

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών: Εργαστηριακή Άσκηση 2014-2015

22 Μαρτίου 2015

Περίληψη

Σκοπός της παρούσας εργασίας είναι η εξοικείωσή σας με τις θεμελιώδεις θεωρητικές και πρακτικές πτυχές της υλοποίησης λεκτικών/συντακτικών αναλυτών με τη βοήθεια των μεταεργαλείων flex και bison.

Ερώτημα 1

(1 μονάδα) Δίνεται η παρακάτω γραμματική σε συμβολισμό EBNF που περιγράφει, χωρίς πολλές λεπτομέρειες, τους ορισμούς συναρτήσεων της C και ένα μικρό υποσύνολο των εντολών της. (Παραλείπονται οι ορισμοί των μη τερματικών συμβόλων *simple*, *formals* και *expr*.)

```
<decl> ::= <result> <id> ( <formals> ) { <stmt>* }  
<result> ::= <type> | void  
<stmt> ::= <simple> | break ; | continue ; | return [ <expr> ] ;  
          | while ( <expr> ){ <stmt>* }  
          | switch ( <expr> ) { <stmt>* }
```

Στη γραμματική αυτή, λείπουν οι εξής περιορισμοί που επιβάλλει η γλώσσα C:

- Στο εσωτερικό μιας συνάρτησης που επιστρέφει **void** δεν μπορεί να εμφανίζεται η εντολή **return** συνοδευόμενη από έκφραση.
- Στο εσωτερικό μιας συνάρτησης που δεν επιστρέφει **void**, αν εμφανίζεται η εντολή **return** θα πρέπει να συνοδεύεται από έκφραση.
- Η εντολή **break** μπορεί να εμφανίζεται μόνο στο εσωτερικό μιας εντολής **while** ή μιας εντολής **switch**.
- Η εντολή **continue** μπορεί να εμφανίζεται μόνο στο εσωτερικό μιας εντολής **while**.

Να τροποποιήσετε κατάλληλα την παραπάνω γραμματική ώστε οι παραγόμενες συμβολοσειρές να πληρούν αυτούς τους περιορισμούς.

Ερώτημα 2 : Υλοποίηση Parser της γλώσσας Golub

Θεωρείστε την γλώσσα **Golub**, το EBNF της οποίας δίνεται παρακάτω:

```

<πρόγραμμα> ::= <εξωτερικές-δηλώσεις><κεφαλίδα-προγράμματος><τμήμα-ορισμών><τμήμα-εντολών>
<εξωτερικές-δηλώσεις> ::= (<εξωτερικό-πρωτότυπο>)*
<εξωτερικό-πρωτότυπο> ::= extern<πρωτότυπο-συνάρτησης>
<κεφαλίδα-προγράμματος> ::= void<id>{}
<τμήμα-ορισμών> ::= (<ορισμός>)*
<ορισμός> ::= <ορισμός-μεταβλητών>|<ορισμός-συνάρτησης>|<πρωτότυπο-συνάρτησης>
<ορισμός-μεταβλητών> ::= <τύπος-δεδομένων><λίστα-μεταβλητών>;
<τύπος-δεδομένων> ::= integer| boolean| string
<λίστα-μεταβλητών> ::= <id>(<id>)*
<ορισμός-συνάρτησης> ::= <κεφαλίδα-συνάρτησης><τμήμα-ορισμών><τμήμα-εντολών>
<πρωτότυπο-συνάρτησης> ::= <κεφαλίδα-συνάρτησης>;
<κεφαλίδα-συνάρτησης> ::= <τύπος-συνάρτησης><id>([<λίστα-τυπικών-παραμέτρων>])
<τύπος-συνάρτησης> ::= integer| boolean| void
<λίστα-τυπικών-παραμέτρων> ::= <τυπικές-παραμέτροι>(<,><τυπικές-παραμέτροι>)*
<τυπικές-παραμέτροι> ::= <τύπος-δεδομένων>[&]<id>
<τμήμα-εντολών> ::= begin(<εντολή>)* end
<εντολή> ::= <απλή-εντολή>;|<δομημένη-εντολή>|<σύνθετη-εντολή>
<σύνθετη-εντολή> ::= {(<εντολή>)*}
<δομημένη-εντολή> ::= <εντολή-if>
<απλή-εντολή> ::= <ανάθεση>|<κλήση-συνάρτησης>|<εντολή-return>|<εντολή-null>
<εντολή-if> ::= if(<γενική-έκφραση>)<εντολή>[<else-clause>]
<else-clause> ::= else<εντολή>
<ανάθεση> ::= <id>=<γενική-έκφραση>
<κλήση-συνάρτησης> ::= <id>([<λίστα-πραγματικών-παραμέτρων>])
<λίστα-πραγματικών-παραμέτρων> ::= <πραγματική-παραμέτρος>(<,><πραγματική-παραμέτρος>)*
<πραγματική-παραμέτρος> ::= <γενική-έκφραση>
<εντολή-return> ::= return[<γενική-έκφραση>]
<εντολή-null> ::= ε
<γενική-έκφραση> ::= <γενικός-όρος>(|<γενικός-όρος>)*
<γενικός-όρος> ::= <γενικός-παράγοντας>(&&<γενικός-παράγοντας>)*
<γενικός-παράγοντας> ::= [!]<γενικός-πρωτ-παραγ>
<γενικός-πρωτ-παραγ> ::= <απλή-έκφραση>[<τμήμα-σύγκρισης>]
<τμήμα-σύγκρισης> ::= <συγκριτικός-τελεστής><απλή-έκφραση>
<συγκριτικός-τελεστής> ::= ==|!=|<|>|<=|>=
<απλή-έκφραση> ::= <απλός-όρος>((+|-)<απλός-όρος>)*
<απλός-όρος> ::= <απλός-παράγοντας>((/*|/|%)<απλός-παράγοντας>)*
<απλός-παράγοντας> ::= [+|-]<απλός-πρωτ-όρος>
<απλός-πρωτ-όρος> ::= <id>|<σταθερά>|<κλήση-συνάρτησης>[(<γενική-έκφραση>)]
<σταθερά> ::= <σταθερά-ακεραίου>|<σταθερά-συμβολοσειρά>|true|false

