

## ΑΣΚΗΣΗ

Δίνεται η ακόλουθη γραμματική

$$\langle E \rangle ::= \langle E \rangle + \langle E \rangle$$

$$\langle E \rangle ::= \langle E \rangle * \langle E \rangle$$

$$\langle E \rangle ::= (\langle E \rangle)$$

$$\langle E \rangle ::= a | b | \gamma$$

1) είναι LL(1); αν όχι να γνυ

## Λύση

Δεν είναι LL(1) διότι οι 2 πρώτοι κανόνες παρουσιάζουν αμοιβαία αναδρομή

Για την εφευρεμένη γραμματική:

$$\langle E \rangle ::= (\langle E \rangle) \langle B \rangle / a \langle B \rangle / b \langle B \rangle / \gamma \langle B \rangle$$

$$\langle B \rangle ::= + \langle E \rangle \langle B \rangle / * \langle E \rangle \langle B \rangle / \epsilon$$

## ΘΕΜΑ (ΙΟΥΝΙΟΣ 2005)

Απόστολον ορισμο της μεταβλητης σε μια γλωσσα προγραμματισου.  
Ποια ειναι τα 4 συστατικα μερη μιας μεταβλητης? Αναφερετε 3 ιδιοτητες μιας μεταβλητης. Ποτε εχουμε ψευδωμια; Αναφερετε 2 περιπτωσης που μπορεί να εμφανιστει.

### ΑΠΑΝΤΗΣΗ=

- ⊙ Μεταβλητη ειναι μια τετραδα που αποτελείται απο ένα ονομα, ένα συνολο ιδιοτητων, μια αναφορά και μια τιμη.
- ⊙ Τα 4 συστατικα μερη μιας μεταβλητης ειναι το ονομα, οι ιδιοτητες, η αναφορά και μια τιμη της μεταβλητης.

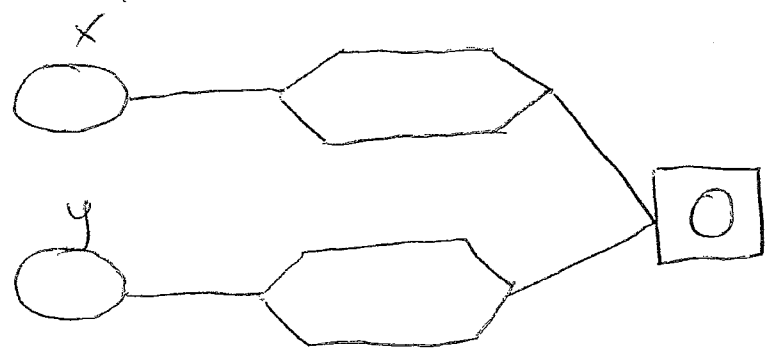
### Σχηματικά



Πρέπει να σημειώσουμε ότι το ονομα και η αναφορά, δηλαδή η θέση μνήμης είναι αμεταβλητα, ενώ η τιμή μπορεί να αλλάξει, αλλά κάθε στιγμή υπάρχει κάποια τιμή στο κελί (και το αόριστο θεωρείται τιμή).

ιδιότητες αυτές μπορούν να ενοποιηθούν μόνο με τη βοήθεια της διάρκειας της μεταγωγής, ή να μεταβληθούν στην πορεία εξέλιξης του προγράμματος. Ο καθορισμός των ιδιοτήτων μπορεί να γίνει με μια δήλωση ή μπορούν να οριστούν εκ νέου, με ανάθεση νέας τιμής στο όνομα της μεταγωγής

Έστω δύο μεταγωγές  $x$  και  $y$ , οι οποίες αναφέρονται στην ίδια θέση μνήμης όπως φαίνεται στο παρακάτω σχήμα



Η μεταγωγή  $x$  αναφέρεται γενικά (αλλά) της  $y$ , και αντίστροφα. Μια δήλωση προγραμματισμού, που επιτρέπει σε μια μεταγωγή να είναι το γενικό μιας άλλης, σημαίνει ότι επιτρέπει την γενικότητα. Η γενικότητα συνάπτει στην Pascal όταν στις τμητικές παραφύσεις χρησιμοποιούν τη λέξη var και στη C όταν οι τμητικές παραφύσεις δηλώνονται σαν δείκτες

