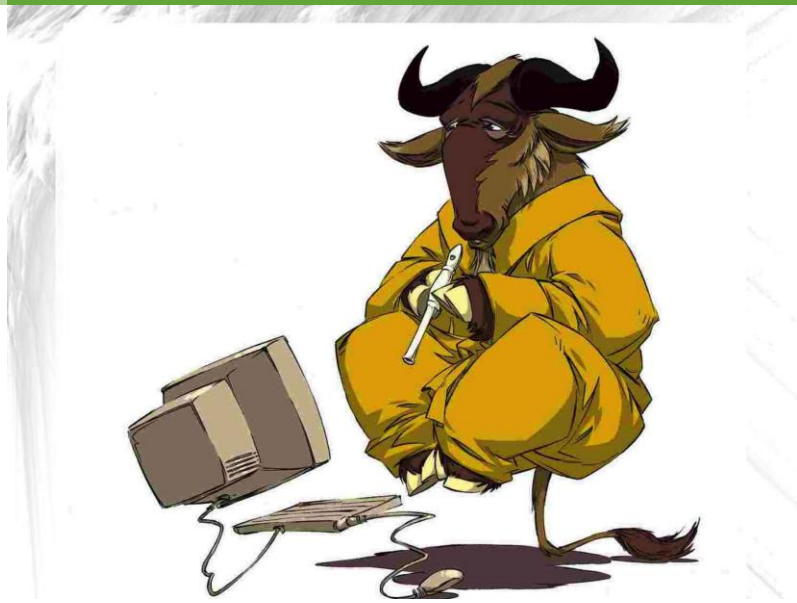
Εισαγωγή στο Bison

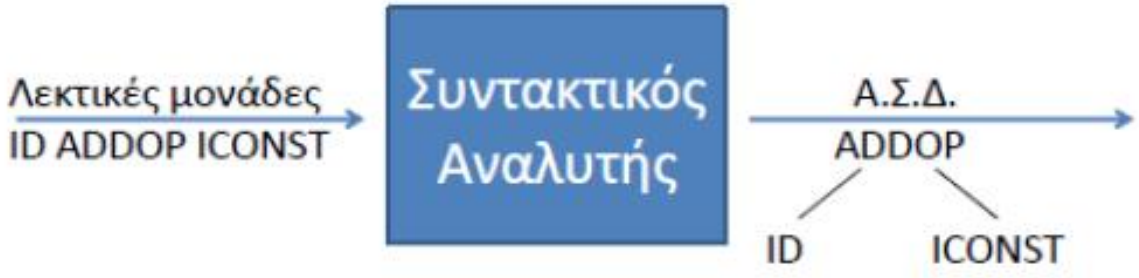
Μεταγλωττιστές,

Χειμερινό εξάμηνο 2018-2019



Συντακτική Ανάλυση

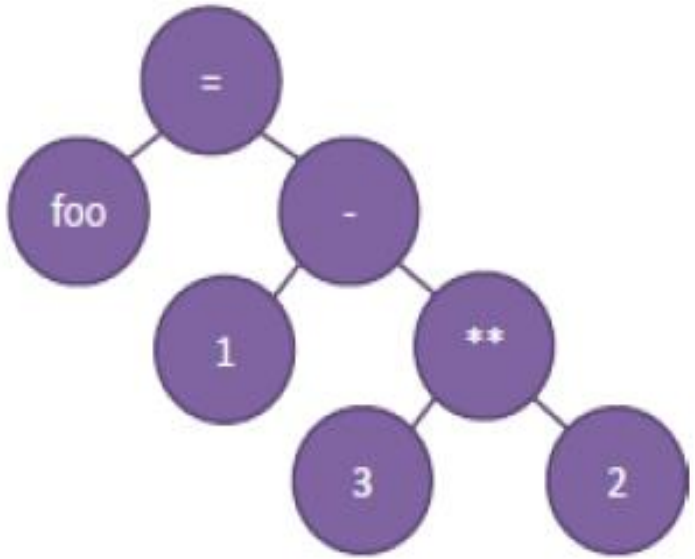
- Αποτελεί την δεύτερη φάση της μετάφρασης.
- Εύρεση της σχέσης που υπάρχει των λεκτικών μονάδων ενός προγράμματος.



