

Αρχές Γλωσσών Προγρ.
κ' Μεταγρ.

→ ^{skleene star} digit* (zov. éva yngio maloudoumeno ano 0 ÷ ÷llo yngio)

Pascal:

n.x. 1
234
234, 787
234, 787 E + 2, 7 (éθetikoi arithmói)

n.x. SOS

ðvadikoi arithmói → perittos arithmós ággvov.
→ kanoniké ékφραsev

$$0^* 10^* (10^* 10^*)^*$$

$$\textcircled{\eta} (0^* 10^* 1)^* 0^* 10^*$$

(kavéva)
 $X^* \rightarrow 0 \text{ ÷ } \text{peribótera } X$

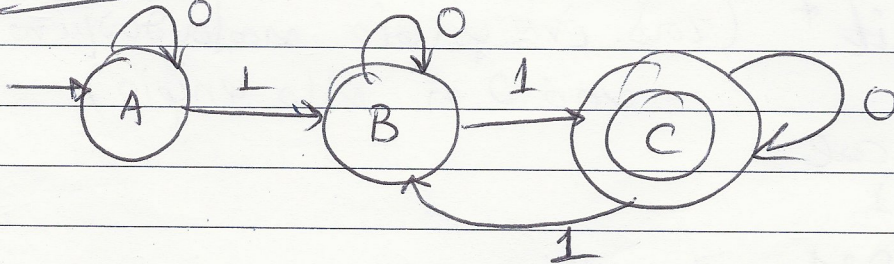
$$X X^* = X^+ \rightarrow 1 \text{ ÷ } \text{peribótera } X$$

$$X/\epsilon = X? \rightarrow 0 \text{ ÷ } 1$$

Π. 7.2

δυναμικοί αριθμοί με άρτιο
αριθμό άεδων. (σχήμα στις διαφάνειες.)

ζουμ 2!



Κανονικές εκφράσεις:

26/03/12
(10)

Αρχές Γλωσσών Προγραμματισμού κ' Μεταφραστών

① Να περιγραφεί η γλώσσα που παράγεται από τις ακόλουθες R.E.

(a) $b^* a b^* a b^*$: ακριβώς 2 α ή περισσότερα β.

(β) $(0^* 001)^* 0^*$: κάθε άσβος αν υπάρχει ακολουθείται τουλάχιστον 2 μηδενικά.

② Να περιγραφείτε με R.E τις παρακάτω γλώσσες

(α) $\Sigma = \{a, b\}$

"Όλες οι λέξεις έχουν άρτιο αριθμό α"

(β) $\Sigma = \{0, 1\}$

"Οι λέξεις της γλώσσας έχουν την ιδιότητα κάθε τρίτο σύμβολο να είναι 0"

22/03/20

(a) $(b^*ab^*ab^*)^* / b^*$

n

$(b^*ab^*ab^*)^* b^*$

(b)



$(0|1) (0|1) 0$ * $(0|1)? (0|1)? (0|1)$

↑ κανένα η 1

κάνοντας *

υποπεριλήψεις

$(0|1) (0|1) 0$ * $(0|1|ε) (0|1|ε)$

n

πάντα υποπεριλήψεις = πεπερασμένα αυτομάτα

3 Να κατασκευαστεί DFA που αναγνωρίζει

όλα τα string που περιέχουν:

(α) 3 συνεχόμενα 1 $\{0,1\}^* = \Sigma^*$ (10)

(β) όταν περιέχουν 1 θα πρέπει αυτά να

εμφανίζονται ως ομάδες των 3 τριών.

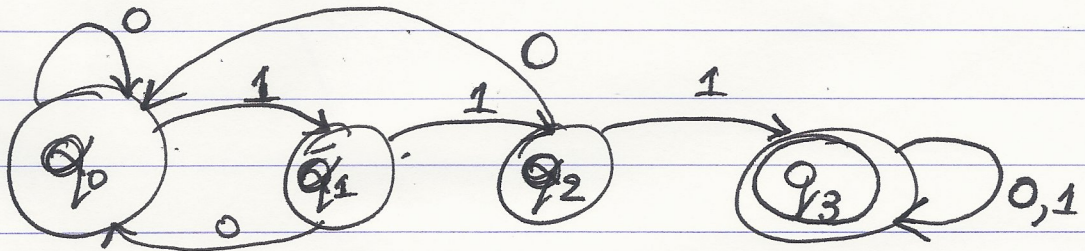
(γ) θα πρέπει τα '1' να εμφανίζονται σε ομάδες 11011 των τριών π.χ. 11111 $\notin L$

"Οι γέφυρες του Πανεπιστημίου είναι 0"

