Το αλφάβητο της γραμματικής που μας περιγράφεται στην εκφώνηση του 4ου θέματος είναι το εξής:

Σ={{A-Z},{"+","-","*","/","%",";","=",{1-9},{a-z}}}

(με τα {..} τα αναπαριστώ ως σύνολα συμβόλων)

και οι κανόνες είναι(από τις περιγραφές 1-5 τις εκφώνησης):

P = {

[A-Z] -> "="{[0-9] | [a-z]}{ "+" | "-" | "*" | "/" | "%"} {[0-9] | [a-z]}{L|";"}

L-> {{ "+" | "-" | "*" | "/" | "%"} {[0-9] | [a-z]}L

}

(τερματικά τα ";" ,{"+","-","*","/","%"},[a-z],[1-9] και μη-τερματικά όλα τα άλλα σύμβολα)

(Το L υπάρχει στο αλφάβητο για να μπορεί ο 3ος κανόνας να επαναλαμβάνεται)

Συνεπώς βάση των παραδειγμάτων(από pdf <<Τρόποι Προσδιορισμού Σύνταξης 2017-2018) προκύπτουν οι περιγραφές <<BNF>> και <EBNF> καθώς και το συντακτικό διάγραμμα.

Σε BNF:

Για ευκολία στο γράψιμο και στην ανάγνωση θεωρώ τα εξής σύνολα:

A = [A-Z]

b = [a-z]

c = [0-9]

Και είναι:

<A> ::= "="(<c>|<b>)d(<c>|<b>)(<L>|";")

<L> ::= d(<c> | <b>)<L>

ΣΕ EBNF:

A = "="(c|b)d(c|b)({L}|";")

L = d(c|b)L

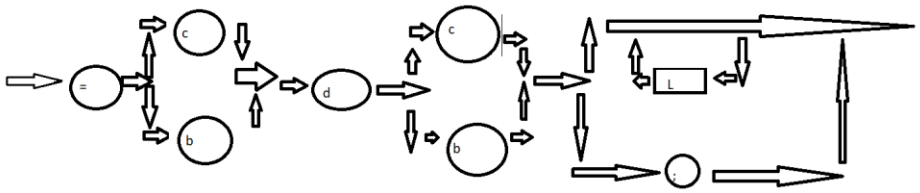
(Θα χρησιμοποιήσω τα ίδια σύνολα που θεώρησα στην αναπαράσταση BNF για ευκολία)

Σε Συντακτικό Διάγραμμα:

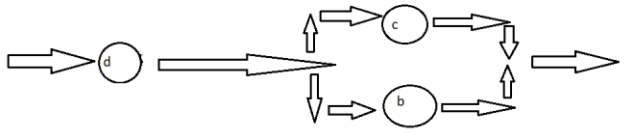
(Θα χρησιμοποιήσω τα ίδια σύνολα που θεώρησα στην αναπαράσταση BNF και EBNF για ευκολία)

Commented [CC1]:

Για A:



Για L



(Για την δημιουργία των διαγραμμάτων χρησιμοποίησα τα έτοιμα σχήματα από το paint)

Για το πρόγραμμα FLEX:

Σε Αυτό το κομμάτι υλοποίησα σύμφωνα με τα δεδομένα της εκφώνησης και το συντακτικό διάγραμμα ένα πρόγραμμα το οποίο δέχεται εκφράσεις και είτε τις αναγνωρίζει δηλώνοντάς <<Valid Expression>> είτε τις απορρίπτει δηλώνοντας τους χαρακτήρες της ως <<Unrecognized Characters>>.

Επεξήγηση κώδικα:

1ο Τμήμα flex κώδικα: Αυτό το τμήμα είναι κενό στο κομμάτι του αυτούσιου κώδικα C, καθώς στην υλοποίησή μου δεν χρειάστηκα κάποια βιβλιοθήκη της C κ.λπ. και έχω δηλώσει κάποια ονόματα τα οποία είναι:

- <<DIGIT>> το οποίο αντιπροσωπεύει τα ψηφία από 1 έως 9.
- <<VAR>> το οποίο αντιπροσωπεύει τα κεφαλαία γράμματα της αγγλικής αλφαβήτου από το <<A-Z>>.
- <<SYMB>> το οποίο αντιπροσωπεύει ένα από τα σύμβολα "+","~","*","/" τα οποία χωρίζονται με το λογικό <<ή>> της γλώσσας δηλαδή το όνομα επαληθεύεται όταν βρεθεί κάποιο από αυτό.
- <<var>> το οποίο αντιπροσωπεύει τα πεζά γράμματα της αγγλικής αλφαβήτου από το <<a-z>>.
- <<LOOPS>> το οποίο αντιπροσωπεύει την ακολουθία ενός από τα σύμβολα του <<SYMB>> ακολουθούμενο από ένα ψηφίο του <<DIGIT>> ή ένα γράμμα από το <<var>>.
- <<ACCEPTED_STARTS>> το οποίο αντιπροσωπεύει το σύνολο όλων των έγκυρων πιθανών εκφράσεων σύμφωνα με την εκφώνηση και χρησιμοποιεί τα παραπάνω ονόματα καθώς και λογικά <<ή>>.