Παράδειγμα

- Είναι ουσιαστικά η διαδικασία κατά την οποία αναλύεται μία σειρά από tokens ώστε να καθορίσει την δομή της γραμματικής.
- Τα συντακτικά λάθη προσδιορίζονται σε αυτό το στάδιο.

Lex	**Token type**
Foo	Variable
=	Assignment operator
1	Number
-	Subtraction operator
3	Number
**	Power operator
2	Number



Συντακτικός Αναλυτής

- Αποτελεί το κεντρικό τμήμα ενός μεταγλωττιστή.
- Αναγνωρίζει συντακτικά ορθές “προτάσεις” μίας γραμματικής.
- Δημιουργεί το Αφηρημένο Συντακτικό Δένδρο.
- Εισάγει πληροφορία στον Πίνακα Συμβόλων.
- Διαχείριση μηνυμάτων ελέγχου λαθών.

Τι κάνει το Bison

- Εργαλείο παραγωγής συντακτικού αναλυτή.
- Default LALR(1) γραμματικές.
- Περιέχει σημασιολογικές ρουτίνες.
- Παράγει ένα C αρχείο από ένα .y αρχείο που κατασκευάζουμε εμείς.
- Συνοπτικά χρησιμοποιείται για:
  - Αναγνώριση ή απόρριψη συμβολοσειρών της γλώσσας.
  - Την παραγωγή ενδιάμεσου κώδικα.
  - Την παραγωγή δένδρου συντακτικής ανάλυσης.

Bison

- Μετά-εργαλείο παραγωγή συντακτικών αναλυτών.
- Open-source.
- Linux OS:
  - Download address:
  - http://www.gnu.org/software/bison

- Windows OS:
  - Διατίθεται μαζί με το περιβάλλον του Cygwin
  - http://www.cygwin.com

Δομή Αρχείου Εισόδου

- Το Bison λαμβάνει ως είσοδο ένα μοναδικό αρχείο προγράμματος με κατάληξη “.y” που αποτελείται από 3 μέρη (όπως και το flex).
  - Ορισμοί.
  - Κανόνες.
  - Συναρτήσεις.
- Δομή:

```
/*Statements block*/
%%
/*Rules block*/
%%
/*User Functions*/
```

Τμήμα Ορισμών

- Επιλογές του εργαλείου.
- Δηλώσεις τερματικών συμβόλων.
  - %token.
- Δηλώσεις προτεραιότητας και προσεταιριστικότητας τελεστών.
  - %left, %right, %nonassoc.
- Δήλωση του YYSTYPE.
- Δηλώσεις των τύπων των μη τερματικών συμβόλων.
  - %type<type_val>, η type_val δηλώνετε στο union YYSTYPE.
- Literal block- περικλείεται εντός %{...%} και περιλαμβάνει C κώδικα.

Τμήμα Ορισμών

- Δήλωση τερματικών:
  - Μόνο κωδικοποιημένες λεκτικές μονάδων.
  - Στην περίπτωση που έχουν δηλωθεί οι λεκτικές μονάδες στο flex(με defines) πρέπει να τις μεταφέρουμε στο bison χωρίς των κωδικό τους.
- Προσεταιριστικότητες:
  - %nonassoc: καμία προσεταιριστικότητα.
  - %left: αριστερή προσεταιριστικότητα.
  - %right: δεξιά προσεταιριστικότητα.
  - Η προτεραιότητα καθορίζεται από τη σειρά που δηλώνουμε τις προσεταιριστικότητες με την πρώτη να έχει την μικρότερη σημασία.
- Δήλωση αρχικού συμβόλου:
  - %start symbol_name

