Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών: Εργαστηριακή Άσκηση 2016-2017

Διδάσκοντες: Καθ. Γιάννης Γαροφαλάκης,
Δρ. Αθανάσιος Ν. Νικολακόπουλος
Υποστήριξη Μαθήματος: Θεοφάνης Μιχαήλ, Υπ.Διδάκτορας

27 Μαρτίου 2017

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.

Υλοποίηση Parser της γλώσσας Edelman

Η γλώσσα Edelman είναι μία απλή γλώσσα προστακτικού προγραμματισμού η οποία βασίζεται σε ένα γνήσιο υποσύνολο της γλώσσας C, με κάποιες μικρές παραλλαγές. Ως προς τους κανόνες των αναγνωριστικών η γλώσσα ακολουθεί πιστά τους κανόνες που ισχύουν για τη γλώσσα C, ενώ επίσης υποστηρίζει την ύπαρξη σχολίων σε οποιοδήποτε σημείο του πηγαίου κώδικα τα οποία μπορούν να ακολουθούν είτε τη σύμβαση της C (/* σχόλιο */), είτε της C++ (// σχόλιο). Η γλώσσα Edelman θα πρέπει να υποστηρίζει και τις δύο μορφές.

Το συντακτικό της γλώσσας **Edelman** δίνεται παρακάτω σε μορφή EBNF. Η γραμματική που ακολουθεί είναι διφορούμενη, ωστόσο οι αμφισημίες μπορούν να ξεπεραστούν αν λάβει υπόψη τους κανόνες προτεραιότητας τελεστών της γλώσσας C, τους οποίους κληρονομεί και η γλώσσα Edelman.

⟨*πρόγραμμα*⟩ ::= (⟨*δήλωση*⟩)$^+$
⟨*δήλωση*⟩ ::= ⟨*δήλωση-μεταβλητής*⟩|⟨*δήλωση-συνάρτησης*⟩|⟨*ορισμός-συνάρτησης*⟩
⟨*δήλωση-μεταβλητής*⟩ ::= ⟨*τύπος-δεδομένων*⟩⟨*δηλωτικό*⟩(**,**⟨*δηλωτικό*⟩)***;**
⟨*τύπος-δεδομένων*⟩ ::= ⟨*βασικός-τύπος-δεδομένων*⟩(*****)*
⟨*βασικός-τύπος-δεδομένων*⟩ ::= **int**|**char**|**bool**|**double**
⟨*δηλωτικό*⟩ ::= ⟨*id*⟩[**[**⟨*σταθερή-έκφραση*⟩**]**]
⟨*δήλωση-συνάρτησης*⟩ ::= ⟨*τύπος-επιστρεφόμενης-τιμής*⟩⟨*id*⟩**(**[⟨*λίστα-παραμέτρων*⟩]**);**
⟨*τύπος-αποτελέσματος*⟩ ::= ⟨*τύπος-δεδομένων*⟩|**void**
⟨*λίστα-παραμέτρων*⟩ ::= ⟨*παράμετρος*⟩(**,**⟨*παράμετρος*⟩)*
⟨*παράμετρος*⟩ ::= [**byref**]⟨*τύπος-δεδομένων*⟩⟨*id*⟩
⟨*ορισμός-συνάρτησης*⟩ ::= ⟨*τύπος-αποτελέσματος*⟩⟨*id*⟩**(**[⟨*λίστα-παραμέτρων*⟩]**){**(⟨*δήλωση*⟩)*(⟨*πρόταση*⟩)***}**
⟨*πρόταση*⟩ ::= **;**|⟨*έκφραση*⟩**;**|**{**(⟨*πρόταση*⟩)***}**|**if(**⟨*έκφραση*⟩**)**⟨*πρόταση*⟩[**else**⟨*πρόταση*⟩]
|[⟨*id*⟩**:**]**for(**[⟨*έκφραση*⟩]**;**[⟨*έκφραση*⟩]**;**[⟨*έκφραση*⟩]**)**⟨*πρόταση*⟩
|**continue**[⟨*id*⟩]**;**|**break**[⟨*id*⟩]**;**|**return**[⟨*έκφραση*⟩]**;**

