

# Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών: Εργαστηριακή Άσκηση 2011-2012

Γιάννης Γαροφαλάκης, Καθηγητής  
Αθανάσιος Ν. Νικολακόπουλος, Υποψήφιος Διδάκτορας

28 Μαρτίου 2012

## Περίληψη

Σκοπός της παρούσας εργασίας είναι η εξοικείωσή σας με τις θεμελιώδεις θεωρητικές και πρακτικές πτυχές της λεκτικής και συντακτικής ανάλυσης. Η εργασία αποτελείται από ένα θεωρητικό μέρος που αφορά τα βασικά θεωρητικά εργαλεία που πρέπει κανείς να γνωρίζει πριν καταπιαστεί με την υλοποίηση ενός λεκτικού/συντακτικού αναλυτή και έπειτα σας ζητά να φτιάξετε με την βοήθεια των μεταεργαλείων *flex* και *bison* έναν parser για μία απλή γλώσσα προγραμματισμού.

## Θεωρητικό Τμήμα

### Ζήτημα 1 *Αυτόματα και Κανονικές Γλώσσες*

1. Να σχεδιαστεί ένα ντετερμινιστικό πεπερασμένο αυτόματο και μία κανονική έκφραση που να αναγνωρίζουν τη γλώσσα  $L$  με αλφάβητο  $\Sigma = \{0, 1\}$ , η οποία αποτελείται από συμβολοσειρές που περιέχουν υποχρεωτικά μία και μόνο μία φορά την υποσυμβολοσειρά 000. Για παράδειγμα, οι συμβολοσειρές: 10001011, 010110001, 000 ανήκουν στην  $L$ , ενώ οι συμβολοσειρές 1000011, 111, 10100 όχι.
2. Να σχεδιαστεί ένα ντετερμινιστικό πεπερασμένο αυτόματο, που να αναγνωρίζει τη γλώσσα  $L$  με αλφάβητο  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  που αποτελείται από την δεκαδική αναπαράσταση όλων των φυσικών αριθμών που διαιρούνται ακριβώς με το 3. Υπενθυμίζεται ότι ένας φυσικός αριθμός διαιρείται ακριβώς με το 3, αν και μόνο αν το άθροισμα των ψηφίων του (στο δεκαδικό σύστημα αρίθμησης) διαιρείται ακριβώς με το 3. (Το αυτόματο δεν πρέπει να αναγνωρίζει την κενή συμβολοσειρά. Επίσης, για απλοποίηση επιτρέπεται η ύπαρξη μηδενικών στην αρχή ενός φυσικού αριθμού.)
3. Έστω ότι  $\Sigma = \{a, b\}$ . Να βρεθεί μία κανονική έκφραση που να παράγει τη γλώσσα  $L$  η οποία αποτελείται από όλες τις λέξεις  $w$  όπου αρχίζουν και τελειώνουν με  $b$ , και δεν περιέχουν ποτέ την υπολέξη  $bab$ .

4. **\*\*\***(bonus) Σχεδιάστε ένα ντετερμινιστικό πεπερασμένο αυτόματο που αποδέχεται τις συμβολοσειρές δυαδικών αριθμών  $w$  που έχουν την εξής ιδιότητα: αν τις διαβάσεις από το τέλος προς την αρχή παίρνεις ένα δυαδικό ακέραιο που διαιρείται ακριβώς με το 5. Για παράδειγμα, το αυτόματο αποδέχεται την συμβολοσειρά 10011, επειδή όταν την αντιστρέψεις παίρνεις  $(11001)_2 = (25)_{10}$ , που είναι πολλαπλάσιο του 5.

.....

## Ζήτημα 2 Γραμματικές

1. 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.

2. Θεωρήστε την παρακάτω γραμματική:

```

<S> ::= [ <S> <X> ] | a
<X> ::= ε | + <S> <Y> | <Y> b
<Y> ::= ε | - <S> <X> c

```

- (α') Να υπολογίσετε τα σύνολα FIRST, FOLLOW και PREDICT για την παραπάνω γραμματική. Είναι η γλώσσα LL(1)? Δικαιολογήστε την απάντησή σας.

- (β') Να φτιάξετε τον πίνακα συντακτικής ανάλυσης για την παραπάνω γλώσσα (όπως στην εικόνα 2.19 του βιβλίου του Scott). Να περιγράψετε την εκτέλεση της καθοδικής συντακτικής ανάλυσης για μία είσοδο της επιλογής σας με τουλάχιστον 7 χαρακτήρες (βλέπε εικόνα 2.20 του βασικού συγγράμματος). Θεωρείστε, τέλος, έναν συντακτικό αναλυτή αναδρομικής κατάβασης που εκτελείται με την ίδια είσοδο. Να περιγράψετε συνοπτικά τα περιεχόμενα της στοίβας εκτέλεσης του αναλυτή κατά τη διάρκεια αναγνώρισης της εισόδου.
3. Τι είναι η σύγκρουση ολίσθησης-ελάττωσης (shift-reduce conflict); Πώς επιλύεται στα διάφορα είδη συντακτικών αναλυτών της οικογένειας LR; Δείξτε πως η παρακάτω γραμματική είναι LR(1) αλλά όχι SLR(1):

$$\begin{aligned} \langle S \rangle &::= a \langle A \rangle \mid b \langle A \rangle c \mid d c \mid b d a \\ \langle A \rangle &::= d \end{aligned}$$

.....

