

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ

Θέμα (30%)

Δίνεται η παρακάτω γραμματική η οποία μπορεί να χρησιμοποιηθεί για τη δημιουργία κώδικα εντολών καταχώρησης και απλών αριθμητικών εκφράσεων.

Ζητείται να κατασκευάσετε συγκεκριμένο μεταφραστικό σχήμα για την παραγωγή ενδιάμεσου κώδικα τριών διευθύνσεων.

$E ::= E^1 + T | T$

$T ::= T^1 * F | F$

$F ::= -P | P$

$P ::= (E) | id | con$

Θέμα (40%)

(α) Γράψτε ό,τι γνωρίζεται για την βελτιστοποίηση (15%)

(β) Εξηγήστε την ή τις διαφορές ανάμεσα σε compilers και interpreters (10%)

(γ) Γράψτε μια ρουτίνα σε μορφή ψευδοκώδικα για την εισαγωγή ενός identifier στο τρέχον block του πίνακα συμβόλων οργανωμένου κατά blocks (εξηγήστε και περιγράψτε ότι δομές ή μεταβλητές χρησιμοποιήσετε) (15%)

Θέμα 10%

Περιγράψτε τις Γλώσσες που δηλώνουν οι παρακάτω κανονικές εκφράσεις:

1. $0(0|1)^*0$

2. $0^*10^*10^*10^*$

Θέμα 20%

Έστω η παρακάτω γραμματική:

$S \rightarrow id = E$

$\quad | id (E)$

$E \rightarrow E + id$

$\quad | id$

a) Κατασκευάστε μια «αριστερότερη παραγωγή» και το αντίστοιχο δένδρο ανίχνευσης για την πρόταση: **id (id + id)**

b) Είναι η παραπάνω γραμματική LL(1)? Εάν δεν είναι μετατρέψτε την σε LL(1) κάνοντας τους απαραίτητους μετασχηματισμούς εξηγώντας πώς και γιατί.

Θέμα 20%

Γράψτε τις ρουτίνες ενός recursive descent αναλυτή για την παρακάτω γραμματική:

$D \rightarrow T L$

$T \rightarrow int | boolean$

L -> id R
R -> , L | ε

Θέμα 10%

Μπορούν τα NFAs να αναγνωρίζουν περισσότερες Γλώσσες απο τα αντίστοιχα DFAs? Γιατί?

Θέμα 40%

Εξηγήστε γιατί η εντολή:

For V := B-C step 1 until B+C+S do S := D-V

δεν μεταφράζεται στον ακόλουθο κώδικα τριών διευθύνσεων:

```
T(1) := B-C
V(1) := T(1)
INC := 1
FIN := B+C+S
L(0) if V<FIN goto L(2)
      goto L(1)
L(1) T(2) := D-V
      S := T(2)
      V := V+T(1)
      Goto L(1)
L(2) ...
```

Δοκιμάστε να εντοπίσετε τα λάθη στον παραπάνω κώδικα και να δώσετε το σωστό, κατά τη γνώμη σας, κώδικα για την εντολή for.

Στη συνέχεια δοκιμάστε να προτείνετε βελτιστοποιήσεις στον κώδικα τριων διευθύνσεων (αν είναι δυνατόν)

Θέμα (40%)

Δίνεται η παρακάτω γραμματική η οποία περιγράφει απλές λογικές εκφράσεις.

Θεωρήστε ότι οι τελεστές +,*,~ αντιστοιχούν στους λογικούς τελεστές OR,AND,NOT και ο relop αντιστοιχεί στους <,>,<=,>=#.

1. $E ::= E^1 + T | T$
2. $T ::= T^1 * F | F$
3. $F ::= \sim F^1 | P$
4. $P ::= (E)$
5. $P ::= |id$
6. $P ::= |id^1 \text{ relop } id^2$

Ζητείται να κατασκευάσετε συγκεκριμένο μεταφραστικό σχήμα για την παραγωγή ενδιάμεσου κώδικα τριών διευθύνσεων :

- α. Για αριθμητική αναπαράσταση των λογικών εκφράσεων και,
- β. Για control flow αναπαράσταση.

Στην περίπτωση που κάνετε τροποποιήσεις ή χρησιμοποιήσετε κάποιες ρουτίνες πρέπει να περιγράψετε επακριβώς τι κάνατε και τι χρησιμοποιήτε.

Θέμα (20%)

(α). Σε top-down ανίχνευση υπάρχουν διάφορα προβλήματα. Περιγράψτε τα προβλήματα αυτά και τη λύση τους.

(β). Περιγράψτε τη λειτουργία ενός Interpreter

Θέμα (10%)

Περιγράψτε τη δομή και λειτουργία των Predictive Parsers.

Θέμα (30%)

A. Κατασκευάστε ένα μεταφραστικό σχήμα για τον κανόνα:

$S \rightarrow \text{for } V := E(1) \text{ step } E(2) \text{ until } E(3) \text{ do } S(1) \quad (1)$

με την υπόθεση ότι V είναι η μεταβλητή της εντολής for και ότι οι εκφράσεις E(2) και E(3) δεν υπολογίζονται με κάθε επανάληψη της ανακύκλωσης.

B. Για τον ίδιο κανόνα (1) σχεδιάστε το μεταφραστικό σχήμα για την περίπτωση που οι εκφράσεις E(2) και E(3) υπολογίζονται σε κάθε επανάληψη της ανακύκλωσης.