in logOp: and
val in expr: false
val in bool: false
val in expr: )
val in expr: or
val in logOp: or
val in expr: 0
val in bool: 0
val in expr: )
val in expr: and
val in logOp: and
val in expr: 0
val in bool: 0
val in expr: )
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse3
val in stmt: thisshouldntworkalso
val in expr: =
val in assignOp: =
PRINT_COMMAND
print
Parser Error: in assignOp: Assignment operator expected at line 12 char 1

(venv)  0:27:03.95
C:\Project\Python\CompilersVirtualEnv>
```

Στο πρόγραμμα χωρίς λάθη που ακολουθεί το πρόγραμμα μας τερματίζει κανονικά χωρίς σφάλματα

recursive-descent-parsing.txt

```
1  shouldBeTrue1 = (TRUE and t) or (False and F)
2  print shouldBeTrue1
3  shouldBeTrue2 = (1 or (0 and 1))
4  print shouldBeTrue2
5  shouldBeFalse1 = 0 and 1
6  print shouldBeFalse1
7  shouldBeFalse2 = not 1
8  print shouldBeFalse2
9  shouldBeFalse3 = (((true and false) or 0 ) and 0 )
10 print shouldBeFalse3
```

```
val in expr: 1
val in bool: 1
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse2
val in stmt: shouldBeFalse3
val in expr: =
val in assignOp: =
val in expr: (
val in expr: (
val in expr: (
val in expr: true
val in bool: true
val in expr: and
val in logOp: and
val in expr: false
val in bool: false
val in expr: )
val in expr: or
val in logOp: or
val in expr: 0
val in bool: 0
val in expr: )
val in expr: and
val in logOp: and
val in expr: 0
val in bool: 0
val in expr: )
val in expr: print
val in printCheck: print
val in printCheck: shouldBeFalse3
val in stmt:
You have reached the end of the program.
Successful parsing of program

(venv)  0:28:03.74
C:\Project\Python\CompilersVirtualEnv>
```

Τα μηνύματα που φαίνονται στην κονσόλα βοήθησαν κατά τη διαδικασία του debug και διατηρήθηκαν για την αναφορά για να φανεί η πορεία εκτέλεσης των εντολών και πως διαβάζεται το αρχείο εισόδου.

Στο αρχείο runner.py αφαιρέσαμε τα επιπλέον αυτά μηνύματα της κονσόλας και προσθέσαμε τις εντολές ώστε να γίνεται η εκτέλεση των εντολών και να εμφανίζεται το αποτέλεσμα. Χρησιμοποιήσαμε το προηγούμενο αρχείο όπου υπάρχουν 5 διαφορετικά παραδείγματα πράξεων ώστε να δούμε αν διερμηνεύεται σωστά το αποτέλεσμα. Για κάθε πράξη εμφανίζουμε την έκφραση που τη συνοδεύει μαζί με το αποτέλεσμα της. Όπως βλέπουμε στην επόμενη εικόνα ο διερμηνευτής παρουσιάζει τα σωστά αποτελέσματα, οι δύο πρώτες εκφράσεις επιστρέφουν 1/true και οι τρεις τελευταίες επιστρέφουν 0/false.