

Τμήμα Πληροφορικής Α.Π.Θ.
Προαιρετική Εργασία στο μάθημα
Σχεδίαση Γλωσσών Προγραμματισμού

Διδάσκων: Γ. Μακρής

Ακ. Έτος: 2018-2019

Bonus βαθμού ή απαλλαγή από εξέταση

Η προαιρετική εργασία εκπονείται **αποκλειστικά σε ατομική βάση** και το όφελος που θα έχουν οι φοιτητές ανάλογα με την περίπτωση είναι:

- **απαλλαγή από την εξέταση του Σεπτεμβρίου**
οι φοιτητές αυτοί θα προσέλθουν στην εξέταση μόνο για να δηλώσουν ότι παρέδωσαν ή πρόκειται να παραδώσουν εργασία· οι ολοκληρωμένες εργασίες που εκτελούνται σωστά **κατά την επίδειξη του αποτελέσματος στο διδάσκοντα** θα βαθμολογείται με ΑΡΙΣΤΑ (10)
- **δυνατότητα να παραδοθεί μέρος μόνο της εργασίας**
οι φοιτητές που θα επιδείξουν ολοκληρωμένη λεξική, συντακτική και σημασιολογική ανάλυση με πίνακα συμβόλων και των οποίων ο μεταγλωττιστής θα εκτυπώνει το συντακτικό δέντρο του πηγαίου προγράμματος (αντί να παράγει κώδικα στη γλώσσα-στόχο) **κατά την επίδειξη του αποτελέσματος στο διδάσκοντα** θα δικαιούνται 30% bonus στον τελικό βαθμό της εξέτασης

Αξιολόγηση εργασίας

Για να βαθμολογηθεί η εργασία, θα πρέπει **να παρουσιαστεί στο διδάσκοντα**. Η συνάντηση θα κανονιστεί σε συνεννόηση με το διδάσκοντα πριν από το τέλος της εξεταστικής, στην οποία θα περαστεί ο βαθμός (μπορεί ο βαθμός να περαστεί και στην εξεταστική του Σεπτεμβρίου). Κατά την παρουσίασή σας θα περιγράψετε τη σχεδίαση του μεταγλωττιστή, θα επιδείξετε σε φορητό υπολογιστή τη λειτουργία του και θα απαντήσετε σε σχετικές διευκρινιστικές ερωτήσεις. Το παραδοτέο για να είναι αποδεκτό πρέπει να περιλαμβάνει:

- σε ηλεκτρονική μορφή
 - i. τον κώδικα της γλώσσας σε πηγαία (source) και εκτελέσιμη (executable) μορφή
 - ii. αρχεία κειμένου ASCII με παραδείγματα προγραμμάτων έτοιμα για μεταγλώττιση και εκτέλεση
- αναφορά με
 - i. συνοπτική περιγραφή της γλώσσας και της υλοποίησης της
 - ii. παραδείγματα και αποτελέσματα από τη μεταγλώττιση τους και την εκτέλεση του κώδικα μηχανής

Αν ο διδάσκων κρίνει ότι το πρόγραμμα, που παραδίνετε δεν είναι αποτέλεσμα προσωπικής εργασίας, τότε δε θα λαμβάνεται υπόψη στον τελικό βαθμό.

Υλοποίηση εργασίας

Ο μεταγλωττιστής μπορεί να υλοποιηθεί στη γλώσσα προγραμματισμού της προτίμησής σας αρκεί να έχετε στη διάθεσή σας τα απαραίτητα εργαλεία (π.χ. flex και yacc ή κλώνους για γλώσσες προγραμματισμού εκτός της C). Τα περισσότερα εργαλεία αυτού του τύπου λειτουργούν όπως περιγράφεται στις σημειώσεις – διαφάνειες και όπως φαίνεται από την υλοποίηση του πρότυπου εκπαιδευτικού μεταγλωττιστή, που ο κώδικάς του (σε γλώσσα C) διατίθεται στα εργαστηριακά μαθήματα Lex Analysis Lab, Parsing Lab, Syntax Tree Lab και Code Generation Lab, που έχουν αναρτηθεί στην ιστοσελίδα του μαθήματος στο <http://elearning.auth.gr>.

