

ΑΡΧΕΣ ΠΡΟΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ & ΜΕΤΑΦΡΑΣΤΩΝ

1] Δίνεται μια BNF Γραμμή με τελεστές a, b

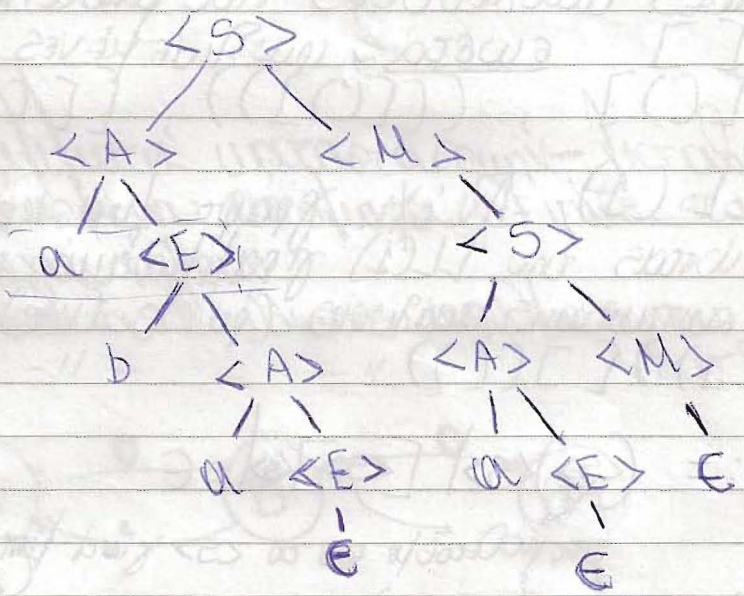
- $\langle S \rangle ::= \langle A \rangle \langle M \rangle$ (start symbol)
- $\langle M \rangle ::= \langle S \rangle \mid \epsilon$
- $\langle A \rangle ::= a \langle E \rangle \mid b \langle A \rangle \langle A \rangle$
- $\langle B \rangle ::= b \langle E \rangle \mid a \langle B \rangle \langle B \rangle$
- $\langle E \rangle ::= a \langle B \rangle \mid b \langle A \rangle \mid \epsilon$

- a) Parse tree για το $abaa$
- b) Είναι δισημίτηρη γραμματική (ambiguous); αποδείξε με βάση το α)
- c) Περιγράψε τη γλώσσα που παράγει η γραμματική.

ΑΠΑΝΤΙ

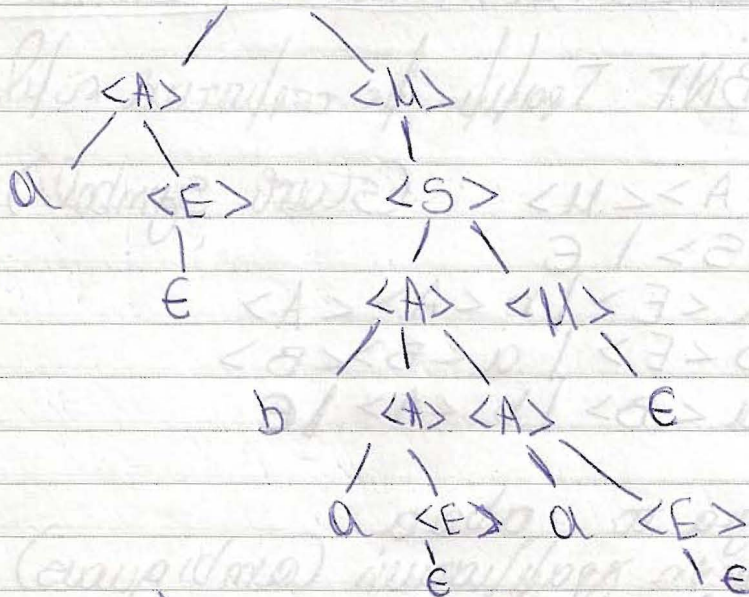
a) Ξεκινάμε από το start symbol, το οποίο αποτελεί την υψηλότερη γραμματική κατηγορία.

Ο γρίφος είναι το $\langle E \rangle$ να γίνει a ή b ή ϵ ή a ή b ή ϵ



Σε κάθε κόμβο, τοποθετούμε ένα τεμάχιο της ερωτήσεως της BNF που μας "αφήνει", ανάλογα με το string που έχουμε να κάνουμε parsing. Αναζητούμε μέχρι να υποστηρίξουμε σε κάποιο τελεστικό σύμβολο που βγαίνει στο string ή σε ϵ .