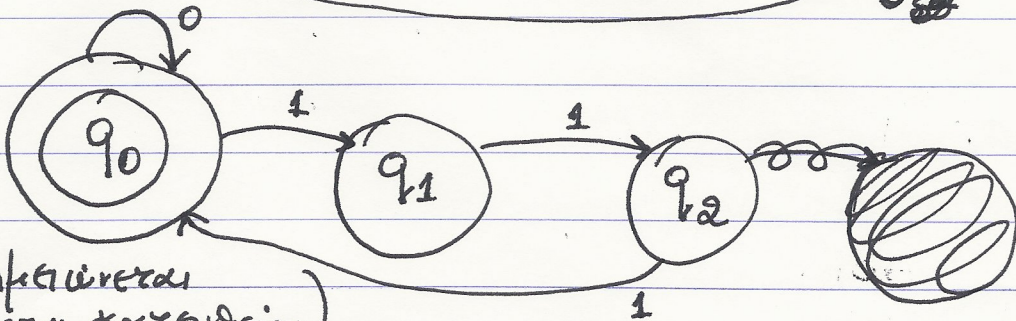
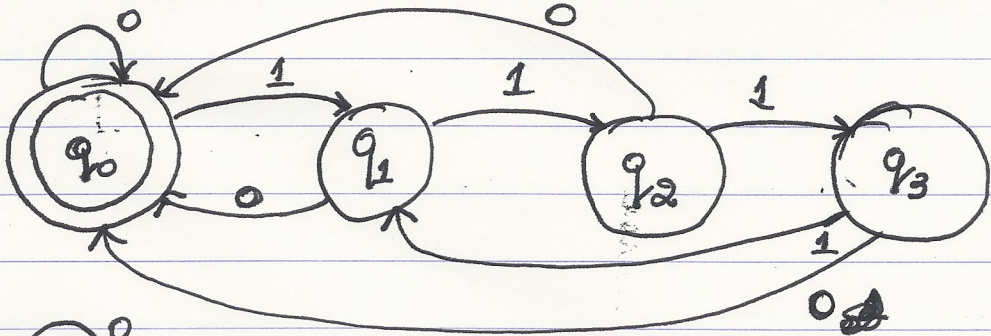
Κάθε άλλο αριθμό οτιδήποτε είναι 0

→ Ότι δε σηκώνεται
πάει σε κατάσταση καταβόθρας
(trash)

(α)

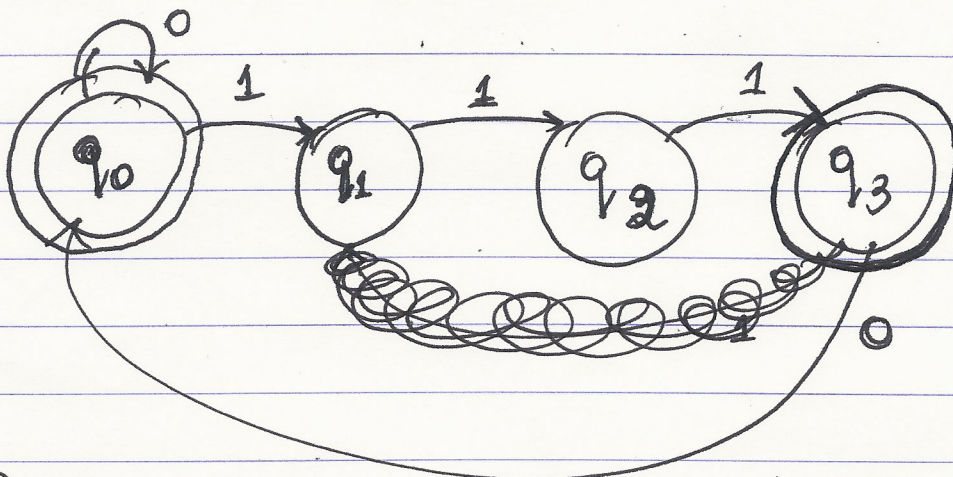


(β)



(Ότι δε σηκώνεται
αποπνίγεται καταβόθραν)

(γ)

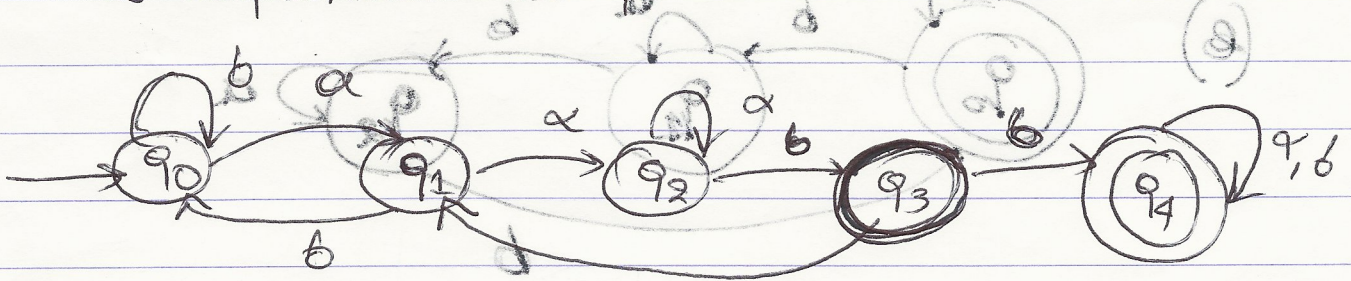


* από από $q_3 \rightarrow q_1$ με 1 μπορεί να έχω 6-άδες
9-άδες κτλ. ενώ δέλω μόνο τω 3.

