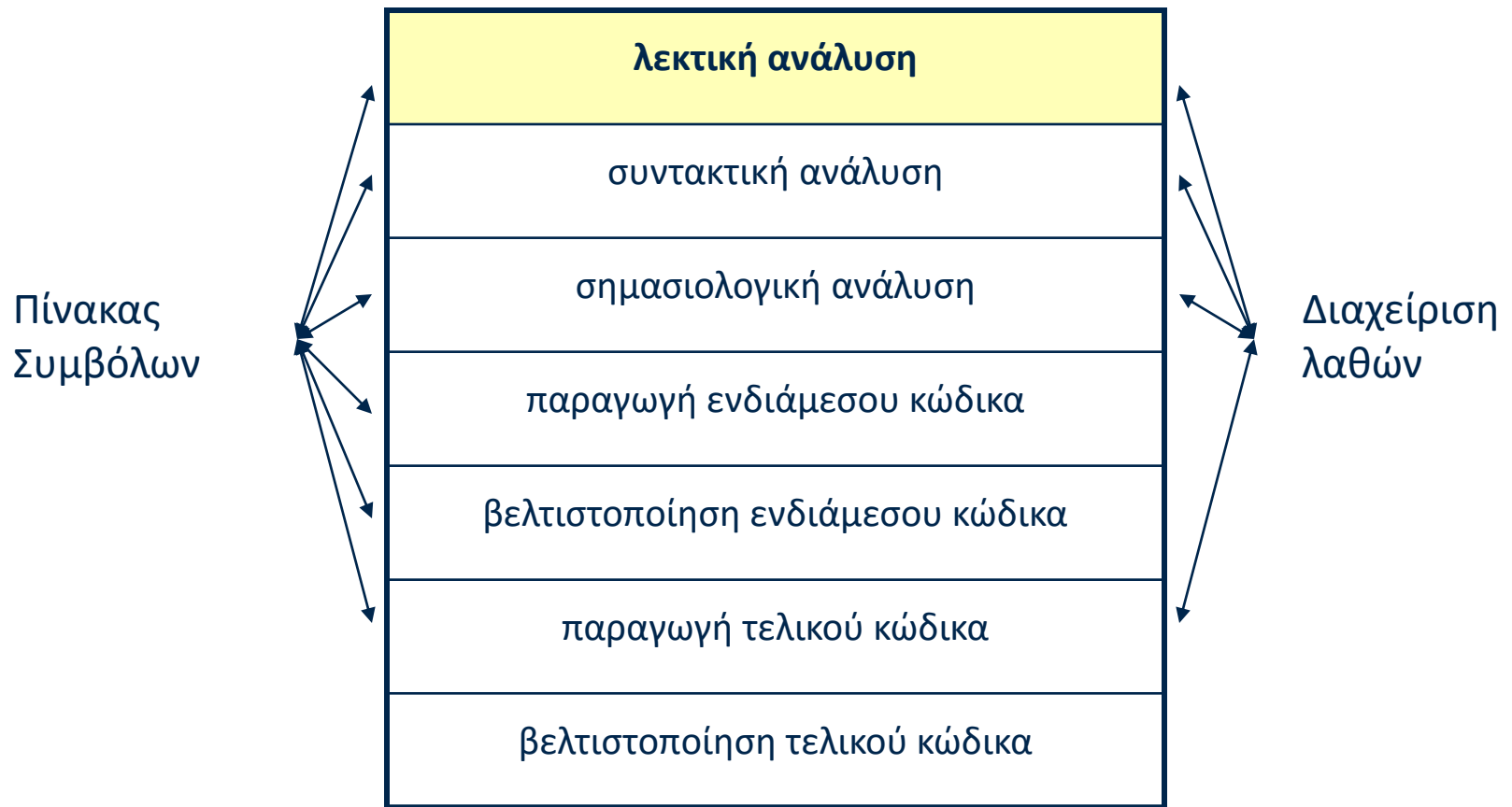

Λεκτικός Αναλυτής

Διαλέξεις στο μάθημα: Μεταφραστές
Γιώργος Μανής

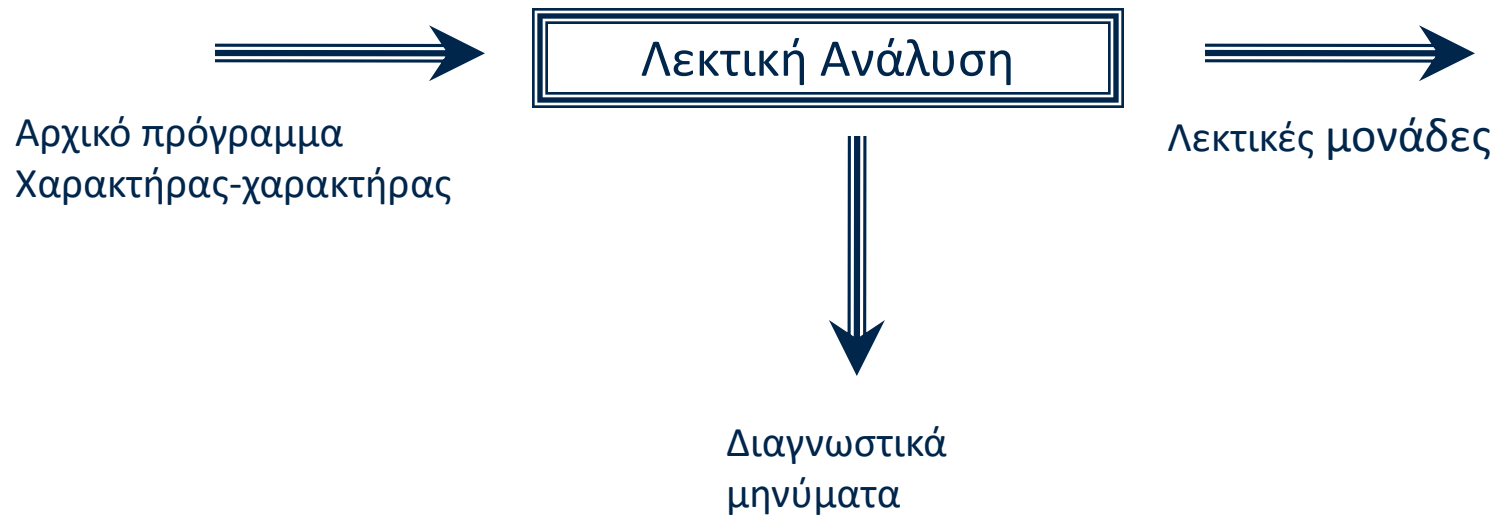
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA



Οι Φάσεις της Μεταγλώττισης



Λεκτική Ανάλυση



Διαπροσωπεία Λεκτικού Αναλυτή

- # Καλείται ως **συνάρτηση** από το συντακτικό αναλυτή
- # Διαβάζει **γράμμα-γράμμα** το πηγαίο πρόγραμμα
- # Κάθε φορά που καλείται επιστρέφει την **επόμενη λεκτική μονάδα**
- # Επιστρέφει στο συντακτικό αναλυτή
 - έναν **ακέραιο** που χαρακτηρίζει τη λεκτική μονάδα
 - τη **λεκτική μονάδα**