εγγυησέ τι είναι ισοδυναμία ονομάτων και τι ισοδυναμία-36-  
ομοίων στον αγορά τους χωρίς δεδομένων. Ποια από τις δύο  
χρησιμοποιεί η C;

Απάντηση:

Κανόνας 1: Δύο μεταβλητές θα αναγνωρίζονται ότι είναι του  
ίδιου τύπου αν και μόνο αν, είτε εγγυηθούν (π.χ.) είτε  
εγγυηθούν με τα ίδια αναγνωριστικά τύπου

Κανόνας 2: Δύο μεταβλητές θα αναγνωρίζονται ότι είναι του  
ίδιου τύπου αν και μόνο αν, τα βασικά μέρη των τύπων  
τους είναι ίδια από κάθε άποψη

Ο κανόνας 1 ονομάζεται ισοδυναμία ονομάτων ενώ ο  
κανόνας 2 ονομάζεται ισοδυναμία δομών

Η C χρησιμοποιεί την ισοδυναμία δομών καθώς οι ακολουθίες  
σημάτων είναι ισοδύναμες

struct circle

typedef circle kyklos;

```
int x, y;  
float radius;  y  
};
```

Δήλωση:  
kyklos k;

Δήλωση:  
struct circle c;

Οι μεταβλητές c και k είναι του ίδιου τύπου

Ποια είναι τα 2 βασικά κριτήρια με βάση τα οποία σχεδιάζονται και ταξινομούνται οι εντοχές επανάληψης των προγραμμάτων; Δώστε ένα παράδειγμα από κάθε είδος εντοχής επανάληψης στη C.

Τα 2 κριτήρια με βάση τα οποία σχεδιάζονται και ταξινομούνται οι εντοχές επανάληψης είναι τα εξής:

\* Πώς ελέγχεται η επανάληψη? (με λογική έκφραση ή με φερένη)

\* Σε ποιο σημείο του βρόχου θα βρίσκεται ο μηχανισμός επανάληψης; (στην αρχή ή στο τέλος του βρόχου)

Οι εντοχές επανάληψης στη C είναι οι ακόλουθες:

- α) έντοξη for  
for (έκφραση-1; έκφραση-2; έκφραση-3).  
έντοξη;

Αν υπάρχουν περισσότερες από μία εντοχές που θέλω να επαναληφθούνται περιέχονται σε {}· Η έκφραση-1 είναι συνήθως μία αρχικοποίηση ενός μετρητή, η έκφραση-2 είναι συνήθως έλεγχος μιας συνθήκης και η έκφραση-3 είναι συνήθως αύξηση ενός μετρητή

- β) έντοξη while  
while (συνθήκη)  
έντοξη ή  
while (συνθήκη) {  
έντοχές  
}
- γ) έντοξη do-while  
{  
do  
έντοχές;  
while (συνθήκη);

Η συνθήκη στο while ελέγχεται στην αρχή της επανάληψης στο do-while στο τέλος της επανάληψης. Άρα οι εντοχές στο do-while εκτελούνται έστω 1 φορά

12  
Τίτλοι είναι οι φράσεις μεταγράφως ενώ προγράμματος.  
Τίτλοι είναι οι βασικές περιλήψεις και φάση; Πόσα φωνήες  
χρησιμοποιούνται με την επινόηση των δύο πρώτων φάσεων;

Απάντηση:

1 μεταγραφή (μεταφράζω) χωρίζεται σε επιφώνη φάσεις (phases),  
και με αυτό τις οποίες δέχεται ως είσοδο ένα πρόγραμμα  
βασισμένο με το αρχικό, σε κάποια μορφή και δίνει ως έξοδο  
το ίδιο πρόγραμμα σε μια άλλη μορφή. Οι φάσεις μεταγράφως  
είναι οι εξής:

- Δεξιά ανάγνωση: Στη φάση αυτή διαβάζεται το αρχικό πρόγραμμα,  
έναν χαρακτήρα κάθε φορά και γίνεται με ομαδοποίηση των  
χαρακτήρων σε γερμικές μονάδες (tokens) που δίνονται ως είσοδος  
στην επόμενη φάση. Η φάση και η μορφή των γερμικών μονάδων  
εξαρτώνται από την αρχική φάση καθώς και από άλλους παρά-  
γοντες που σχετίζονται με τη σχεδίαση των μεταφράσεων.  
Τα στοιχεία των μεταφράσεων, οι λέξεις κλειδιά, οι σταθερές, οι  
τελεστοί και οι διαχωριστές αναφέρονται συνήθως παραδείγματα  
Γερμικών μονάδων

Συνακτική ανάγνωση: Κατά τη συνακτική ανάγνωση διαβάζονται οι γερμικές  
μονάδες και ομαδοποιούνται σε συνακτικές μονάδες βάσει μιας  
προσφρασής που περιγράφει τη σύνταξη της αρχικής φάσης.  
Στο τέλος αυτής της φάσης έχει σχηματιστεί ένα δέντρο που  
αντικατοπτρίζει τη συνακτική δομή των αρχικών προγράμματος.  
Το δέντρο αυτό ονομάζεται συνακτικό δέντρο. Κατά τη  
διάρκεια της συνακτικής ανάγνωσης, ορίζεται από τις γερμικές  
μονάδες τονόθετοίαν των συμβόλων (symbol table)

πινακός αυτός συμπληρώνει σωστά τη μορφή με αναλυτικά  
 να ορίσει τις προτάσεις με τον τρόπο των ελεύθερων  
 μεταβλητών, το μέγεθος των ειδών, τον αριθμό και τον τρόπο των  
 παραμέτρων, των υποπροτάσεων της συνθήκης με τις εφόρους  
 γάβια της προαγωγικής που χρησιμοποιούν αυτές τις πληροφορίες.  
 Η διαχείριση των πινακίδων, αν και δεν αποτελεί γρήγορο  
 γάβια της προαγωγικής, κατέχει εξέχουσα θέση στη σχεδίαση ενός  
 προαγωγικού και επιρρέει σημαντικά στην αρχή των  
 διαδικασιών της προαγωγικής.

Συμβασιολογική διατύπωση: Στη γάβια αυτή, που συχνά αναφέρεται  
 σταυρική συμβασιολογική γάβια με πρόταση για συμβασιολογική  
 γάβια και συνηθισμένη μορφή που χρησιμοποιείται με την  
 παραγωγή του ενδιαφέροντος κώδικα. Το αποτέλεσμα αυτής της γάβιας  
 είναι να γίνει το συντακτικό δέντρο, λιγότερο προσαρμοσμένο και  
 υπερηχημένο με πληροφορίες συμβασιολογικής γάβιας. Για τον λόγο  
 αυτό η αυτή συμβασιολογική διατύπωση αφορά στον έλεγχο των  
 ελέγχων προαγωγικών και το γεγονός κάθε γεγονός έχουν  
 τους κώδικες που προέρχονται από τις προαγωγικές της  
 αρχικής γάβιας και οι παράμετροι των υποπροτάσεων  
 σύμφωνα με τον τρόπο που καθορίζονται στις διατάξεις των

Παραγωγή ενδιαφέροντος κώδικα: Στη γάβια αυτή παράγεται  
 κώδικας με τις ενδιαφερόμενες με βάση το συντακτικό  
 δέντρο. Ο πιο σημαντικός τύπος υπάρχει ενδιαφέροντος  
 κώδικας είναι για να υπάρχει γάβια διαχωρισμός μεταξύ  
 των εφόρων και των εφόρων που προαγωγική