27/03/12

Αρχές Γλωσσών Προγραμματισμού κ' Μεταφραστήων

1) Να περιγράψετε τη γλώσσα που αναγνωρίζει το παρακάτω αυτόματο:



περιέχει μια φορά οπωδήποτε
των υποέτη: $a^2 b^2$ κ' από p^3 κει λέρα οτιδήποτε.
είτε a είτε b

2) Να κατασκευαστούν τα παρακάτω DFA:

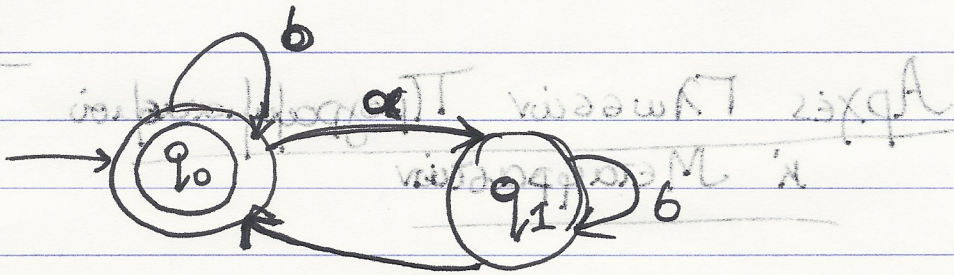
a) Άρτιο μήκος από a .

b) Το μήκος των b είναι πολλαπλό του 3.

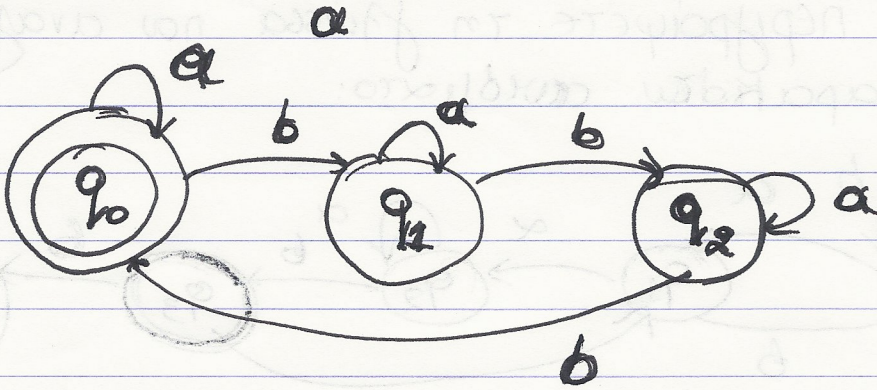
γ) Ταυτόχρονα τα (a), (b)

8/1/80/138

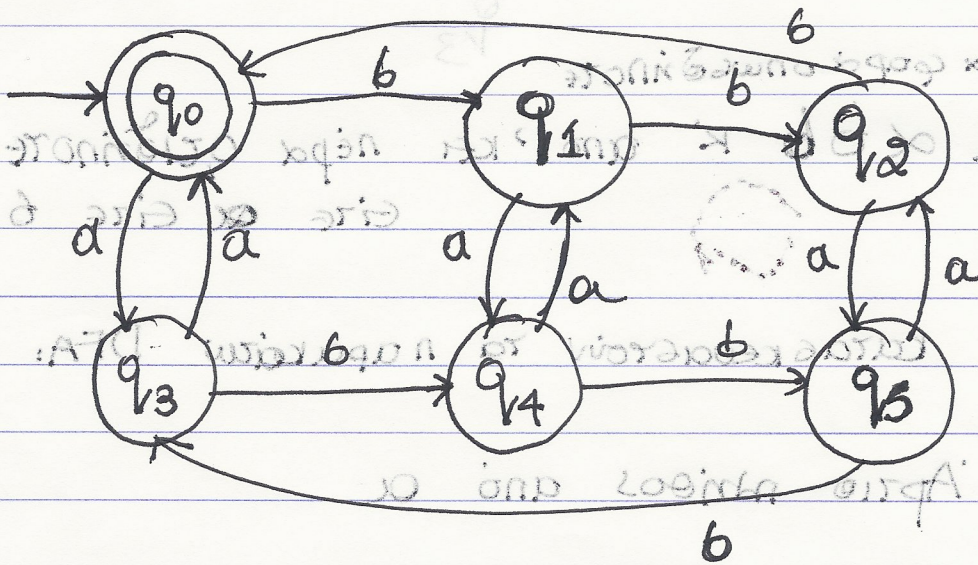
(a)



(b)



(c)



(d) to (f) are similar to (a) and (b).

(g) is similar to (c).

3) Έστω η γραμματική: (4)

$\langle S \rangle \rightarrow a \langle A \rangle$

$\langle A \rangle \rightarrow a \langle B \rangle$

$\langle B \rangle \rightarrow b \langle B \rangle$

$\langle B \rangle \rightarrow a$

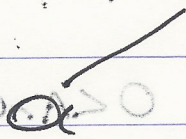
ή $\langle B \rangle \rightarrow a$ από τις 2 περιπτώσεις.

α) Να γράψετε το δέντρο συντακτικής ανάλυσης για τη λέξη $aaabaaa$

β) Να περιγράψετε τη γλώσσα που παράγει η γραμματική.