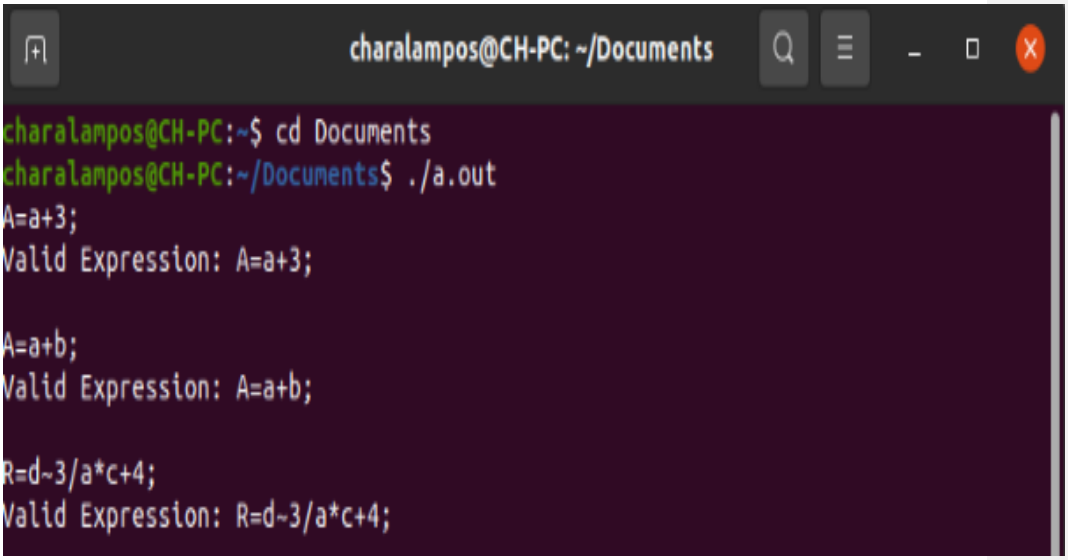
2ο Τμήμα flex κώδικα: Σε αυτό το τμήμα υπάρχουν 2 κανόνες**. 0 Πρώτος** ορίζει πως αν βρεθεί κάποια έκφραση που πληροί τις προϋποθέσεις για να είναι έγκυρη, σύμφωνα με το όνομα <<ACCEPTED_STARTS>>, τότε να εμφανίζεται στην οθόνη με το μήνυμα <<Valid Expression>>. **Ο Δεύτερος** ορίζει πως οποιαδήποτε έκφραση που δίνεται δεν πληροί τις προϋποθέσεις ώστε να είναι αποδεκτή δεν θα αναγνωρίζεται κανένας χαρακτήρας της και θα εμφανίζονται όλοι με το χαρακτηρισμό <<Unrecognized Character>>.

3ο Τμήμα flex κώδικα: Είναι αυτό που περιέχει τη <<main()>> και το <<yylex()>> περιέχει τις εντολές που μας έχουν διδαχθεί στις παραδόσεις για να δέχεται το <<input>>(την έκφραση από το χρήστη) και να καλεί με το <<yylex()>> το κώδικα που έχουμε γράψει στα πάνω τμήματα.

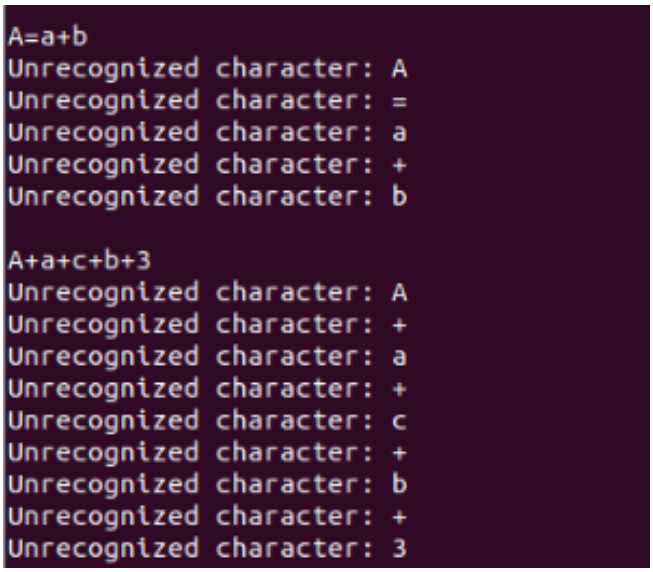
Περιγραφή Λειτουργίας:

Κατά την εκκίνηση του προγράμματος το τερματικό περιμένει την εισαγωγή της έκφρασης από τον χρήστη και υπάρχουν τα εξής ενδεχόμενα:

α)Η Έκφραση να γίνει αποδεκτή:



b)Η Έκφραση να μην είναι αποδεκτή και συνεπώς να μην αναγνωριστεί:



<u>Παραδοχές στην υλοποίηση:</u>

- Το σύμβολο που αντιπροσωπεύει την αφαίρεση σύμφωνα με την εκφώνηση(δηλαδή το “-”) το έχω αντικαταστήσει με το “~”.Συνεπώς τη πράξη της αφαίρεσης τη συμβολίζει το “~” και όχι το “-”

- Θεωρώ πως το όνομα <<VAR>> δηλαδή τα κεφαλαία είναι μη-τερματικά σύμβολα και με κάποιο αυτά πρέπει να ξεκινάει η κάθε αποδεκτή έκφραση, αντίθετα το όνομα <<var>> δηλαδή τα πεζά είναι τερματικά.

Οδηγίες Χρήσης/Manual:

- Ανοίγουμε το τερματικό και καθοδηγούμαστε στο φάκελο όπου βρίσκεται το αρχείο thema4.l
- Εκτελούμαι τις εξής εντολές με τη ροή που δείχνουν τα βελάκια(flex thema4.l → gcc lex.yy.c -lfl → ./a.out).
- Δίνουμε εκφράσεις στο πρόγραμμα και στο τερματικό μας αναγράφονται τα αντίστοιχα μνήματα.
- Για να κλείσουμε την εφαρμογή πατάμε την ίδια στιγμή ctrl+D στο πληκτρολόγιο μας.