8)



Είναι σημαντικό να μην πιστέψουμε να έχουμε να
 είναι parsing tree που να παράγει το επιθυμητό string

γ) Κάθε string που παράγεται α αμφίσημο των α
 είναι παραγόμενο από τον αμφίσημο των β.

12) α) Γράψτε στο BNF γραμμάρει η οποία παράγει
 όλες τις δυνατές ακολουθίες που αποτελούνται από
 () ή/και [] σωστά κωθισμένες.
 π.χ. () , [()] , (([()])) , [[()]]

β) Είναι LL(1); Αν όχι, γράψτε μια ισοδύναμη.

γ) Χρησιμοποιώντας το LL(1) γραμμάρει, παραστήστε
 το δέντρο αναζήτησης αιώλων (parse tree) για το:
 ([[([()])]] (())

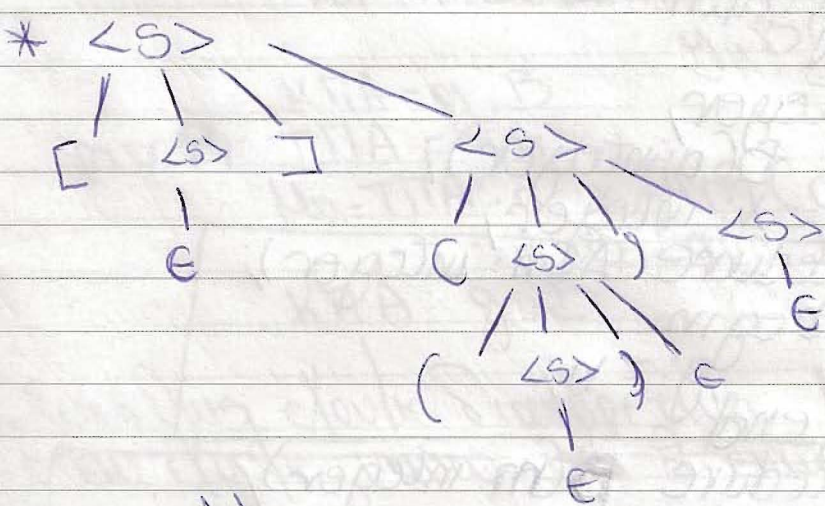
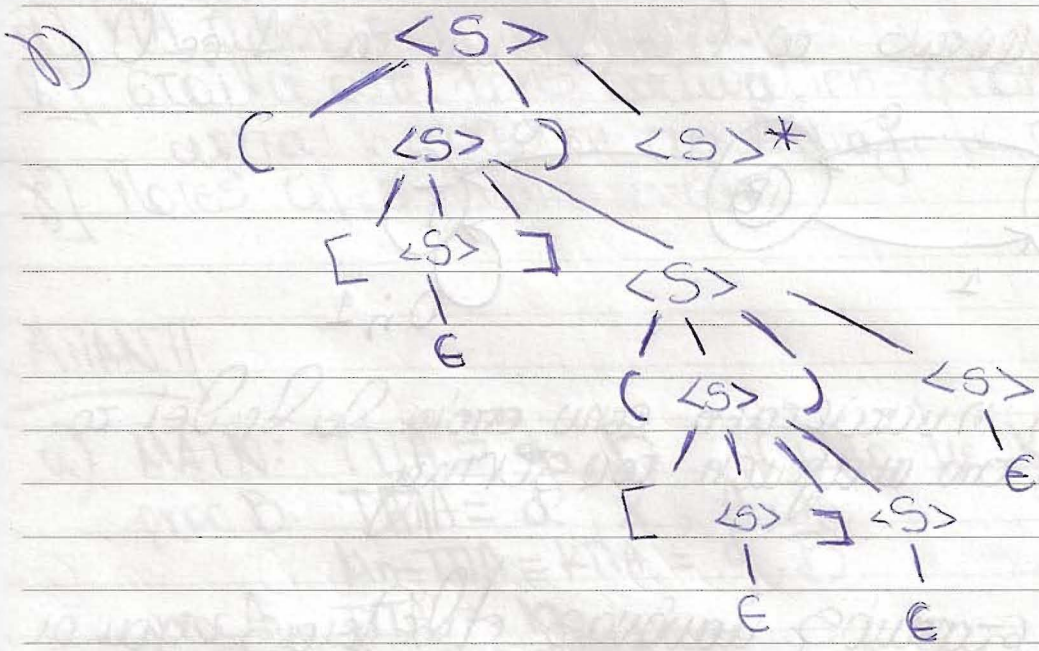
ΑΠΑΝΤΗ