Τις τελευταίες εκδόσεις των εργαλείων flex και yacc σε εκτελέσιμη μορφή για windows μπορείτε να τις βρείτε στη διεύθυνση <http://gnuwin32.sourceforge.net/packages.html>. Τεκμηρίωση για τη χρήση τους

και τη χρήση των κλώνων τους μπορείτε να βρείτε στη διεύθυνση <http://dinosaur.compilertools.net/>. Ένας πλήρης κατάλογος άλλων σχετικών εργαλείων υπάρχει στη διεύθυνση <http://catalog.compilertools.net/>. Ο μεταγλωττιστής που θα κατασκευάσετε θα κάνει μετάφραση στη γλώσσα της μηχανής MIX, που περιγράφεται παρακάτω.

Λεξική Δομή Γλώσσας Προγραμματισμού

1. Όλα τα σύμβολα της γραμματικής που ακολουθεί με πεζούς χαρακτήρες αναπαριστούν κυριολεκτικές τιμές (literals), που είναι δεσμευμένες λέξεις και δεν επιτρέπεται η χρήση τους ως ονόματα σε μεταβλητές.
2. Οι μη αλφαριθμητικοί χαρακτήρες, που περικλείονται σε ' και ', αναπαριστούν την κυριολεκτική τιμή, που αποτελείται μόνο από τους μη αλφαριθμητικούς χαρακτήρες.
3. Οι υπόλοιπες κυριολεκτικές τιμές είναι οι **id** και **num**, που ορίζονται από τις παρακάτω κανονικές εκφράσεις (γίνεται διάκριση μεταξύ πεζών και κεφαλαίων – το * εκφράζει 0 ή περισσότερες εμφανίσεις, το ? εκφράζει 0 ή 1 εμφάνιση, το [...] εκφράζει διάστημα χαρακτήρων και το | εκφράζει εναλλακτικές):
 id = letter (letter | digit | ' _ ')*
 num = [1-9] digit* ('.' digit*)?
 letter = a | ... | z | A | ... | Z
 digit = 0 | ... | 9
4. Τα κενά (διάστημα, αλλαγές γραμμής, tabs) αγνοούνται, αλλά λαμβάνονται υπόψη ως διαχωριστές μεταξύ των κυριολεκτικών τιμών.

Σύνταξη Γλώσσας Προγραμματισμού

Τα μη τερματικά σύμβολα εμφανίζονται με κεφαλαίους χαρακτήρες.

```
PROGRAM    → class id { public static void main ( ) COMP-STMT }
COMP-STMT  → { STMT-LIST }
STMT-LIST  → STMT-LIST STMT
            | ε
STMT       → ASSIGN-STMT
            | FOR-STMT
            | WHILE-STMT
            | IF-STMT
            | COMP-STMT
            | DECLARATION
            | NULL-STMT
            | println (EXPR);
DECLARATION → TYPE ID-LIST;
TYPE       → int
            | float
ID-LIST    → id, ID-LIST
            | id
NULL-STMT  → ;
```

$ASSIGN-STMT \rightarrow ASSIGN-EXPR;$
 $ASSIGN-EXPR \rightarrow id = EXPR$
 $EXPR \rightarrow ASSIGN-EXPR$
 $\quad | \quad RVAL$
 $FOR-STMT \rightarrow \text{for} (OPASSIGN-EXPR; OPBOOL-EXPR; OPASSIGN-EXPR) STMT$
 $OPASSIGN-EXPR \rightarrow ASSIGN-EXPR$
 $\quad | \quad \epsilon$
 $OPBOOL-EXPR \rightarrow BOOL-EXPR$
 $\quad | \quad \epsilon$
 $WHILE-STMT \rightarrow \text{while} (BOOL-EXPR) STMT$
 $IF-STMT \rightarrow \text{if} (BOOL-EXPR) STMT ELSE-PART$
 $ELSE-PART \rightarrow \text{else} STMT$
 $\quad | \quad \epsilon$
 $BOOL-EXPR \rightarrow EXPR C-OP EXPR$
 $C-OP \rightarrow == | < | > | <= | >= | !=$
 $RVAL \rightarrow RVAL + TERM$
 $\quad | \quad RVAL - TERM$
 $\quad | \quad TERM$
 $TERM \rightarrow TERM * FACTOR$
 $\quad | \quad TERM / FACTOR$
 $\quad | \quad FACTOR$
 $FACTOR \rightarrow (EXPR)$
 $\quad | \quad - FACTOR$
 $\quad | \quad id$
 $\quad | \quad num$