(α)

$\langle S \rangle$



α

$\langle B \rangle$

b

$\langle B \rangle$

a

(β)

$\langle T \rangle \rightarrow a^r b^r a^r, r \geq 0$

φορές

αριθμ

αριθμ

αριθμ

4) Να κατασκευάσετε τις γραμματικές για τις παρακάτω γλώσσες που ορίζονται επί του αλφαβήτου $\Sigma = \{0, 1\}$

α) L_1 : Όλες οι λέξεις που αρχίζουν κι τελειώνουν με το ίδιο σύμβολο

β) L_2 : Περιέχει λέξεις περιττού μήκους με μεσαίο σύμβολο το 0.

γ) L_3 : Περιέχει παλίνδρομα.

δ) L_4 : Περιέχει το σύνολο των λέξεων στις οποίες τα 0 είναι περισσότερα (από τα 1).

α) $\langle S \rangle \rightarrow \langle A \rangle 0 \langle A \rangle | 1 \langle A \rangle 1$
 $\langle A \rangle \rightarrow 0 \langle A \rangle | 1 \langle A \rangle | \epsilon$

β) $\langle S \rangle \rightarrow 0 \langle S \rangle 0 | 1 \langle S \rangle 1 | \epsilon$

γ) $\langle S \rangle \rightarrow 0 \langle S \rangle 0 | 1 \langle S \rangle 1 | \epsilon$

δ) $\langle S \rangle \rightarrow \langle T \rangle 0 \langle T \rangle$
 $\langle T \rangle \rightarrow \langle T \rangle 1 \langle T \rangle | 0 \langle T \rangle | \epsilon$

↓
 επάνω
 τη σύμμετρία
 του β.

STANDARD ΘΕΜΑ.

5

$$\langle S \rangle \rightarrow \langle A \rangle a / b$$

$$\langle A \rangle \rightarrow \langle A \rangle c / \langle S \rangle d / \#$$

d) Είναι LL(1)?

β) Αν όχι, μετατρέψτε σε LL(1).

a) Όχι γιατί έχω έμφυση αριστερή αναδρομή.

$$\beta) \langle S \rangle \rightarrow \langle A \rangle a / b$$

$$\langle A \rangle \rightarrow \langle A \rangle c / \langle A \rangle ad / bd / \#$$

Πάνω π) παραγωγές δ' αυτές που έχουν αριστερή αναδρομή κ' δ' αυτές που δεν έχουν.

Στο πρόχειρο: (ως βρέθηκα)

$$\langle A \rangle \rightarrow \langle A \rangle c / \langle A \rangle ad$$

$$\langle A \rangle \rightarrow bd / \#$$

$$\langle A \rangle \rightarrow bd \langle B \rangle / \# \langle B \rangle$$

$$\langle B \rangle \rightarrow c \langle B \rangle / ad \langle B \rangle / \epsilon$$

Απαιτούμε για τεχνική δεξιά αναδρομή.

Δίνεται η παρακάτω BNF γραμματική

$$\langle S \rangle ::= \langle S \rangle (\langle S \rangle) | \langle S \rangle [\langle S \rangle] | \epsilon$$

a) Περιγράψτε τα χαρακτηριστικά των μελών της γλώσσας που ~~παράγεται~~ παράγεται από την παραπάνω γραμματική.

b) Χρησιμοποιώντας την παραπάνω γραμματική, παρουσιάστε δένδρο συντακτικής ανάπτυξης

(parse tree) για την ακολουθία " $([] ([]))$ "

κ' αποφανθείτε αν ανήκει στη γλώσσα.

γ) Είναι γραμματική LL(1) κ' γιατί;

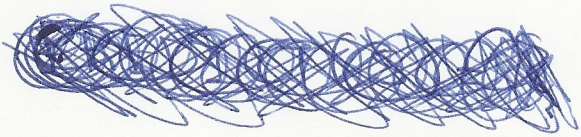
Αν όχι παρουσιάστε μία ισοδύναμη που είναι.

a) σωστά εφωτισμένα ζεύγη παρενθέσεων κ' αγκυλών.

δ) Όχι γιατί έχει απίστευτη άμεση αναδρομή.

$$\text{FIRST}(S) = \{ C, [, \epsilon \}$$

$$\text{FOLLOW}(S) = \{ (,), [,], \$ \}$$



1. $\langle S \rangle ::= \langle S \rangle \langle S \rangle$

2. $\langle S \rangle ::= \langle S \rangle [\langle S \rangle]$

3. $S ::= \epsilon$

$$\text{PREDICT}(1) = \text{scribble} \{ C, [\}$$

$$\text{PREDICT}(2) = \{ C, [\}$$

$$\text{PREDICT}(3) = \text{FOLLOW}(S)$$

Δεν ζήτησε ποιοι κανόνες να κάνει predict.