Προτεραιότητα και Προσεταιριστικότητα Τελεστών

- Τα %left, %right δηλώνουν τις προσεταιριστικότητες των tokens που τα ακολουθούν. Η σειρά δήλωσής τους καθορίζει την προτεραιότητα των αντίστοιχων τελεστών.
- Ισχύουν τα εξής:
  - τα tokens που εμφανίζονται στην ίδια γραμμή έχουν την ίδια προτεραιότητα.
  - η προτεραιότητα αυξάνεται από πάνω προς τα κάτω.

Παράδειγμα

```
%left ‘+’ ‘-’
%left ‘*’ ‘/’
```

- Τα ‘+’ και ‘-’ έχουν μικρότερη προτεραιότητα από τα ‘*’ και ‘/’.
- Η έκφραση a+b+c υπολογίζεται ως (a+b)+c.
- Το %nonassoc χρησιμοποιείται για τελεστές που δεν μπορούν να συνδυαστούν μεταξύ τους, π.χ., το ‘=‘.

Τμήμα κανόνων

- Περιλαμβάνει τους κανόνες της γραμματικής της γλώσσας σε μορφή BNF.
- Μορφή: rule {C code};
- Παράδειγμα:
  - Date: month SPACE day SPACE year {printf(“date found”);}
- Ο κώδικας των ενεργειών εκτελείται όταν ο αναλυτής φθάσει σε εκείνο το σημείο της γραμματικής.
- Περιέχονται σημασιολογικές ρουτίνες γραμμένες σε C

Τμήμα Συναρτήσεων Χρήστη

- Περιλαμβάνει τις οριζόμενες από τον χρήστη συναρτήσεις.
- Αντιγράφεται αυτούσιο στο παραγόμενο αρχείο.
- Παραδείγματα:
  - main()
  - void yyerror(const char *message)
  - Άλλες συναρτήσεις σημασιολογικής ανάλυσης.

Χρήσιμες Συναρτήσεις και Μεταβλητές

- int yypasre()
  - Υλοποιεί τον συντακτικό αναλυτή.
  - Επιστρέφει 1 εάν υπάρξει συντακτικό σφάλμα και 0 αλλιώς
- void yyerror(cosnt char* message)
  - Πρέπει να υλοποιηθεί υποχρεωτικά.
  - Καλείται αυτόματα σε περίπτωση συντακτικού λάθους.
  - yyerror(“An error occured”);
- YYSTYPE yylval
  - Μεταβλητή για την επικοινωνία λεκτικού-συντακτικού.
  - Περιέχει την σημασιολογική τιμή της λεκτικής μονάδας.
  - Παράδειγμα ορισμού τύπου yylval typedef int YYSTYPE; (στο πρώτο μέρος).
- yyloc: Μεταβλητή που αποθηκεύτεται η θέση κάθε συμβόλου.

yylval

- Στο τμήμα ορισμών ορίζεται σαν union ως εξής

```
%union{
        int intval;
        double doubleval;
        char *strval;
}
```

- Στο flex καλείται πάντα το κατάλληλο πεδίο από το union:
  - yylval.doubleval = hexRealHandle(yytext);
  - yylval.strval = strdup(yytext);
  - …

Υλοποίηση main

- Μεταφορά της main από το flex
- Ανοίγει το αρχείο προς μεταγλώττιση (fopen)
- Δημιουργεί τον πίνακα συμβόλων
- Καλεί την yyparse()
- Καταστρέφει τον πίνακα συμβόλων
- Κλείνει το αρχείο προς μεταγλώττιση

Κανόνες

- Γραμματική Bison → set κανόνων.
- Το LHS έχει ένα μη τερματικό σύμβολο της γραμματικής.
- Το RHS μπορεί να περιέχει τερματικά και μη τερματικά σύμβολα.
- Σε περίπτωση που αντιστοιχούν περισσότεροι από ένας κανόνες για το ίδιο LHS, χρησιμοποιούμε τον τελεστή |.
- Για παραγωγή της κενής συμβολοσειράς ε, αφήνουμε κενό RHS.