Σημασία Γλώσσας Προγραμματισμού

Κάθε μεταβλητή πρέπει να έχει δηλωθεί και αρχικοποιείται με την τιμή 0. Μία μεταβλητή δεν επιτρέπεται να δηλωθεί ξανά, δηλαδή τα ονόματα των μεταβλητών με ίδια εμβέλεια είναι διακριτά μεταξύ τους. Οι μεταβλητές, οι σταθερές, αλλά και όλες οι εκφράσεις έχουν τύπο, που υπολογίζεται με βάση τους κανόνες τύπων της C++. Έτσι, ο αριθμός 3.14156 αναγνωρίζεται ως **float**, ενώ ο αριθμός 42 αναγνωρίζεται ως **int** και η έκφραση 3.14156 + 42 αποτιμάται ως **float**, δηλαδή όταν ένας από τους τελεσταίους είναι **float** σε έκφραση αθροίσματος, τότε ο άλλος μετατρέπεται σε **float** και η έκφραση γίνεται **float** (οι συνήθεις μετατροπές τύπων για αριθμητικούς τελεστές αναφέρονται στο κεφ. 5 της προδιαγραφής της C++ που θα βρείτε στο e-learning). Ο τύπος μιας έκφρασης εκχώρησης *ASSIGN-EXPR* δίνεται από τον τύπο της lvalue της εκχώρησης. Οι εντολές έχουν την οικεία σημασία εκτέλεσης που γνωρίζετε. Οι υποθετικές εντολές, καθώς και τα for, while loops εκτελούνται όπως και στη C++.

Πράξεις αριθμών

Οι ακέραιοι της γλώσσας είναι προσημασμένοι με αναπαράσταση στο δυαδικό σύστημα και μέγεθος λέξης 31 bits (μέγιστος ακέραιος $2^{30}-1=1.073.741.823$), που περιορίζεται από τις δυνατότητες αποθήκευσης της μηχανής εκτέλεσης. Αντίστοιχα η αναπαράσταση

±	e	f ₁	f ₂	f ₃	f ₄
---	---	----------------	----------------	----------------	----------------

όπου το εκθετικό μέρος e είναι με βάση το 64 και χρησιμοποιείται για αριθμούς κινητής υποδιαστολής της μορφής $\pm 0.f_1f_2f_3f_4 \times 64^{e-32}$ που αν το f_1 δεν είναι 0 εκφράζει αριθμούς στο εύρος από 64^{-32} μέχρι 64^{31} (περίπου 10^{-58} μέχρι 10^{56}) με μία ακρίβεια 5 δεκαδικών θέσεων.

Η σημασία των τελεστών πρόσθεσης, αφαίρεσης, πολλαπλασιασμού και αρνητικού προσήμου ορίζεται κατά τα γνωστά, αλλά οι πράξεις αυτές μπορεί να οδηγήσουν σε λάθη υπερχείλισης (overflow). Η ακέραια διαίρεση i/k επιστρέφει το αποκομμένο ηλίκιο της διαίρεσης του i με το k αγνοώντας το όποιο κλασματικό μέρος. Αυτό σημαίνει ότι γίνεται πάντα στρογγυλοποίηση προς τον μικρότερο ακέραιο. Η διαίρεση i/k απαιτείται να οδηγεί σε εξαίρεση αν είτε $k=0$, είτε το αποτέλεσμα είναι πάρα πολύ μεγάλο ή πάρα πολύ μικρό για να χωρέσει σε λέξη της μηχανής εκτέλεσης.

Οι τελεστές σύγκρισης $<$, $<=$, $>$, $>=$, $==$ και $!=$ έχουν την τυπική σημασία που προβλέπεται για τους προσημασμένους ακεραίους στη σημασία της C++.