	C)	[]	
S	1,2,3	3	1,2,3	3	

I Γραμμή LL(1)

(αόριστη επιβεβαίωση είναι προφανές)

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

$$\langle S \rangle ::= [\langle S \rangle] \langle S \rangle$$

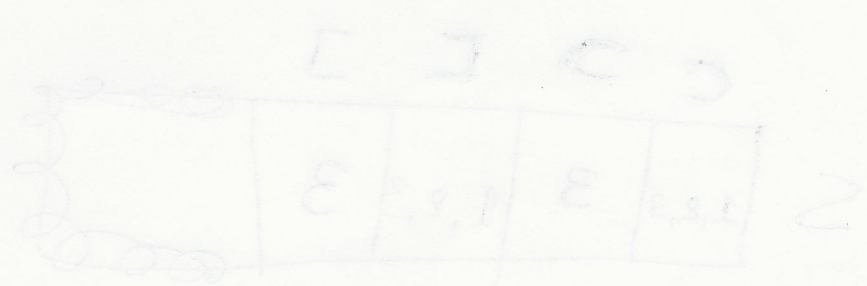
$$\langle S \rangle ::= \epsilon$$

1. $\langle S \rangle ::= (\langle S \rangle) \langle S \rangle$
2. $\langle S \rangle ::= [\langle S \rangle] \langle S \rangle$

3. $\langle S \rangle ::= \epsilon$
PREDICT(1) = {C, E}

PREDICT(2) = {C, E}

PREDICT(3) = FOLLOW(2)



① Αρχές γλωσσών προγραμματισμού

κ' μεταφραστών

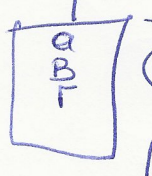
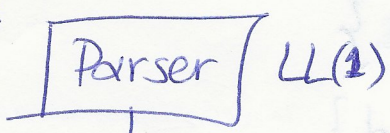
• $PREDICT(A \rightarrow \alpha) \equiv (FIRST(\alpha) - \{\epsilon\}) \cup$

$\cup (\text{αν } \alpha \Rightarrow^* \epsilon \text{ τότε FOLLOW}(A) \text{ αλλιώς } \emptyset)$

• $FIRST(\alpha) \equiv \{ \alpha : \Rightarrow^* \alpha \beta \} \cup (\text{αν } \alpha \Rightarrow^* \epsilon \text{ τότε } \{ \epsilon \} \text{ αλλιώς } \emptyset)$
 (Σύνολο όρων των πιθανών συμβόλων που μπορεί να ξεκινήσει μια συμβολοσειρά)

• $FOLLOW(A) \equiv \{ \alpha : S \Rightarrow^+ \alpha A \beta \} \cup (A \text{ αν } S \Rightarrow^* \alpha A \text{ τότε } \{ \epsilon \} \text{ αλλιώς } \emptyset)$

εμβόλοεισόδου
 →
 εισόδου



Στοιβά συντακτικής ανάλυσης (κανόνες γραμματικής)

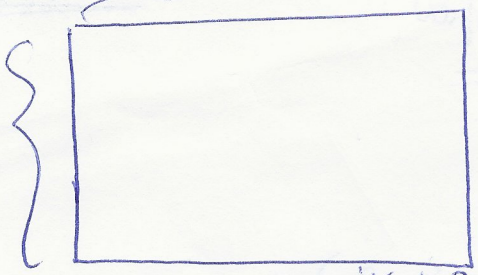
1) Διαβάζει το πάνω πάνω εμβόλο της στοιβάς.
 Αν είναι τελευταίο προχωράει παρακάτω.

2) Αν όχι :

Κάνει predict

μόνο μια επιλογή κάθε φορά : LL(1)

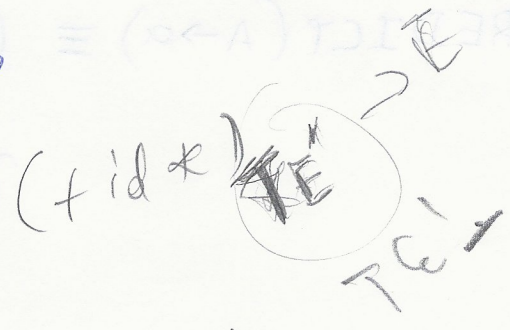
MT



→ ζητείται από predict sets

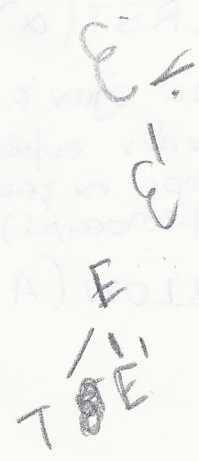
ΕΚΘΕΤΑ 2.19 του βιβλίου

Π.Χ. Γραμματική



- $E \rightarrow TE'$
- $E' \rightarrow \epsilon$
- $E' \rightarrow +TE'$
- $T \rightarrow FT'$
- $T' \rightarrow *FT'$

- $T' \rightarrow \epsilon$
- $F \rightarrow (E)$
- $F \rightarrow id$



ΣΥΝΟΛΑ FIRST()

- $FIRST(E) = \{ (, id \}$
- $FIRST(E') = \{ +, \epsilon \}$
- $FIRST(T) = \{ (, id \}$
- $FIRST(T') = \{ *, \epsilon \}$
- $FIRST(F) = \{ (, id \}$

* 'Από $F \rightarrow E$ και άλλο
 ημείρωσε στο
 πρώτο σύμβολο
 του T' .

