

ΜΕΤΑΓΛΩΤΙΣΤΕΣ 2018

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ #2

Ονοματεπώνυμο : Μαρία Δήμα

ΑΜ : Π2013029

Έτος : 2018

29 / 5 / 2018

Ερώτημα Α)

Βασιζόμενοι στη γραμματική των αριθμητικών εκφράσεων που χρησιμοποιήθηκε στο εργαστήριο, σχεδιάσαμε την ακόλουθη γραμματική:

<Program> -> Stmt_list # (βοηθητικός κανόνας!)

Stmt_list -> Stmt Stmt_list | ε

Stmt -> id = Expr | print Expr

Expr -> Term Term_tail

Term_tail -> ORop Term Term_tail | ε

Term -> Factor Factor_tail

Factor_tail -> ANDop Factor Factor_tail | ε

Factor -> Statmnt Statmnt_tail

Statmnt -> NOTop | ε

Statmnt_tail -> (Expr) | id | bool_value

ORop -> OR

ANDop -> AND

NOTop -> NOT

Συμπληρωματικά στοιχεία :

print: το keyword 'print'

id: όνομα μεταβλητής

bool_value: true, false, t, f, 0 και 1

Οι δύο πρώτοι κανόνες (Stmt_list και Stmt), έχουν διατηρηθεί όπως το ζητήθηκε στην άσκηση. Οι λογικές πράξεις είναι οι NOT, AND και OR και στην γραμματική ακολουθείται η σειρά προτεραιότητας των τελεστών. Η γραμματική είναι έτσι υλοποιημένη ώστε να οι λογικές εκφράσεις θα μπορούν να συνδυαστούν με παρενθέσεις σε οποιοδήποτε βάθος.

Ερώτημα Β)

Επιβεβαιώσαμε ότι η γραμματική είναι LL(1) χρησιμοποιώντας το εργαλείο που δίνεται στην σελίδα <http://smlweb.cpsc.ucalgary.ca/start.html>.

Στην συνέχεια παραθέτω τα START και FOLLOW SETs :

Κανόνας	FirstSET	FollowSET
Stmt_list	id, print, ε	#
Stmt	id, print	id, print, #
Expr	NOT, (, id, bool_value), id, print, #
Term_tail	OR, ε), id, print, #
Term	NOT, (, id, bool_value	OR,), id, print, #
Factor_tail	AND, ε	OR,), id, print, #
Factor	NOT, (, id, bool_value	AND, OR,), id, print, #
Statmnt	NOT, ε	(, id, bool_value
Statmnt_tail	(, id, bool_value	AND, OR,), id, print, #
ORop	OR	NOT, (, id, bool_value
ANDop	AND	NOT, (, id, bool_value
NOTop	NOT	(, id, bool_value

Ερώτημα Γ)

Χρησιμοποιώντας την βοηθητική [σελίδα](#) πήραμε τον κώδικα και υλοποιώντας έναν parser βασισμένο στην γραμματική που αναπτύξαμε προηγουμένως, μπορούμε να κάνουμε parsing οποιασδήποτε λογικής έκφρασης προκύπτει σύμφωνα με τα δεδομένα της άσκησης. Προκειμένου να ελέγξουμε τον parser

χρησιμοποιήσαμε το βοηθητικό αρχείο recursive-descent-parsing-bool.txt όπου εκεί παραθέταμε τις λογικές εκφράσεις που θέλαμε να ελέγξουμε. Ο parser είναι υλοποιημένος στο αρχείο parser.py

Ερώτημα Δ)

Σε ένα δεύτερο αντίγραφο του προγράμματός, το runner.py, μετά το parsing του κώδικα που δίνεται στο βοηθητικό αρχείο, εκτελούνται οι λογικές πράξεις και εκτυπώνεται το αποτέλεσμα.

Ερώτημα Ε)

Στο αρχείο log.txt παραθέτουμε ένα αποτέλεσμα από την εκτέλεση του runner.py. Όπου μπορούμε να δούμε ποια εντολή δόθηκε αρχικά, στην συνέχεια ότι έγινε με επιτυχία το parsing και δεν εμφανίζεται κάποιο μήνυμα λάθους και τέλος ότι εκτυπώνεται το αποτέλεσμα της λογικής έκφρασης.

ΤΕΛΟΣ ΑΝΑΦΟΡΑΣ