5 ενδιαφέρουσες γλώσσες ~~χρησιμοποιούνται~~ χρησιμοποιούνται συνηθέστερα

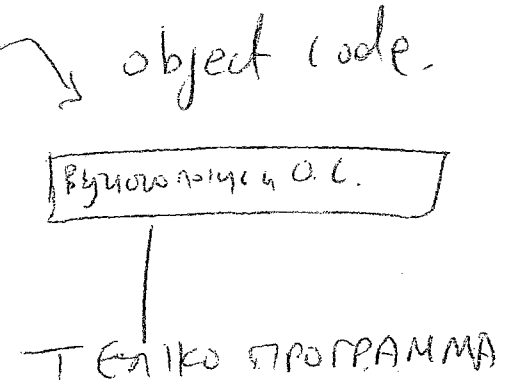
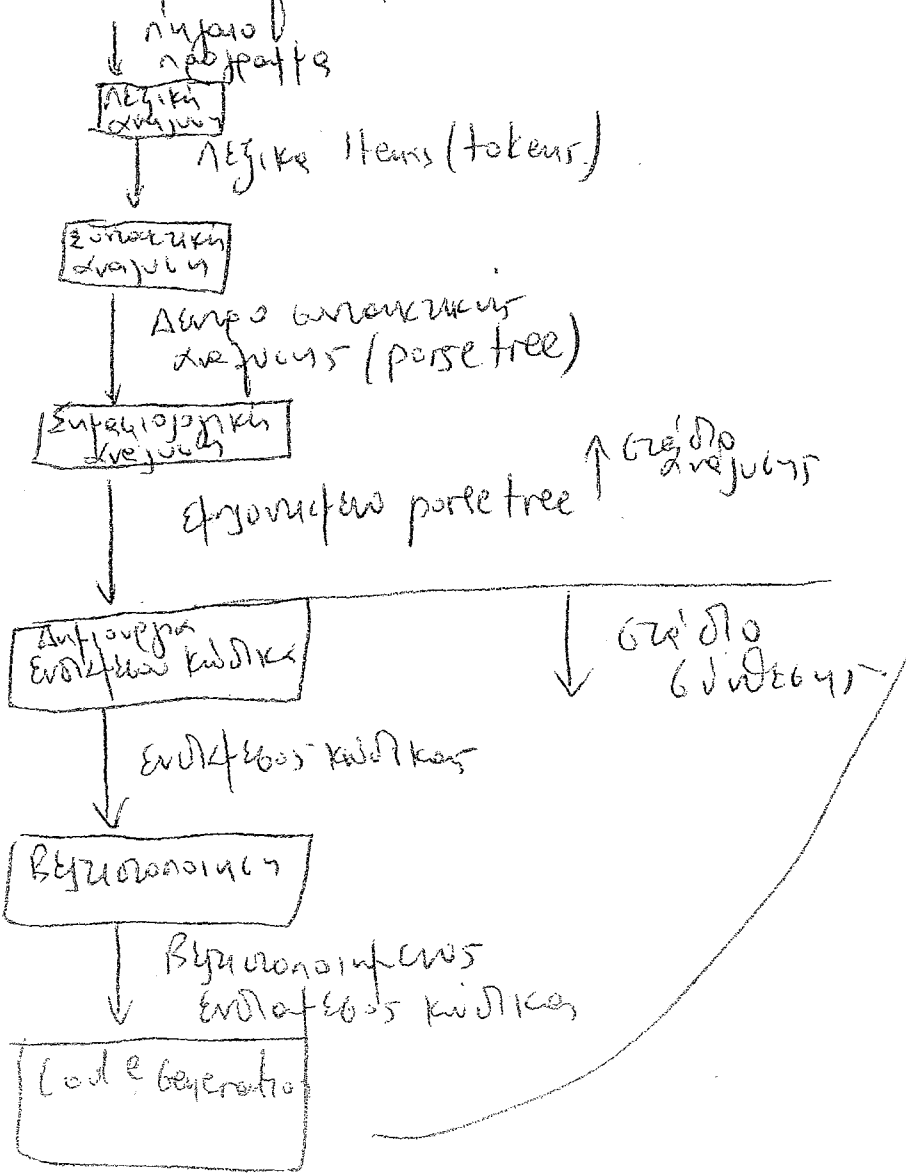
- Οι τρέφοδες
- Ο επιδοτικός κώδικας
- Τα αλληλεπενευστικά στοιχεία