ΣΥΝΟΛΑ FOLLOW()

↓
 ποιο μπορεί να είναι το πρώτο σύμβολο που ακολουθεί μια παραγωγή A.

$$FOLLOW(E) = \{ \$, \$,) \}$$

$$FOLLOW(E') = FOLLOW(E) = \{ \$, \$,) \}$$

$$FOLLOW(T) = FIRST(E') - \{ \epsilon \} \cup FOLLOW(E) = \{ + \} \cup \{ \$, \$,) \}$$

$$FOLLOW(T') = FOLLOW(T) = \dots$$

$$FOLLOW(F) = FIRST(T') - \{ \epsilon \} \cup FOLLOW(T) = \{ *, +,) \} \cup \{ \$, \$ \}$$

→ προσθέτω πάντα το EOF ή το ε

στην αρχή
 κοιτάω τα δεξιά μέλη των παραγωγών.
 δεξιά του E υπάρχει μόνο το ')'

→ δεν ακολουθεί κάτι το E' ή αν ακολουθεί ότι ακολουθεί το E (1ος κανόνας)

άρα: ')'

ΣΥΝΟΛΑ PREDICT()

Αν είναι γένε είναι LL(1)

→ αν έχει κενό κοιτάω το E' αλλιώς FIRST(T), αν όλο E κοιτάω το FOLLOW(E)

$$PREDICT(E \rightarrow TE') = \{ C, id \}$$

$$PREDICT(E' \rightarrow \epsilon) = FOLLOW(E')$$

$$PREDICT(E' \rightarrow +TE') = \{ + \}$$

$$PREDICT(T \rightarrow FT') = \{ (, id \}$$

$$PREDICT(T' \rightarrow *)$$

Ε' Μεταφραστών.

ΘΕΜΑ 2, Άσκηση Σεντεμπερίου 2011

- α) float * [n];
- β) float (* b) [n];
- γ) double (* c [n]) ();
- δ) double (* d ()) [n];

Να περιγράψετε το είδος των δεδομένων που επιτρέπει
 η ως παραπάνω συνήχηση C

- α) Array από η pointers σε float
- β) Pointer σε array που αποτελείται από η floats
- γ) Array από η pointers σε συναρτήσεις που
 επιστρέφουν double.
- δ) Function που επιστρέφει έναν pointer σε
 array που αποτελείται από η doubles.

④ FIRST κτλ των μη τελεατικών πόρων.

① ④ EBNF → BNF % T-PROG

② BNF → BISON (y)
Corομίζω token T- .
T-PROG = 271

③ flex λέει στο bison
ότι βρήκε ένα keyword "program".

(επικοινωνία με κρέπαιους τελικά))

program?

```
{ return T-PROG; }
```

<EOF>

}

Αρχές Γλωσσών Προγραμματισμού

κ' Μεταφραστών

①

```
program MAIN;  
  var Z: integer;  
  procedure p(x: integer);  
    begin  
      <BODY>  
    end;  
  
  BEGIN  
    Z := 1;  
    p(Z);  
    write(Z);  
  
  END
```

Γράψτε μέχρι 2 αντίστοιχες για το <BODY> τέτοιες ώστε να χρησιμοποιούνται διαφορετικές τιμές για το Z, αν η μεταβίβαση παραμέτρων γίνεται με:

- κλήση με τιμή
- " " αναφορά
- " " τιμή-αποτέλεσμα

Ποιες είναι οι τιμές που θα χρησιμοποιούνται σε κάθε περίπτωση; Εξηγήστε συνοπτικά τη διαδικασία υπολογισμού των τιμών.

2 εντολές στο <BODY>:

```
1) X := X + 1
2) Z := Z + 2
```

Τι θα τρωθεί;

(a)

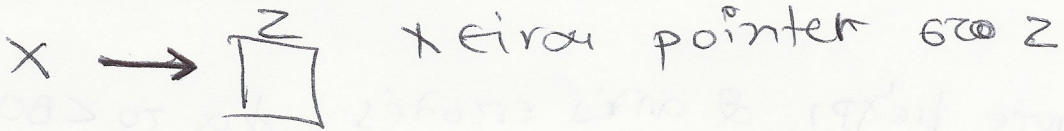
value



Τρωάται: 3

(b)

Reference



1) X = 2 Z = 2

2) Z = 4

Τρωάται: 4

c) value-result

$x \rightarrow \boxed{1}$

1) $x = 2$

2) $z = 3$

Στο τέλος αυτό που λέγεται στο z είναι το 2, γιατί x είναι η τιμή που παίρνεται.

Τυπώματα: 2

2

```
int x, y, z;
```

```
void main()
```

```
{ x = 15; y = 10;
```

```
z = A(x, y);
```

```
print(z);
```

```
}
```

```
int A(int u, int v)
```

```
{ if (v == 0) return u;
```

```
else return A(v, u % v);
```

```
}
```


a) Ποια είναι η λειτουργία ως function A;

b) Ποια είναι τα περιβάλλοντα αναφοράς (ζονικά, μη-ζονικά, καθαριστά) όλων των σημείων του προγράμματος;

c) Παρουσιάστε τη στοίβα εκτέλεσης (run-time stack) των εγγραφών ενεργοποίησης (Activation Records - AR) με ως τρέχουσες τιμές των ζονικών μεταβλητών για κάθε AR, μόλις έχει γίνει η τελευταία κλήση της A.