Παραδείγματα

- Μπορούν να αντιστοιχούν περισσότερη κανόνες σε κάθε περίπτωση.
  - Expression : expression OPER expression
    | expression → UNOPER expression;
- Παραγωγή κενής συμβολοσειρά:
  - Expession_list: general_expression
    |
    ;

Παραδείγματα ΙΙ

- Γραμματική:

  E -> T | E + T

  T -> F | T * F

  F -> (E) | num

- Υλοποίηση σε bison

```
%token TK_NUM
%left '+‘
%left '*'
%%
program :expression
                ;
expression :term
                | expression ‘+’ term
                ;
term :factor
                | term ‘*’ factor
                ;
factor :‘(‘ expression ‘)’
                | TK_num
                ;
```

Παραδείγματα III

```
line : expr '\n' { printf ("%d\n" , $1 ) ; }
expr : expr '+' term { $$ = $1 + $3 ; }
     | term
     ;
term : term '*' factor { $$ = $1 * $3 ; }
     | factor
     ;
factor : '(' expr ')' { $$ = $2 ; }
       | DIGIT
       ;
```

- Στο δεύτερο μέρος γράφουμε την γραμματική. Κάθε κανόνας μπορεί να έχει και σημασιολογικές πράξεις.
- Η έκφραση $$ αφορά το σύμβολο που είναι στην αριστερή πλευρά ενός κανόνα. Οι εκφράσεις $1, $2, κ.τ.λ. αφορά τα σύμβολα που είναι στην δεξιά πλευρά του κανόνα.

Πίνακας Συμβόλων

- Συγκεντρώνει πληροφορίες για τα ονόματα (identifiers) που εμφανίζονται στο αρχικό πρόγραμμα.
- Ονόματα είναι: Πρόγραμμα, μεταβλητές, σταθερές...
- Η πληροφορία που αποθηκεύεται στον πίνακα συμβόλων χρησιμοποιείται κατά τη σημασιολογική ανάλυση (έλεγχος τύπων) και κατά την παραγωγή κώδικα (πχ. πόσο χώρο στη μνήμη απαιτούν).
  - int x; float y;

