Project Report: Flex/Bison 2025

Η εργασία είχε ως στόχο την δημιουργία ενός lexer/parser για την γλώσσα MyHTML.

**Δομή των αρχείων:**

project\_directory

CMakeLists.txt

example.myhtml

grammar.bnf

project\_report.pdf

src (directory)

lexer.l

parser.y

main.c

ckecks.c

checks.h

**Επεξήγηση αρχείων:**

**grammar.bnf:** περιέχει την γραμματική σε BNF μορφή.

**example.myhtml:** αποτελεί το αρχείο το οποίο ο parser παίρνει ως είσοδο.

**CMakeLists.txt:** Για το build system χρησιμοποιήθηκε το cmake, το οποίο ενώνει τα αρχεία flex/bison, το main.c, καθώς και τα checks.c, checks.h ως static library. Με αυτόν τον τρόπο το build του προγράμματος γίνεται κατευθείαν το make, χωρίς περεταίρω εντολές όπως bison …, flex …, gcc …

Οδηγίες για το build του project υπάρχουν παρακάτω.

**lexer.l:** περιέχει τον λεκτικό αναλυτή. Η δομή του είναι σχετικά απλή, καθώς το μόνο που κάνει είναι να μετατρέπει μία λέξη σε ένα token και να τα στέλνει στον parser.



Η παραπάνω εντολή επιστρέφει ένα “START\_HEAD” token κάθε φορά που η λέξη “<head>” εμφανίζεται στο αρχείο.

*Σημείωση: Το macro debug\_token(tok) χρησιμοποιήθηκε για λόγους debugging (όριζε μία συνάρτηση printf()).*

Ιδιαίτερη προσοχή ίσως θέλει το comment token, το οποίο λέει στον lexer να μπει σε κατάσταση comment κάθε φορά που το συναντάει:



Υλοποιήθηκε με αυτόν τον τρόπο ώστε ο κανόνας για το comment\_body να μην επεκτείνεται στους υπόλοιπους κανόνες.

**parser.y:** περιέχει τον συντακτικό αναλυτή. Ο σκοπός του είναι να δέχεται τα tokens από τον λεκτικό αναλυτή και να ελέγχει την ορθότητα της σύνταξης. Οι κανόνες που ορίζονται σε αυτόν μπορούν να περιέχουν κώδικα c για περεταίρω ελέγχους.

Περισσότερα για τον συντακτικό αναλυτή αναφέρονται παρακάτω.

**main.c:** Περιέχει την main() συνάρτηση, η οποία δέχεται ως είσοδο το όνομα του MyHTML αρχείου. Στη συνέχεια ανοίγει το αρχείο, το εκτυπώνει στην οθόνη, επιστρέφει το input στην αρχή του αρχείου και καλεί τον parser. Αφού αναλυθεί το αρχείο, καλείται η συνάρτηση show\_errors() για την εμφάνιση των προβλημάτων που μπορεί να υπήρξαν.

A screenshot of a computer screen

AI-generated content may be incorrect.Επιστροφή του input στην αρχή του αρχείου

**checks.c:** αποτελεί αρχείο πηγαίου κώδικα για την βιβλιοθήκη που υλοποιεί τους ελέγχους για τα ερωτήματα 2 και 3.

Περισσότερες λεπτομέρειες υπάρχουν παρακάτω.

**checks.h:** αποτελεί αρχείο header για την βιβλιοθήκη checks.

**Οδηγίες building:**

Στο directory του project:

1. mkdir build (δημιουργία build directory)
2. cmake .. (εκτέλεση cmake)
3. make (compile το πρόγραμμα)

**Χρήση του προγράμματος:**

Στο directory του project:

1. ./myhtml example.myhtml (εκτέλεση του προγράμματος με το όνομα του αρχείου

που θα αναλυθεί)

**Έξοδος του προγράμματος:**

Το πρόγραμμα έχει ως έξοδο το αρχείο που παίρνει ως είσοδο, καθώς και τυχόν προβλήματα.