d) Ποια τιμή θα τυπωθεί στο main για το z;

a) $u = 15$, $v = 10$

(Αλγόριθμος Ευκλείδειου)
για υπολογισμό ΜΚΔ δύο ακεραίων

$$\begin{array}{r} 10 \qquad 5 \\ \hline \end{array}$$

$$\begin{array}{r} 5 \qquad 0 \\ \hline \end{array}$$

$\rightarrow \text{ΜΚΔ} = 5$

$X=15$ $Y=10$
$U=15$ $V=10$
$U=10$ $V=5$
$U=5$ $V=0$

A.R. του MAIN

A.R. του A (1^η κλήση)

A.R. του A (2^η κλήση)

A.R. του A (3^η κλήση)

d) $Z=5$

Προγραμματισμού κ' Μεταφραστών

```

3 program MAIN;
  var i: Integer;
      a: real;

  function S(i, Lo, Hi: integer; term: real): real;
    var Temp: real;
  begin
    Temp := 0;
    for i := Lo to Hi do Temp := Temp + term;
    S := Temp;
  end;

BEGIN
  i := 1;
  a := S(i, 1, 100, 1/i);
  write(a)
END.
    
```

Τι υπολογίζει εδώ η συνάρτηση S κ' τινώνεται στο
 γέγραφο με την εντολή write(a), πως η συνάρτηση ορίστηκε.
 a) call by value
 b) " " name

Δώστε τον αντιστοιχισμό των (α) με τις μεταβλητές
στα (β) με μαθηματικό τρόπο.

α) Περιβάλλοντα

Για την S:

ΠΠΑ: $i, Lo, Hi, term, Temp, S$ (σαν μια τοπική μεταβλητή
αφού επιστρέφει τιμή)

ΜΥΠΠΑ?

Κ'ΚΠΑ } : a, S (ορίζεται αντίστοιχως πρόγραμμα)

(το i όχι γιατί ορίζεται ε' στην S)

Για το MAIN:

ΠΠΑ?

Κ'ΚΠΑ } : i, a, S

(α) call by value

$i = 1$ $Lo = 1$ $Hi = 100$ $term = 1$

$Temp = 0$

$i = 1$ to 100

$Temp = Temp + term$

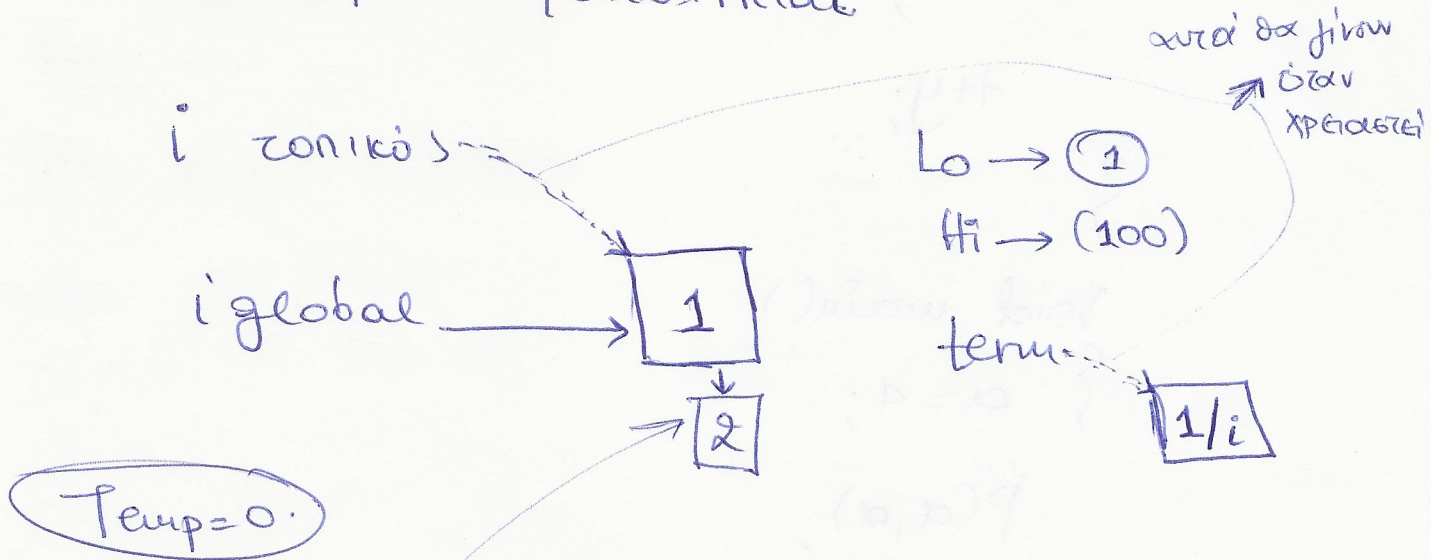
$S = Temp = 100$

$write(a) \rightarrow 100$

Case by name (κάθε φορά που χρησιμοποιούμε οι αριθμούς παράμετροι γίνεται σύνδεση)

(100 φορές θα γίνει υπολογισμός του $\frac{1}{i}$ term)

κάθε φορά θα μεταβάλλεται



for $i=1$ to 100 do $Temp = 0 + 1$

\vdots
 $i=2$

↓
 γίνεται πάλι υπολογισμός
 $Temp = 1 + \frac{1}{2}$

$i=3$

↓
 γίνεται γανά υπολογισμός
 $Temp = 1 + \frac{1}{2} + \frac{1}{3}$

$i=100 \rightarrow Temp = \sum_{i=1}^{100} \frac{1}{i}$

4

```
int a;  
void P (int x, int y)  
{  
    ++x;  
    ++y;  
}  
void main()  
{  
    a = 4;  
    P(a, a);  
    print(a);  
}
```