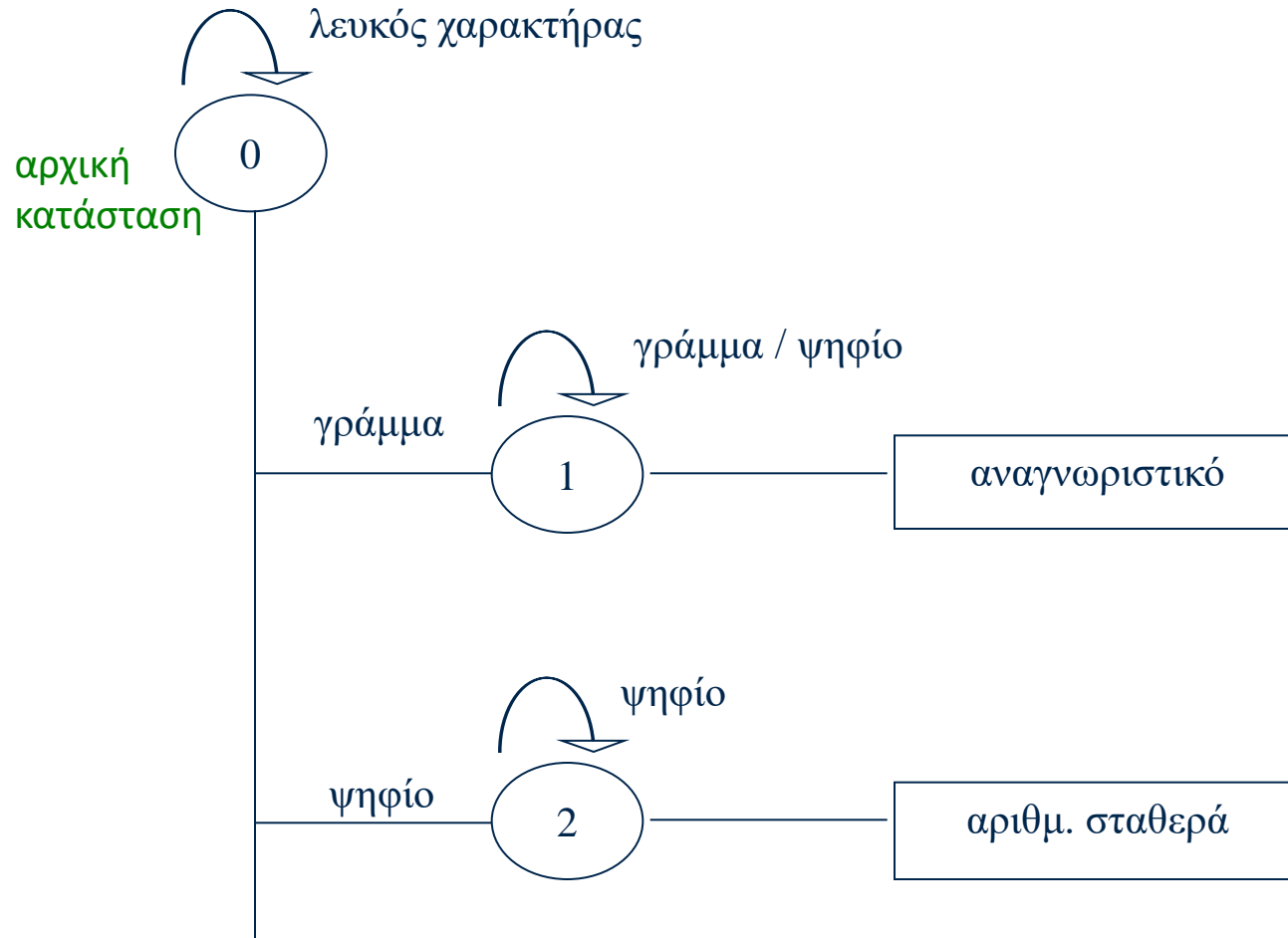
Εσωτερική Λειτουργία

- # Ο λεκτικός αναλυτής εσωτερικά λειτουργεί σαν ένα **αυτόματο καταστάσεων** το οποίο ξεκινά από μία αρχική κατάσταση, με την είσοδο κάθε χαρακτήρα αλλάζει κατάσταση έως ότου συναντήσει μία τελική κατάσταση
- # Το αυτόματο καταστάσεων αναγνωρίζει
 - δεσμευμένες λέξεις
 - πχ. if, for, while
 - σύμβολα της γλώσσας
 - π.χ. «+», «;», «=»
 - αναγνωριστικά και σταθερές
 - π.χ. counter, a12, 32768
 - λάθη
 - π.χ. μη επιτρεπτός χαρακτήρας, κλείσιμο σχολίων χωρίς να έχουν ανοίξει προηγουμένως