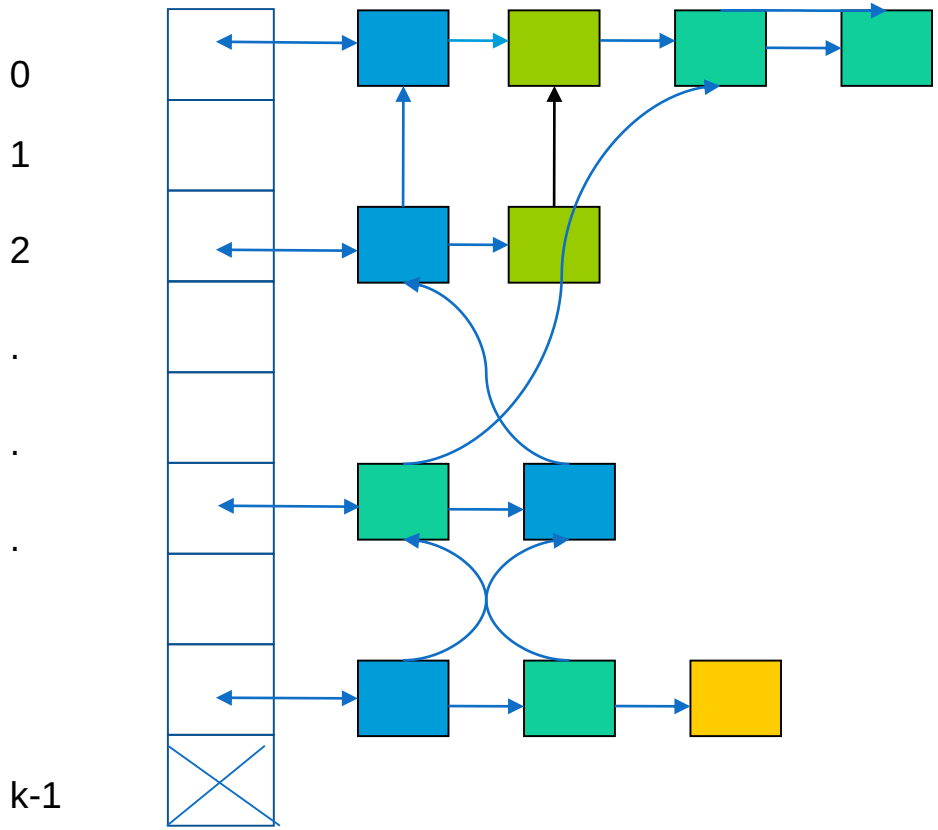
Οργάνωση πίνακα συμβόλων

- Βασικές λειτουργίες:
  - Προσθήκη ονόματος.
  - Αναζήτηση ονόματος.
  - Διαγραφή ονόματος.
- Κόστος προσθήκης ή αναζήτησης ανάλογα με την υλοποίηση:
  - Γραμμική λίστα → O(n)
  - Δυαδικό δέντρο αναζήτησης → O(log n)
  - Πίνακας κατακερματισμού →O(n/k) (η πιο συμφέρουσα υλοποίηση)

Πίνακες Κατακερματισμού

- Η αποδοτικότερη των τριών δομών.
- Βασίζεται στην ύπαρξη δύο πινάκων: του πίνακα αποθήκευσης και του πίνακα κατακερματισμού.
- Πίνακας κατακερματισμού:
  - Αποτελείται από k θέσεις.
  - Κάθε θέση έχει ένα δείκτη σε μια συνδεδεμένη λίστα.
  - Η συνάρτηση κατακερματισμού (hash function) δίνει σε ποια θέση του πίνακα κατακερματισμού αντιστοιχεί ένα όνομα, και οδηγεί στην αντίστοιχη σειριακή (συνδεδεμένη) λίστα.

Σχηματικά



Συνάρτηση Κατακερματισμού

- Σημαντικό ρόλο για την επιτυχία της μεθόδου.
- Επιθυμητή η ομοιόμορφη κατανομή ονομάτων στη λίστα.
- Εύκολος υπολογισμός όταν το όνομα είναι ακολουθίες χαρακτήρων.
  - Σ (name) = Άθροισμα ASCII χαρακτήρων ονόματος
  - hash(name) = Σ mod k