- (a) Περιβάλλοντα αναφοράς(?)
- (b) Παρουσιάστε τη στοίβα εκτέλεσης των εγγραφών ενεργούντων φορές έχοντας την κλήση της P.
- (c) (i) call by value.
(ii) call by reference
(iii) " " value-result

a) Περιβάλλοντα αναφοράς:

Για την P:

Για την main:

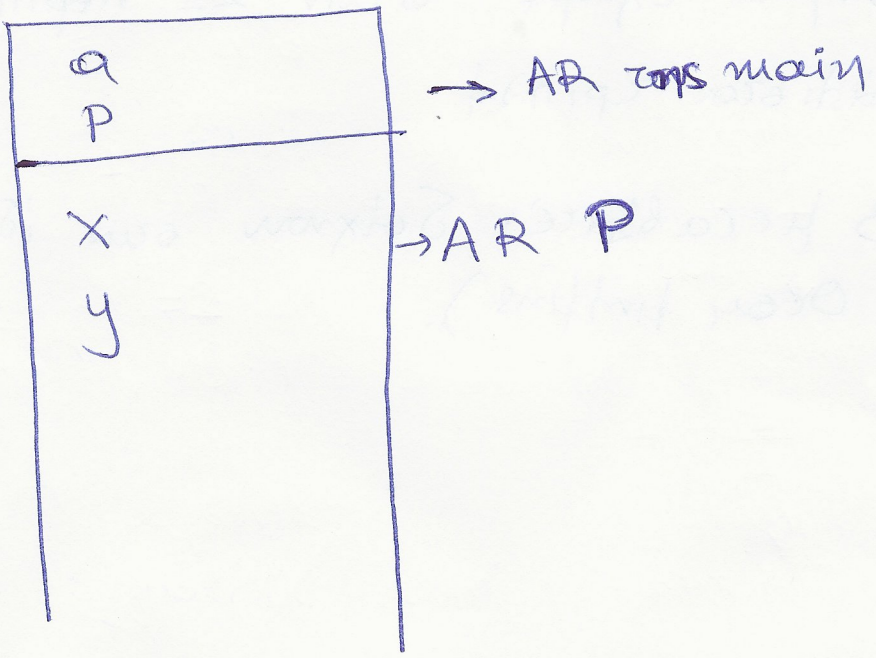
ΤΠΑ: x, y

ΤΠΑ } → α, P
 κ'εΠΑ }

{ ΜΤΠΑ }
 { κ'εΠΑ } : α, P

επεδύ έχουμε
 2 επίπεδα
 (δυναμική + κερύς πρόγραμμα)

(b)



(c)

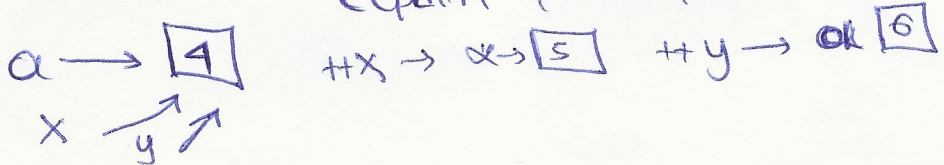
(i) call by value

Τυπώνεται: 4 (το a δεν έχει αλλάξει)

(ii) call by reference

Τυπώνεται: 6

Τα x, y γίνονται pointers στο α
 (στην ψευδωπία)



(11)

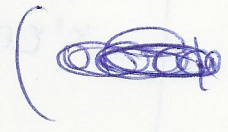
call by value - result

Writeln: 5

$a \rightarrow [4]$

$x \rightarrow [5]$

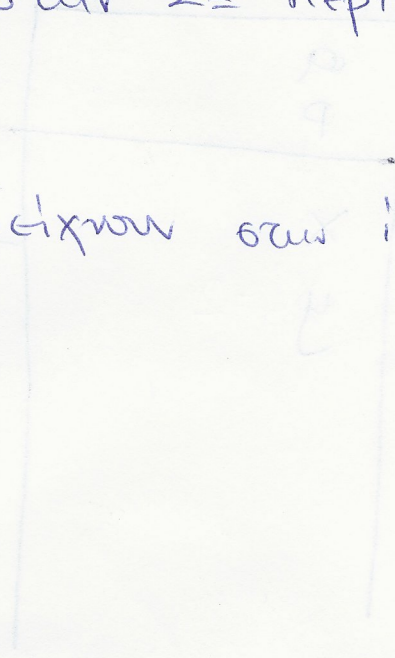
$y \rightarrow [5]$



Η τιμή της μεταβλητής
παράτηρου ανυπαρξεί
σαν παράταξη
μολις τελείωσει
η εκτέλεση

Ψευδωνυμία έχουμε σαν 2^η περίπτωση
κ' μαδιστα τριπλή

(3 μεταβλητές δείχνουν σαν ίδια
όσον αφορά)



(1) call by value

(2) call by reference

$a \rightarrow [4]$

$x \rightarrow [5]$