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

τοποθετούμε ως το <S> για να μπορεί να αντικαταστήσει ε

β) ~~Είναι~~ Είναι LL(1). Δεν έχει αμφίσημη
 ακολουθία ($\langle S \rangle ::= \langle S \rangle (\langle S \rangle) \mid \langle S \rangle [\langle S \rangle] \mid \epsilon$)
 ούτε πάνω από 2 κωθισ το ε. Όσο 2 κωθισ που
 να έχουν το πρώτο σύμβολο ίδιο.

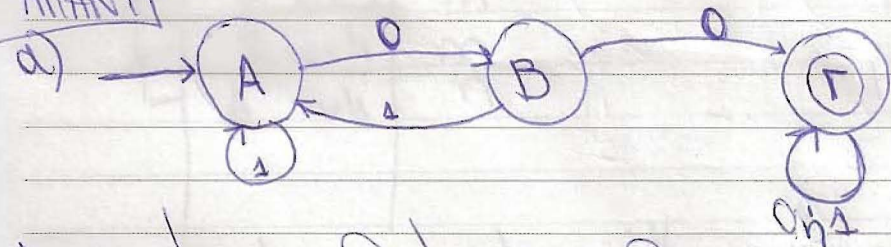
$\langle S \rangle ::= () \mid [] \mid (\langle S \rangle) \mid [\langle S \rangle] \mid \langle S \rangle \langle S \rangle$



13] Διαγράμματα μετασχηματισμού-περατοποίηση για τα
2 πεπερατά αυτόματα (finite-state automata)
που αναγνωρίζουν:

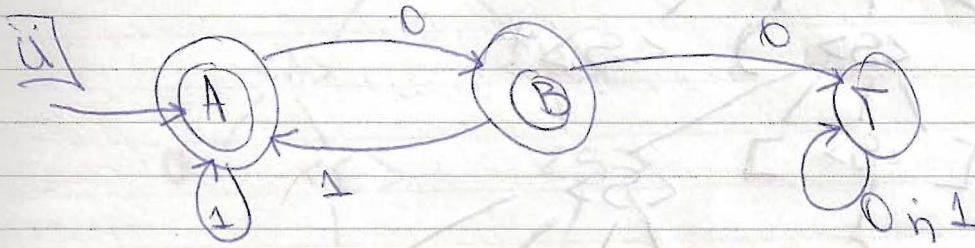
a) Όρους του διωνύμου αρθρού που περιέχουν το 00
b) -//- -//- που δεν περιέχουν 00

ΑΠΑΝΤΗ



Αναρωτιέμαι η επιδιωκόμενη είσοδος να παρασχεθεί είναι μια λέξη
Από το γεγονός για να μην είναι σαν τελική κατάσταση, ότι και
αν δώσει δεν υπάρχει σαν είσοδος είναι αποδεκτό.

Q) 1] Automata language to 0 or 1 and to 1 or 0.



Das endliche Automaten sind ein Modell der Sprache zu beschreiben und auch für die String.

4] Texter a standard language of letters and of which each be taken.

program MAIN

var g: integer;

procedure B(a: integer);

var x: integer;

procedure A(n: integer);

begin

g := n;

end

procedure B(m: integer);

begin

write(x);

x := x / 2;

if x >= 1 then R(m+1);

else A(m);

end;

begin

x := a * a;

R(1);

end;

BEGIN

B(3);

write(g);

END.