Παράδειγμα 1^ο προγράμματος της Γλώσσας

Το παρακάτω πρόγραμμα στη Γλώσσα Προγραμματισμού της εργασίας υπολογίζει τους πρώτους 10 όρους της σειράς Fibonacci:

```
class Fibonacci {  
    public static void main ( )  
    {  
        int first, second, i, tmp;  
        first=0;  
        second=1;  
        i=0;  
        while (i<10)  
        {  
            i=i+1;  
            tmp=first+second;  
            println (tmp);  
            first=second;  
            second=tmp;  
        }  
    }  
}
```

Παράδειγμα 2^ο προγράμματος της Γλώσσας

Το παρακάτω πρόγραμμα στη Γλώσσα Προγραμματισμού της εργασίας υπολογίζει την παράσταση:

$$mo = \frac{3.5 + 5 + 6.5 + 8 + 9.5}{5}$$

```
class Example {  
    public static void main ( )  
    {  
        int c;  
        float x, sum, mo;  
        c=0;  
        x=3.5;  
        sum=0.0;  
        while (c<5)  
        {  
            sum=sum+x;  
            c=c+1;  
            x=x+1.5;  
        }  
        mo=sum/5;  
        println (mo);  
    }  
}
```

Παραγωγή κώδικα στη γλώσσα - στόχο

Στην περίφημη σειρά βιβλίων “The Art of Computer Programming”, που δημοσίευσε ο D.Knuth, χρησιμοποιείται ένας ιδεατός υπολογιστής που αποκαλείται MIX. Ο MIX διαθέτει γλώσσα μηχανής και συμβολικής αναπαράστασης (assembly) των εντολών της.

Η αρχιτεκτονική του MIX είναι μία απλοποιημένη εκδοχή της σχεδίασης των πραγματικών CISC CPUs και η συμβολική γλώσσα του, που λέγεται MIXAL, παρέχει ένα σύνολο βασικών εντολών, που είναι πολύ οικείες σε όποιον έχει εμπειρία προγραμματισμού σε γλώσσα assembly. Ο ορισμός του MIX και της MIXAL είναι αρκετά ισχυρός και πλήρης, ώστε να παρέχει μία ιδεατή πλατφόρμα ανάπτυξης σύνθετων προγραμμάτων, αρκετά κοντά στους πραγματικούς υπολογιστές, χωρίς όμως την πολυπλοκότητα της αρχιτεκτονικής των. Για αυτό το λόγο εξάλλου ο MIX και η MIXAL χρησιμοποιούνται ευρέως ως εκπαιδευτικό περιβάλλον εκμάθησης προγραμματιστικών τεχνικών.

Στην ιστοσελίδα <http://www-cs-faculty.stanford.edu/~knuth/taocp.html> θα βρείτε κάτω από τον τίτλο MIXware μία εκτενή λίστα με συνδέσμους σε open source εργαλεία συμβολομετάφρασης, αποσφαλμάτωσης και εκτέλεσης της γλώσσας MIXAL. Από τα εργαλεία αυτά, εγκαταστάθηκε με επιτυχία σε σύστημα Linux και φαίνεται αρκετά ώριμο το GNU MIX Development Kit, ενώ για συστήματα Win32 προτείνεται το περιβάλλον MIXBuilder. Ένα πολύ φιλικό στη χρήση περιβάλλον, ανεξάρτητο από λειτουργικό σύστημα (Java εφαρμογή) είναι το [JMixSim](http://home.telkomsa.net/dlaing/mix.html). Τέλος, από τη διεύθυνση <http://home.telkomsa.net/dlaing/mix.html> διατίθεται σε πηγαίο κώδικα C ένας μεταγλωττιστής της MIXAL και ένας προσομοιωτής της μηχανής MIX.

Αναλυτική περιγραφή της αρχιτεκτονικής της μηχανής MIX και της γλώσσας MIXAL θα βρείτε στο online βιβλίο <http://www.jklp.org/profession/books/mix/index.html> και στον οδηγό (manual), που διατίθεται στη διεύθυνση <http://www.gnu.org/software/mdk/manual/mdk.pdf>.

Τέλος, πρέπει να τονιστεί ότι η γλώσσα MIXAL υλοποιείται στα προαναφερόμενα εργαλεία με μικρές διαφορές (π.χ. στον ορισμό των σταθερών string) και γι' αυτό πρέπει ο κώδικας που θα δημιουργήσει ο μεταγλωττιστής σας να συμμορφώνεται με τις ιδιαιτερότητες της μηχανής MIX που χρησιμοποιείτε. Συμβουλευτείτε για το σκοπό αυτό τα παραδείγματα των MIXAL προγραμμάτων που συνοδεύουν την πλατφόρμα MIX που έχετε επιλέξει.