## Υλοποίηση: Parser της γλώσσας Buzen

Να φτιάξετε έναν λεκτικό και συντακτικό αναλυτή για την παρακάτω γλώσσα της οποίας η περιγραφή σε σημειογραφία EBNF δίνεται παρακάτω:

$$\begin{aligned} \langle \text{program} \rangle &::= \text{program } \langle \text{id} \rangle ; \langle \text{block} \rangle . \\ \langle \text{block} \rangle &::= ( \langle \text{local-definition} \rangle )^* \langle \text{compound-statement} \rangle \\ \langle \text{local-definition} \rangle &::= \langle \text{variable-definition} \rangle \\ &\quad \mid \langle \text{procedure-definition} \rangle \\ &\quad \mid \langle \text{function-definition} \rangle \\ \langle \text{variable-definition} \rangle &::= \text{var } ( \langle \text{def-some-variables} \rangle ; )^+ \\ \langle \text{def-some-variables} \rangle &::= \langle \text{id} \rangle ( , \langle \text{id} \rangle )^* : \langle \text{data-type} \rangle \\ \langle \text{procedure-definition} \rangle &::= \langle \text{procedure-header} \rangle \langle \text{block} \rangle ; \\ \langle \text{procedure-header} \rangle &::= \text{procedure } \langle \text{id} \rangle \langle \text{formal-parameters} \rangle ; \\ \langle \text{function-definition} \rangle &::= \langle \text{function-header} \rangle \langle \text{block} \rangle ; \\ \langle \text{function-header} \rangle &::= \text{function } \langle \text{id} \rangle \langle \text{formal-parameters} \rangle : \langle \text{data-type} \rangle ; \\ \langle \text{formal-parameters} \rangle &::= [ ( \langle \text{formal-parameter} \rangle ( ; \langle \text{formal-parameter} \rangle )^* ) ] \\ \langle \text{formal-parameter} \rangle &::= \langle \text{id} \rangle ( , \langle \text{id} \rangle )^* : \langle \text{data-type} \rangle \\ \langle \text{data-type} \rangle &::= \text{integer} \mid \text{boolean} \end{aligned}$$

```

<statement> ::= |
               | <assignment>
               | <if-statement>
               | <while-statement>
               | <proc-func-call>
               | <compound-statement>
<assignment> ::= <id> := <expression>
<if-statement> ::= if <expression> then <statement> [ else <statement> ]
<while-statement> ::= while <expression> do <statement>
<proc-func-call> ::= <id> [ ( <actual-parameters> ) ]
<actual-parameters> ::= <expression> ( , <expression> ) *
<compound-statement> ::= begin [ <statement> ( ; <statement> ) * ] end
<expression> ::= <expression> <binary-operator> <expression>
               | <unary-operator> <expression>
               | <proc-func-call>
               | ( <expression> )
               | <integer-constant>
               | <id>
<binary-operator> ::= = | < > | or | and
                  | < | > | >= | <=
                  | + | - | * | div | mod
<unary-operator> ::= + | - | not

```

Για την παραπάνω γλώσσα η έγκυρη μορφή των αναγνωριστικών καθώς και η προτεραιότητα των τελεστών θεωρούμε πως ακολουθούν τους αντίστοιχους κανόνες της γλώσσας C. Επίσης, η γλώσσα buzen υποστηρίζει την ύπαρξη C-like σχολίων σε οποιοδήποτε σημείο του προγράμματος (*/\* σχολio \*/*). Σημειώνεται τέλος ότι η γραμματική της γλώσσας, στη μορφή που δίνεται, δεν ανήκει στον τύπο LL(1). Κατά συνέπεια για να μπορεί να χρησιμοποιηθεί στην κατασκευή συντακτικού αναλυτή αναδρομικής κατάβασης θα πρέπει να τροποποιηθεί κατάλληλα.

1. Να υλοποιηθεί ένας προεπεξεργαστής και ένας λεκτικός και συντακτικός αναλυτής για την παραπάνω γλώσσα με χρήση των μεταεργαλείων flex και bison. Ο parser που θα υλοποιήσετε θα πρέπει να δέχεται ως είσοδο ένα κείμενο γραμμένο στην παραπάνω γλώσσα το οποίο θα έχει κατάληξη ".buz" και θα ελέγχει αν το πρόγραμμα είναι συντακτικά ορθό.

Αρχικά θα καλείται ο προεπεξεργαστής ο οποίος θα παίρνει ως είσοδο ένα πρόγραμμα γραμμένο στην παραπάνω γλώσσα και θα παράγει ένα ενδιάμεσο πρόγραμμα χωρίς σχόλια, το οποίο θα δίνεται ως είσοδος για λεκτική και συντακτική ανάλυση. Ο συνολικός parser θα καλείται από τη γραμμή εντολών ως εξής:

```
prompt> myParser.exe file.buz
```

και θα επιστρέφει διαγνωστικό μήνυμα για το αν ήταν ορθά γραμμένο, ή κατάλληλο μήνυμα σφάλματος (Θα πρέπει να φαίνεται η γραμμή όπου υπάρχει το σφάλμα)

## 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 Τυπωμένη Αναφορά 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 διαχωρισμένα με το χαρακτήρα \_ από τον μικρότερο στο μεγαλύτερο (π.χ. 3000\_3543\_4788\_4972.zip), και να σταλεί (ΥΠΟΧΡΕΩΤΙΚΑ) με email στο nikolako@... με θέμα ASKISI ARXES GLWSSWN. Στο μήνυμα να αναφέρονται τα ονοματεπώνυμα, το έτος και οι αντίστοιχοι αριθμοί μητρώου των μελών της ομάδας.

Για τυχόν απορίες σχετικά με την άσκηση τηρούνται ώρες γραφείου Δευτέρες 12:00-14:00 στο γραφείο του υποψήφιου διδάκτορα Θάνου Νικολακόπουλου στα ΠΡΟΚΑΤ δίπλα στο γραφείο του κ.Τσακαλίδη.