- Βελτιστοποίηση

Η βελτιστοποίηση έχει ως σκοπό να κάνει αποδοτικότερο το τελικό πρόγραμμα. Κύριες μέθοδοι απόδοσης είναι σε αυτή την περίπτωση η ταχυστική εκτέλεση του τελικού προγράμματος και το μέγεθος του. Η βελτιστοποίηση είναι φυσικά προαιρετική σε ένα προγραμματιστή και μπορεί να εφαρμοστεί στον ενδιαφέρον ή στον τελικό κώδικα.

- Πράσινη Τυλική Κώδικα

- Επίσης αυτή η τεχνική αφορά τον ενδιαφέρον κώδικα σε τυλικό κώδικα. Το αποτέλεσμα της είναι το τυλικό μεταγλωσσισμένο πρόγραμμα.
- Για την υλοποίηση του γρήγορου διαλογισμού χρησιμοποιείται ως εργαλείο flex
- Για την υλοποίηση του βηματικού διαλογισμού χρησιμοποιείται ως εργαλείο bison





Είναι η  $z_{f, l}$  και η  $z_{f, r}$  μας μεταβλητές? Αναφέρετε δύο διαφορετικούς τρόπους με τους οποίους γίνεται η διακρίση των 2 περιπτώσεων, σε 2 διαφορετικές ημι-συνάρτησεις.

Απάντηση:

Οι όροι  $z_{f, l}$  και  $z_{f, r}$  αναφέρονται στην αναγωγή και στη  $z_{f, l}$  μιας μεταβλητής αντίστοιχα. Οι όροι αυτοί προέρχονται από το διαφορετικό τρόπο με τον οποίο χειρίζονται το σήμα μιας μεταβλητής, στην αριστερή ( $l = left$ ) και στη δεξιά ( $r = right$ ). Αξιώστε το γεγονός αυτό.

Στη ημί-συνάρτηση  $P_{256}$  η  $z_{f, l}$  μιας μεταβλητής προσδιορίζεται μέσω του σήματος της ενώ η αναγωγή (διεύθυνση) μιας μεταβλητής προσδιορίζεται με το bit. Στη ημί-συνάρτηση  $C$  η  $z_{f, l}$  μιας μεταβλητής προσδιορίζεται μέσω του σήματος της, ενώ η διεύθυνση της προσδιορίζεται με το bit 30 &

Παράδειγμα C

```
int * x, y = 5;
```

```
x = &y; // στο δείκτη καταχωρείται  
// η διεύθυνση της y
```

```
printf ("%d", *x); // τονώνει τον  
// το περιεχόμενο της  
// μεταβλητής στην  
// απόρ. δείκτη *x
```

Παράδειγμα Pascal

```
x: integer; // το x δηλώνεται  
ως δείκτης στο  
integer;
```

ΕΜΑ

επιγραφή το είδος δείκτη που ορίζεται για τις μεταβλητές  
δηλώνεται στη γραμμή C:

- float \*a [4];
- float (\*b) [4];
- double (\*c [4]) ();
- double (\*dc) [4];

ΑΠΑΝΤΗΣΗ:

- float \*a [4]; Το a δηλώνεται ως ένας πίνακας 4 στοιχείων όπου το καθένα από αυτά είναι δείκτης σε float
- float (\*b) [4]; Το b δηλώνεται ως δείκτης σε πίνακα 4 ή στοιχεία όπου το καθένα είναι float
- double (\*c [4]) (); Το c δηλώνεται ως πίνακας 4 δεικτών ή όπου το καθένα στοιχείο του πίνακα είναι δείκτης σε συνάρτηση χωρίς ορισμό που επιστρέφει αποτέλεσμα < double.
- double (\*dc) [4]; Το d δηλώνεται ως ένας πίνακας 4 δεικτών ή, κάθε στοιχείο του οποίου είναι δείκτης σε συνάρτηση χωρίς ορισμό που επιστρέφει αποτέλεσμα < double