Βελτιώσεις και εξηγήσεις πάνω στον κώδικα Τμήμα Β3

Διονύσης Νικολόπουλος

6 Ιουνίου 2021

Ομάδα 15

Αναλυτικά τα μέλη:

Διονύσης Νικολόπουλος AM : 18390126

Θανάσης Αναγνωστόπουλος AM : 18390043

Αριστείδης Αναγνωστόπουλος AM : 16124

Σπυρίδων Φλώρος AM : 141084

Αναλυτικά οι ρόλοι:

Γενικός Συντονιστής: Διονύσης Νικολόπουλος

Υπεύθυνος Τμήματος Εργασίας Β3: Διονύσης Νικολόπουλος

Usernames στο Github

Διονύσης Νικολόπουλος : dnnis

Θανάσης Αναγνωστόπουλος : ThanasisAnagno

Αριστείδης Αναγνωστόπουλος : Aris-Anag

Σπυρίδων Φλώρος : spirosfl

Η εργασία αυτή πραγματοποιήθηκε με χρήση LaTeX

1 Εισαγωγή

Σε αυτό το έγγραφο θα σας ενημερώσω για τις αλλαγές πάνω στον κώδικά μας μέχρι το μέρος B3.
Θα εξηγηθεί ο κώδικας τμηματικά, θα δωθούν πρόσθετες εξηγήσεις, μαζί με αυτές που δίνονται απο τα σχόλια στον κώδικα τον ίδιο.

1.1 Σημειώσεις για τους φακέλους

- **bison-part**: Σε αυτόν τον κατάλογο βρίσκονται τα αρχεία πηγαίου κώδικα του bison.
- **flex-part**: Σε αυτόν τον κατάλογο βρίσκονται τα αρχεία πηγαίου κώδικα του flex.
- **compile-room**: Σε αυτόν τον κατάλογο βρίσκεται το makefile, το οποίο αντιγράφει τα απαραίτητα αρχεία από τους προαναφερόμενους δύο καταλόγους στον τρέχοντα (compile-room), και με αυτά τα αρχεία κάνει compile στο τελικό μας πρόγραμμα, που ονομάζεται uni-c-analyser.

2 Τμηματικά ο κώδικας - Εξηγήσεις

2.1 FLEX

2.1.1 Includes και μεταβλητές

```
/* Kwdikas C gia orismo twn apaitoumenwn header files kai twn
   metablhtwn.
   Otidhpote anamesa sta %{ kai %} metaferetai autousio sto arxeio C
   pou
   tha dhmiourghsei to Flex. */

%{

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "bison-SA.tab.h"

// Για να μετράμε τις κολόνες
int columns = 1;
// Αρχικοποιούμε τις μεταβλητές για το άθροισμα των σωστών και λάθος
   λέξεων
int cor_words = 0;
int inc_words = 0;
// Για να καταφέρνουμε να δίνουμε στον χρήστη σωστό output.
char panic_cause_char[100];

%}
```

Εδώ βλέπουμε τα include μας, που είναι απαραίτητα τόσο για την λειτουργεία του προγράμματος (πχ. '#include <stdlib>' κτλ) όσο και για την σύνδεση του flex με του bison (πχ. '#include "bison-SA.tab.h"')
Επίσης, βλέπουμε τις μεταβλητές που ορίζουμε για την ομαλή διεπαφή του χρήστη με το πρόγραμμα.
Πλέον το πρόγραμμά μας μετρά κολόνες και μπορεί να κατευθύνει τον χρήστη στο ακριβές σημείο που το πρόβλημα δημιουργήθηκε (με κάποιο άγνωστο token).
Επίσης, σε σύγκριση με το μέρος B2, τώρα το πρόγραμμα μετρά σωστά

τις σωστές και λάθος εκφράσεις.
Τις λάθος λέξεις τις μετρά ο λεκτικός αναλυτής, ο FLEX. Τις λάθος εκφράσεις ο συντατικός αναλυτής, το BISON.
Επίσης, στην μεταβλητή panic_cause_char αποθηκεύουμε την άγνωστη λέξη, και την αναφέρουμε στον χρήστη μετέπειτα.

2.1.2 TOKENS, Κανονικές Εκφράσεις, Καταστάσεις

```
/* Onomata kai antistoixoi orismoi (ypo morfh kanonikhs ekfrashs).
   Meta apo auto, mporei na ginei xrhsh twn onomatwn (aristera) anti
    twn,
   synhthws idiaiterws makroskelwn kai dysnohtwn, kanonikwn ekfrasewn
    */

SEMI                ;
HASH                #
TILDE               ~
NEQ                 !=
MOD                 \%
POW                 \^
DOT                 \.
COMMA               \,
COLON               \:
AMPER               \&
PAR_END             \)
PAR_START           \(
BRACE_END           \}
BRACE_START         \{
BRACKET_END         \]
BRACKET_START       \[
LOGICAL_OR          \|\|
TYPE_EQ             ==?
TYPE_DIV            \/=?
TYPE_MULTI          \*=?
TYPE_EXCLA          \!=?
TYPE_AMPER          \&\&
TYPE_LESSER         \<=?
TYPE_GREATER        \>=?
WHITESPACE          [ \t]
TYPE_PLUS           \+[\+=]?
TYPE_MINUS          \-[\-=]?
STRING              '.*'|\".*\"
```

```
33 INTCONST             0|[1-9]+[0-9]*
34 COMMENT              \/\*(.|\n)*?\*\/|\/\/.*
35 IDENTIFIER           [a-zA-Z_]([0-9_a-zA-Z]*)
36 FLOAT                [0-9]+\.[0-9]+|[0-9]+\.[0-9]+e[0-9]+
37 %x REALLYEND
38 %x PREPANIC
39 %x PANIC
```

Τώρα βλέπουμε τις κανονικές εκφράσεις με τις οποίες ο λεκτικός αναλυτής μας ανιχνεύει τις λέξεις του αναλυόμενου κώδικα.
Στο τέλος έχουμε και τους ορισμούς για τις 3 καταστάσεις τις οποίες ορίσαμε για να λειτουργεί ορθά ο λεκτικός αναλυτής.
Αυτές είναι:

- **Κατάσταση REALLYEND** είναι η κατάσταση στην οποία ο λεκτικός αναλυτής έχει ήδη απο τον συντακτικό αναλυτής το μύνημα ότι ο πρώτος έχει λάβει τις τελευταίες δράσεις πριν τον τερματισμό του προγράμματος, και αρχίζει να κλείνει "πραγματικά" το πρόγραμμα. (Πριν την κατάσταση αυτή στέλνει μήνυμα EOF στον BISON, που παίρνει δράσεις που θα αναλύσουμε παρακάτω. Μετά από τις δράσεις αυτές, ο FLEX μπαίνει στην κατάσταση REALLYEND)

- **Κατάσταση PANIC** είναι η κατάσταση στην οποία ο λεκτικός αναλυτής έχει συναντίσει ένα άγνωστο token (UNKNOWN TOKEN) και προσπαθεί να κάνει επανάκτιση κανονικής λειτουργίας.

- **Κατάσταση PREPANIC**