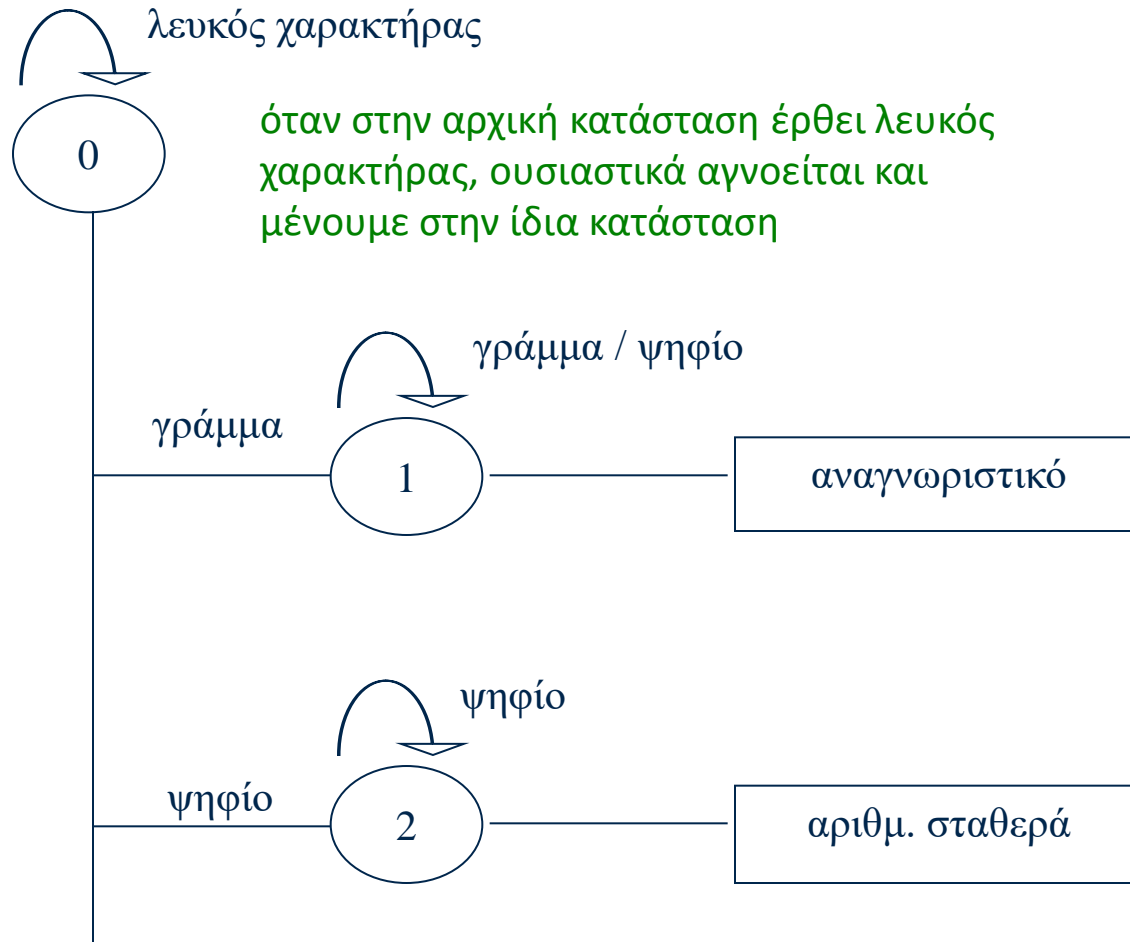
Παράδειγμα Αυτόματου

- ✦ Το αυτόματο του επόμενου παραδείγματος αναγνωρίζει:
 - αναγνωριστικά που ξεκινούν από γράμμα και αποτελούνται από γράμματα ή ψηφία
 - φυσικούς αριθμούς ως αριθμητικές σταθερές
 - τα αριθμητικά σύμβολα “+” “-” “*” “/”
 - τους σχεσιακούς τελεστές “<” “>” “=” “>=” “<=” “<>”
 - τα σύμβολα “:=” “(” “)” “,” “;”
 - σχόλια που περικλείονται σε άγκιστρα “{” “}” και αγνοούνται
-