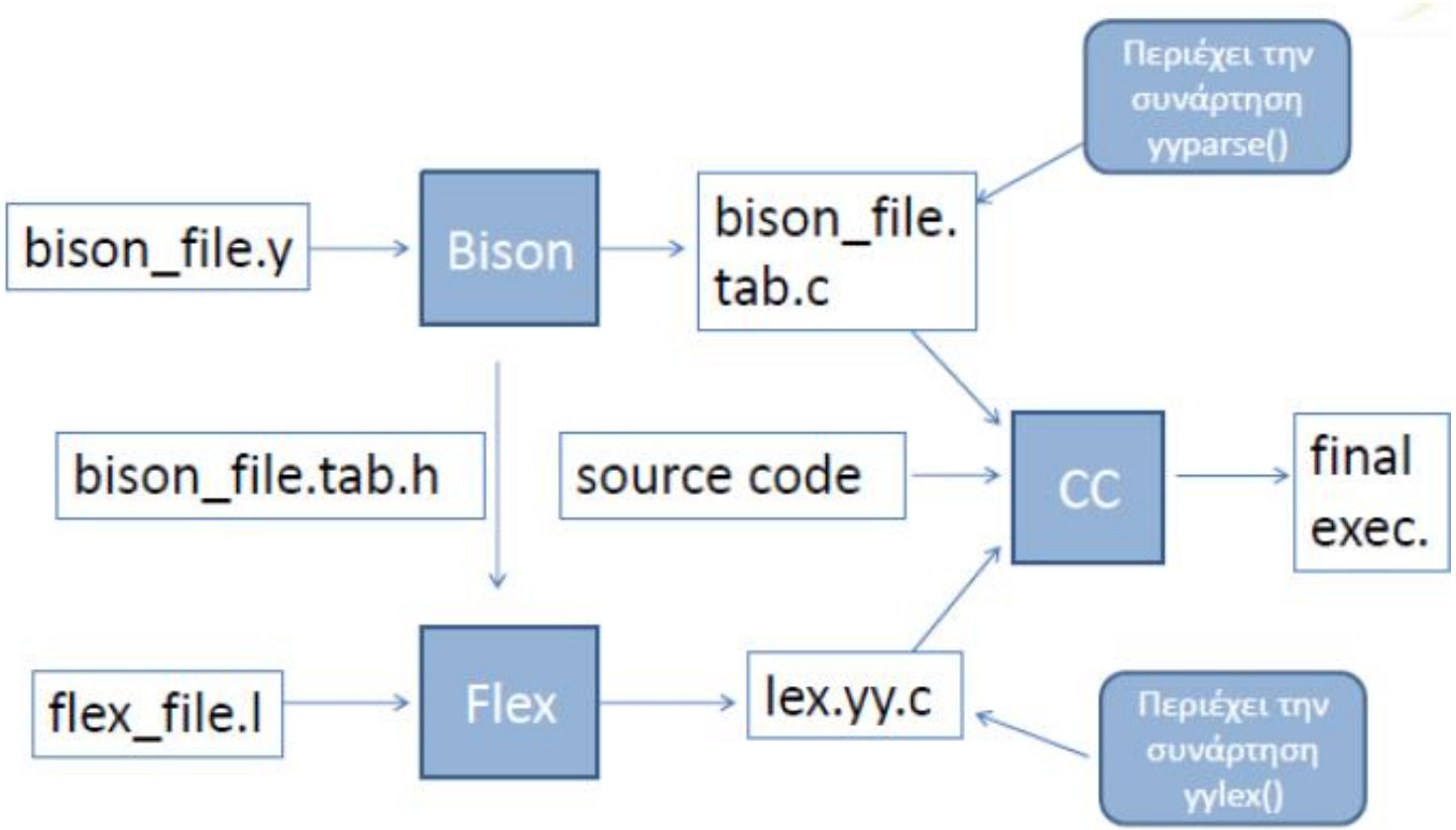
Εμβέλεια

- Ένας τρόπος διαχείρισης της εμβέλειας είναι να αντιστοιχίσουμε σε κάθε εμβέλεια μοναδικό αριθμό εμβέλειας .
- Μια καθολική μεταβλητή μπορεί να δείχνει την τρέχουσα εμβέλεια.
  - η μεταβλητή αυτή θα πρέπει να αυξάνεται / μειώνεται στις ενέργειες (actions) των γραμματικών κανόνων
- Μόλις κλείσει μια εμβέλεια όλες οι καταχωρήσεις με τον αριθμό που αντιστοιχεί στην εμβέλεια εξαλείφονται.

Compilation

- Όνομα αρχείου → file_name.y
- Εντολές μεταγλώττισης → bison –v –d file_name.y
- Παράγονται file_name.tab.c και file_name.tab.h
  - Include στον λεκτικό αναλυτή.
- Compile του παραγόμενου αρχείου
  - gcc lex.yy.c file_name.tab.c –o desired_executable_name –lfl

Compilation Flow



ΕΡΩΤΗΣΕΙΣ/ΑΠΟΡΙΕΣ