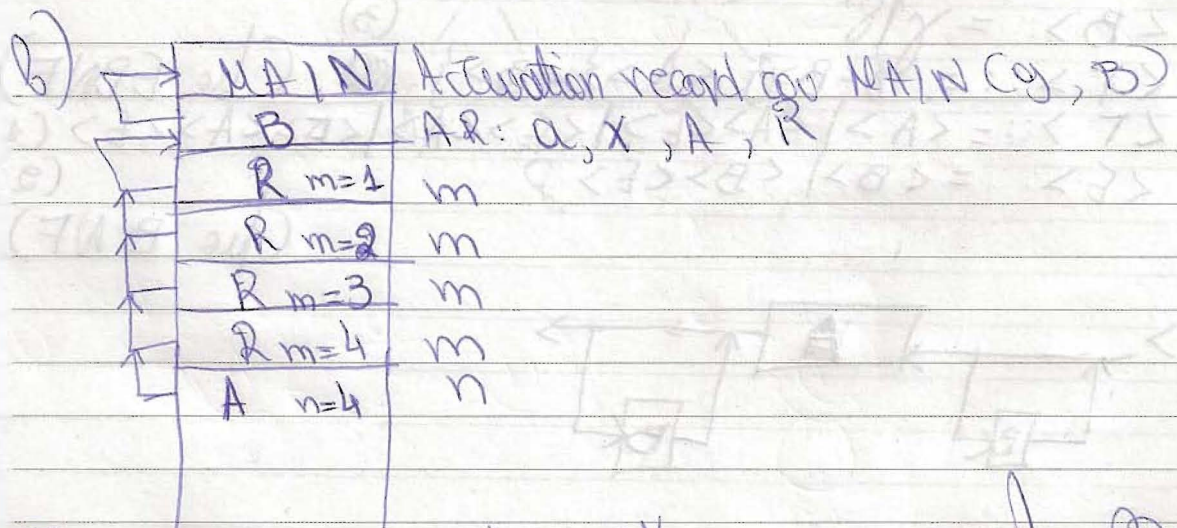
- a) TPA είναι των επαναλήψεων
- b) στοίβα ενεργών (activation records) πορτίς
αρχίζει η ενεργεια της A (γιατί με τους 6 ενεργούς)
- γ) ποτες αυτές τωπίο ποταί;

ΑΠΑΝΤΗ

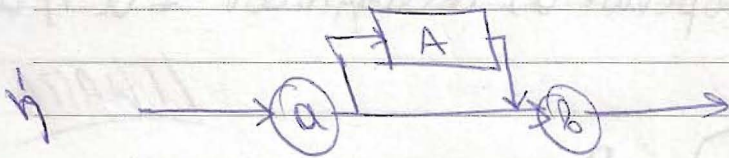
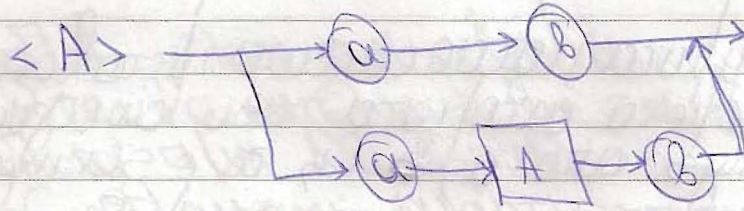
```

a) MAIN: TPA = g, B (είδιο με KPA)
proc B: TPA = a, x, A, R
      Mn-TPA = KPA = g, B
proc A: TPA = n
      Mn-TPA = α, x, A, R = α, x, A, R, (Cario to B)
      g, B (Cario MAIN).
      KPA = g, B
proc B: TPA = m
      Mn-TPA: g, x, A, R (Cario to B)
      g, B (Cario MAIN)
      KPA: g, B
    
```

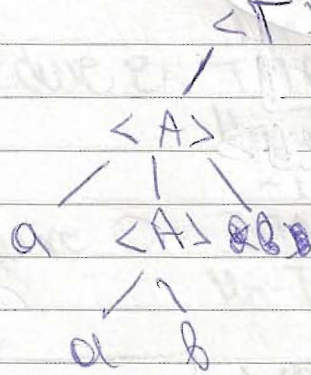
(Συνήθως έχουμε διαφορετικά Mn-TPA για KPA
όπου έχουμε πάνω από 2 ενεργούς επαναλήψεις).



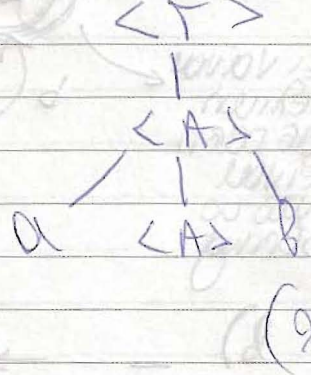
To activation record περιέχει τις τωπίες μεταβλητές
κάθε procedure που δείχνει τις ποτες άλλες procedures
που μπορεί να καλεί.