Τα προβλήματα μπορεί να είναι συντακτικού τύπου, δηλαδή να προέρχονται από τον parser, ή προβλήματα που περιγράφονται στα ερωτήματα 2 και 3 και υλοποιούνται σε C.

Επιτυχής ανάλυση του προγράμματος

A computer screen shot of a program code

AI-generated content may be incorrect.Αποτυχημένη ανάλυση του προγράμματος με ένδειξη του προβλήματος και της γραμμής στην οποία βρίσκεται

**Περιγραφή του parser:**

Ο parser ξεκινάει με τον ορισμό των tokens τα οποία θα δεχτεί. Μερικά από αυτά έχουν και τύπο δεδομένου, ο οποίος ορίζει τον τύπο που θα επιστρέφει το token.

π.χ.



Ακολουθεί ο ορισμός των κανόνων με την γενική μορφή myhtml, η οποία αποτελείται από τα στοιχεία START\_MYHTML, head\_opt, body και END\_MYHTML. Αυτά επεκτείνονται στη συνέχεια, ορίζοντας την γραμματική της γλώσσας.

Σε αρκετά σημεία των κανόνων υπάρχει κώδικας C με συναρτήσεις από το αρχείο checks, όπως:

check\_id() : έλεγχος αν το id έχει χρησιμοποιηθεί

is\_url() : έλεγχος αν το string είναι url

is\_valid\_href() : έλεγχος του format του χαρακτηριστικού href

is\_valid\_style() : έλεγχος αν η τιμή του χαρακτηριστικού style ακολουθεί τους κανόνες του ερωτήματος 2.g

type\_is\_valid() : έλεγχος του format του χαρακτηριστικού type

Αν κάποιος έλεγχος από τους παραπάνω δεν είναι επιτυχής, ένα error προστίθεται στην στοίβα των error μέσω της συνάρτησης

void add\_error(int line\_number, err\_type\_t err);

Οι παράμετροι της συνάρτησης αποτελούνται από την γραμμή στην οποία συναντήθηκε το error, καθώς και τον τύπο του προβλήματος που υπήρξε.

Η γραμμή στην οποία συναντήθηκε το πρόβλημα μετριέται από τον lexer, ο οποίος αυξάνει την μεταβλητή line\_number κάθε φορά που συναντά τον χαρακτήρα \n (new line).

Για τον ορισμό ενός η παραπάνω στοιχείου χρησιμοποιήθηκε left recursion:

π.χ. A black background with yellow and orange text

AI-generated content may be incorrect.

**Περιγραφή του checks.c:**

Το αρχείο checks.c περιέχει τις συναρτήσεις για την υλοποίηση των ελέγχων των ερωτημάτων 2, 3, καθώς και για τον διαχειρισμό των errors.

Για την αποθήκευση των errors χρησιμοποιήθηκε ένα array από error\_t, το οποίο το πρόγραμμα το διαχειρίζεται σαν στοίβα.

To error\_t είναι τύπος που περιέχει έναν ακέραιο αριθμό για την γραμμή που εμφανίστηκε το error, καθώς και ένα err\_type\_t το οποίο είναι ένα enum με τα πιθανά προβλήματα.

A screen shot of a computer program

AI-generated content may be incorrect.Ορισμός των τύπων.

Για την αποθήκευση των id χρησιμοποιήθηκαν δύο 2D πίνακες από χαρακτήρες. Ο ένας περιέχει όλα τα id του προγράμματος και ο άλλος τα id που έχουν χρησιμοποιηθεί. Ο διαχωρισμός αυτός έγινε ώστε να εξασφαλιστεί ένα στοιχείο label με ένα μοναδικό στοιχείο input μέσω του id.

Οι δομές για την αποθήκευση των παραπάνω στοιχείων, καθώς και οι μεταβλητές για την καταμέτρηση των id, των error κλπ. ορίζονται ως static ώστε να είναι προσβάσιμες μόνο μέσα στο αρχείο checks.c. Με αυτόν τον τρόπο, το check.c αρχείο παρέχει στον parser μόνο ένα σύνολο συναρτήσεων για την υλοποίηση των ελέγχων.