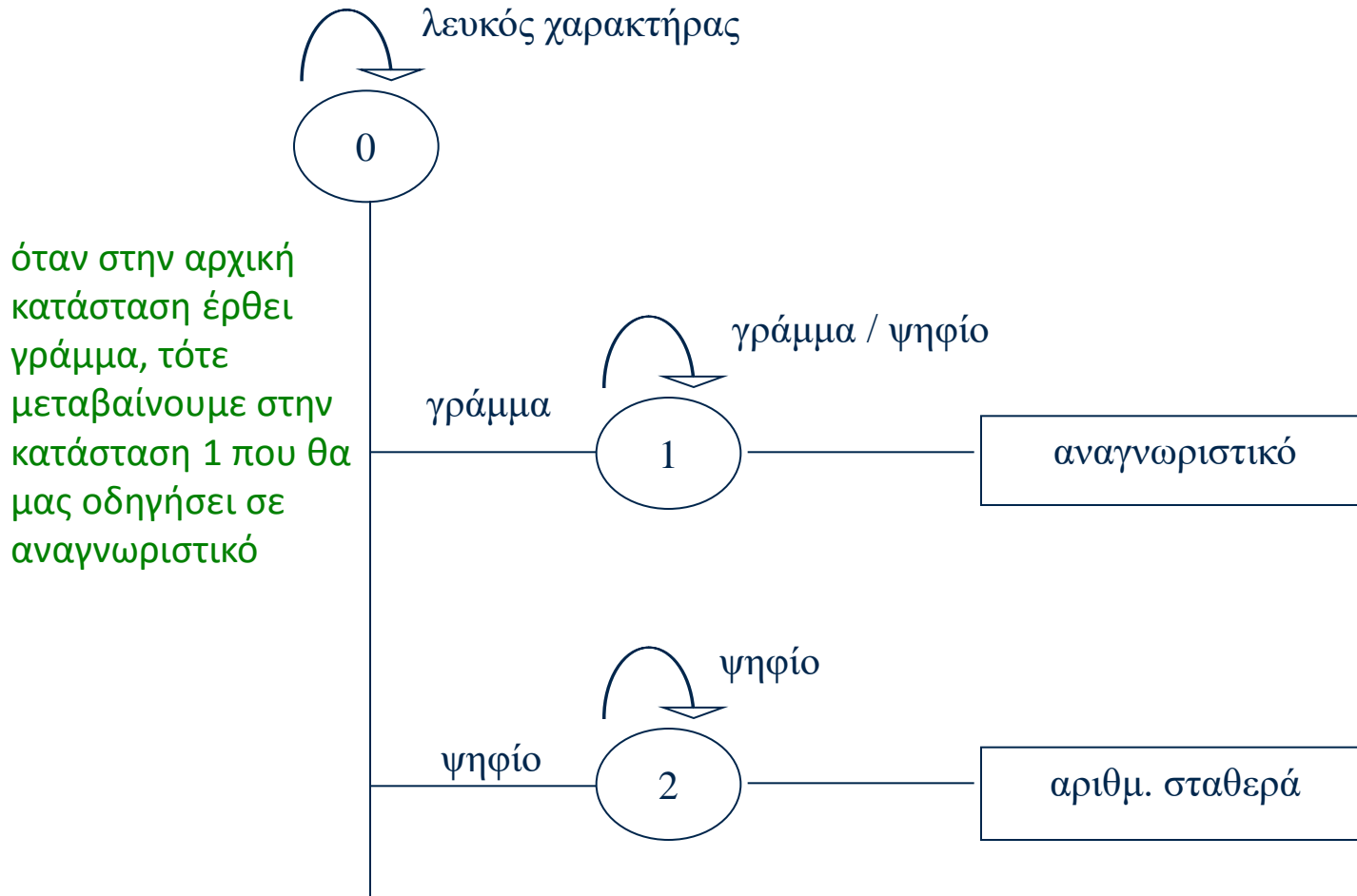
Αυτόματο