<*έκφραση*> ::= <*id*>|**(**<*έκφραση*>**)**|**true**|**false**|**NULL**|<*int-const*>|<*char-const*>|<*double-const*>
|<*string-literal*>|[<*id*>]**(**[<*λίστα-εκφράσεων*>]**)**|<*έκφραση*>**[**<*έκφραση*>**]**
|<*μοναδιαίος-τελεστής*><*έκφραση*>|<*έκφραση*><*δυαδικός-τελεστής*><*έκφραση*>
|<*μοναδιαίος-τελεστής-ανάθεσης*><*έκφραση*>|<*έκφραση*><*μοναδιαίος-τελεστής-ανάθεσης*>
|<*έκφραση*><*δυαδικός-τελεστής-ανάθεσης*><*έκφραση*>|**(**<*τύπος-δεδομένων*>**)**<*έκφραση*>
|<*έκφραση*>**?**<*έκφραση*>**:**<*έκφραση*>|**new**<*τύπος-δεδομένων*>[**[**<*έκφραση*>**]**]|**delete**<*έκφραση*>

<*λίστα-εκφράσεων*> ::= <*έκφραση*>(**,**<*έκφραση*>)*

<*σταθερά-έκφραση*> ::= <*έκφραση*>

<*μοναδιαίος-τελεστής*> ::= **&**|*|**+**|**-**|**!**

<*δυαδικός-τελεστής*> ::= *|**/**|**%**|**+**|**-**|**<**|**>**|**<=**|**>=**|**==**|**!=**|**&&**|**||**|**,**

<*μοναδιαίος-τελεστής-ανάθεσης*> ::= **++**|**- -**

<*δυαδικός-τελεστής-ανάθεσης*> ::= **=**|***=**|**/=**|**%=**|**+=**|**-=**

1. (**70%**) Χρησιμοποιώντας τα μεταεργαλεία Flex και Bison, υλοποιήστε έναν λεκτικό και συντακτικό αναλυτή, ο οποίος θα παίρνει ως είσοδο ένα αρχείο γραμμένο στη γλώσσα **Edelman** 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). Τέλος ο αναλυτής σας θα πρέπει να μπορεί να κάνει μία εκτίμηση του πλήθους των λαθών που υπάρχουν στο πρόγραμμα.

2. (**20%**) Τροποποιήστε τον αναλυτή σας ώστε να ελέγχει πως πριν χρησιμοποιηθεί μία μεταβλητή, έχει προηγουμένως δηλωθεί από τον χρήστη. Το ίδιο θα πρέπει να ισχύει και για συναρτήσεις που έχει ορίσει ο χρήστης. (Συμβουλή: Η ενασχόληση με το ερώτημα 2 να γίνει μόνο σε περίπτωση επιτυχούς ολοκλήρωσης του προηγούμενου ερωτήματος).

3. (**10%**) Τροποποιήστε τον κώδικά σας ώστε να επαυξάνει τον αναλυτή του ερωτήματος 2 παρέχοντας δυνατότητες ενός – έστω πρώιμου – έλεγχου τύπων.

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 2 ή 3 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. Δικαίωμα συμμετοχής στην προφορική εξέταση έχουν μόνο όσοι έχουν περάσει το μάθημα. Η άσκηση κρατιέται για 2 χρόνια (τρέχον και επόμενο ακαδημαϊκό έτος).

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. 5788_5972.zip), και να σταλεί (ΥΠΟΧΡΕΩΤΙΚΑ) με email στο `michail@ceid.upatras.gr` με θέμα "**ASKISI ARXES GLWSSWN 2017**". 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.