8)

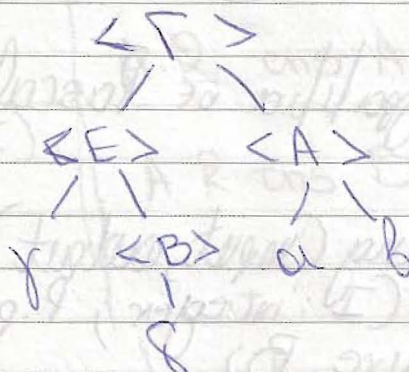


(nodes)



(nodes)

prob:



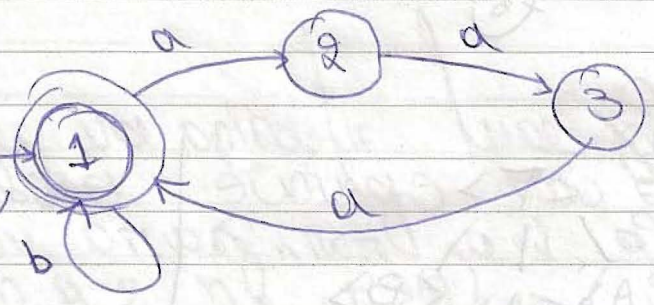
```

    procedure B
    begin
    if I > 1 then B(A, B); end
    procedure C; begin end;
    end
  
```

6) Θεωρήστε τα διηλεκτικά αυτομάτα παρακάτω με αρχικό κατάσταση για τα 2 πεπερασμένα αυτομάτα που αναφέρονται για όλα α) όλα τα strings από a, b (ή το ε string) με την ιδιότητα τα a να είναι σε οφθαλμούς του 3
 b) α ή b εναλλάξ/μετων a, ταυτό/ισοών 1 α ή b.

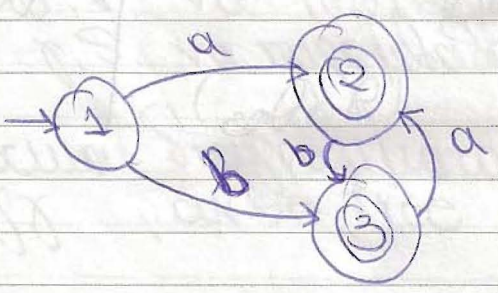
ATLANT

a)



πρέπει να έχει
 από την αρχή
 κάποιο string

b)



7) Δίνετε ένα πρόγραμμα σε Pascal, με στατικό υαύρα επέξευας

```

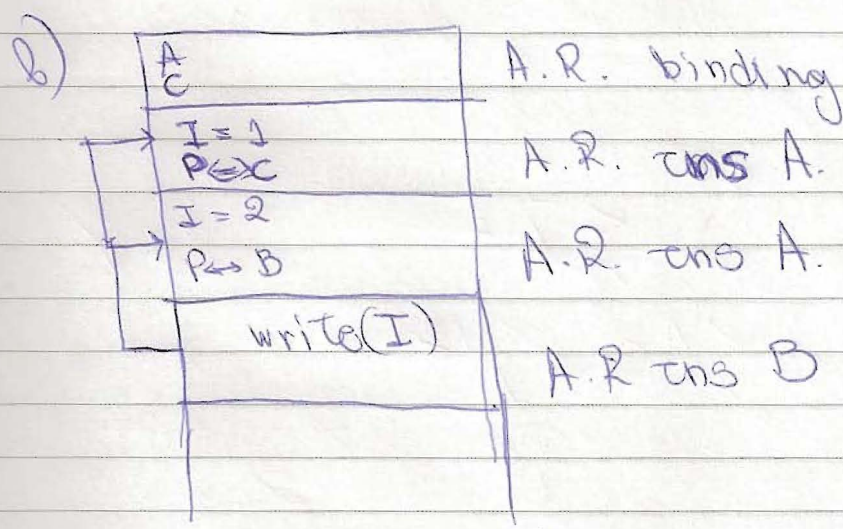
program binding (input, output);
procedure A(I: integer; P: procedure);
begin
  procedure B;
  begin
    write(I)
  end
end
  if I > I then P else A(2, B); end;
procedure C; begin end;
BEGIN
  A(1, C)
END
  
```


- α) T.A. όλων των εφελκυστών
- β) Στοιχεία εκτέλεσης με τα AR ποιος έχει κληθεί η Β
- γ) Ποια αμμή θα απαντήσει
- δ) Αναβλυστική ανάλυση στο γ).

ΑΠΑΝΤΗ

```

α) Program binding TPA, KPA: A, C
procedure A: TPA: I, P, B
             Nn-TPA = KPA = A, C
procedure B: TPA: —
             Nn-TPA: I, P, B, A, C
             KPA: A, C
procedure C: TPA: —
             Nn-TPA = KPA = A, C
    
```



Θα απαντήσει την αμμή 1 γιατί ισχύει ο σταθμός κληθείς εφελκυστός.