```

1. **(6 μονάδες)** Χρησιμοποιώντας τα μεταεργαλεία Flex και Bison, υλοποιήστε έναν λεκτικό και συντακτικό αναλυτή, ο οποίος θα παίρνει ως είσοδο ένα αρχείο γραμμένο στη γλώσσα **Golub** που περιγράφηκε πιο πάνω και θα ελέγχει σε ένα πέρασμα αν το πρόγραμμα είναι συντακτικά ορθό. Το πρόγραμμά σας θα καλείται από τη γραμμή εντολών ως εξής:

```
prompt> myParser.exe file.txt
```

και θα δίνει ως έξοδο το αρχείο εισόδου και στη συνέχεια κατάλληλο διαγνωστικό μήνυμα για το αν ήταν ορθώς γραμμένο, ή κατάλληλο μήνυμα σφάλματος (πρέπει να

φαίνεται η γραμμή όπου υπάρχει το σφάλμα). **(Bonus)** Τροποποιήστε τον κώδικα του συντακτικού αναλυτή σας ώστε να μπορεί να κάνει μία εκτίμηση του πλήθους των λαθών που υπάρχουν στο πρόγραμμα.

2. **(3 μονάδες)** *Loop Handling*: Η παραπάνω γραμματική δεν υποστηρίζει τις εντολές **for** και **while**. Προσθέστε τις στη γραμματική και υλοποιήστε τις ακόλουθες loop handling δυνατότητες:

- Στην περίπτωση που ο αριθμός των επαναλήψεων που προβλέπεται από το for-loop είναι μεγαλύτερος από 3, τότε αυτό να αντικαθίσταται από το ισοδύναμο while-loop.
- Διαφορετικά, να πραγματοποιείται *loop-unrolling*: δηλαδή στο τελικό πρόγραμμα εξόδου να γράφεται ο κώδικας του for-block κατάλληλα τροποποιημένος (πχ αντικατάσταση τυχόν δεικτών με τις τιμές τους...) και τον σωστό αριθμό φορών.

Στο αρχείο εξόδου του παρόντος ερωτήματος θα πρέπει, πριν την εμφάνιση μηνύματος συντακτικής εγκυρότητας ή αναφοράς τυχόν σφαλμάτων, να παρατίθεται **το τροποποιημένο αρχείο εισόδου σύμφωνα με το loop handling**.

Παρατηρήσεις - Διαδικαστικά

Για τη χρήση των εργαλείων Flex και Bison μπορείτε να βρείτε πληροφορίες στη σελίδα του μαθήματος. Για την άσκηση μπορείτε να δουλέψετε σε ομάδες έως 4 ατόμων. Η βαθμολογία της άσκησης προκύπτει μετά από **ατομική προφορική εξέταση** που αφορά τόσο τις λεπτομέρειες της υλοποίησης όσο και την ύλη που καλύπτεται από το θεωρητικό τμήμα της άσκησης. Ως ημερομηνία παράδοσης της άσκησης ορίζεται η ημερομηνία γραπτής εξέτασης περιόδου Ιουνίου και Σεπτεμβρίου αντίστοιχα.

Παραδοτέα

- Γραπτή Αναφορά σε **pdf** που περιλαμβάνει:
 - Τις αναλυτικές λύσεις του θεωρητικού τμήματος μαζί με τις απαραίτητες επεξηγήσεις και τεκμηριώσεις όπου αυτό είναι απαραίτητο.
 - Τα αρχεία περιγραφής της γλώσσας, τα οποία δίνονται ως είσοδος στα μεταεργαλεία Flex και Bison.
 - Screenshots παραδειγμάτων εφαρμογής του parser.
- Ένα αρχείο **zip, rar, tar.gz** που περιλαμβάνει:
 - Την αναφορά σε ηλεκτρονική μορφή
 - Όλα τα αρχεία που αφορούν την υλοποίηση (συμπεριλαμβανομένων των αρχείων που δόθηκαν σαν είσοδο στον parser για να ελεγχθεί η σωστή λειτουργία του).

Το αρχείο zip (ή tar.gz) πρέπει να έχει όνομα τους αριθμούς μητρώου των ατόμων της ομάδας διαχωρισμένους με το χαρακτήρα “_”, και διατεταγμένους από τον μικρότερο στο μεγαλύτερο (π.χ. 4500_5543_5788_5972.zip), και να σταλεί (ΥΠΟΧΡΕΩΤΙΚΑ) με email με θέμα **“ASKISI ARXES GLWSSWN 2015”**. Στο σώμα του email θα πρέπει να αναφέρονται τα **ονοματεπώνυμα**, το **έτος** και οι αντίστοιχοι **αριθμοί μητρώου** των μελών της ομάδας.

Για τυχόν απορίες σχετικά με την άσκηση μπορείτε να χρησιμοποιείτε το forum του μαθήματος.