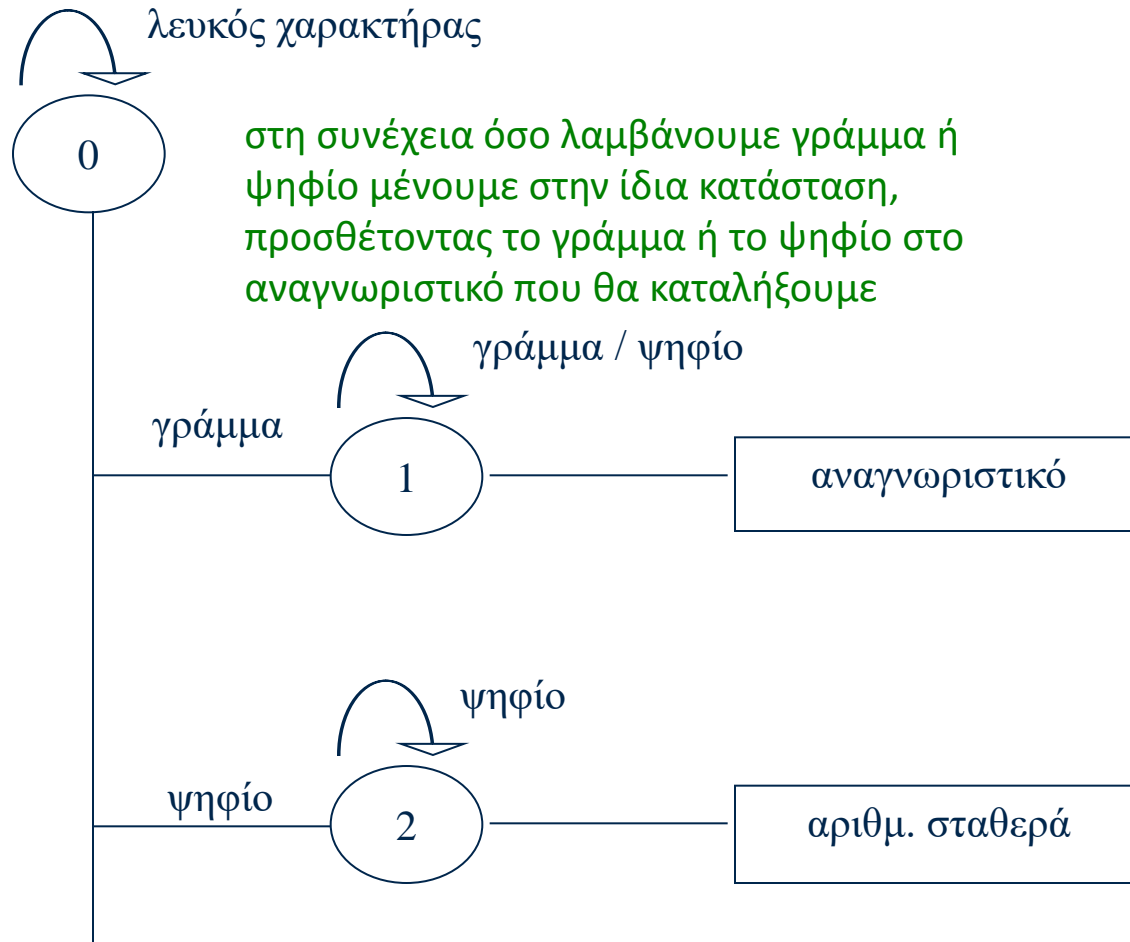
Αυτόματο



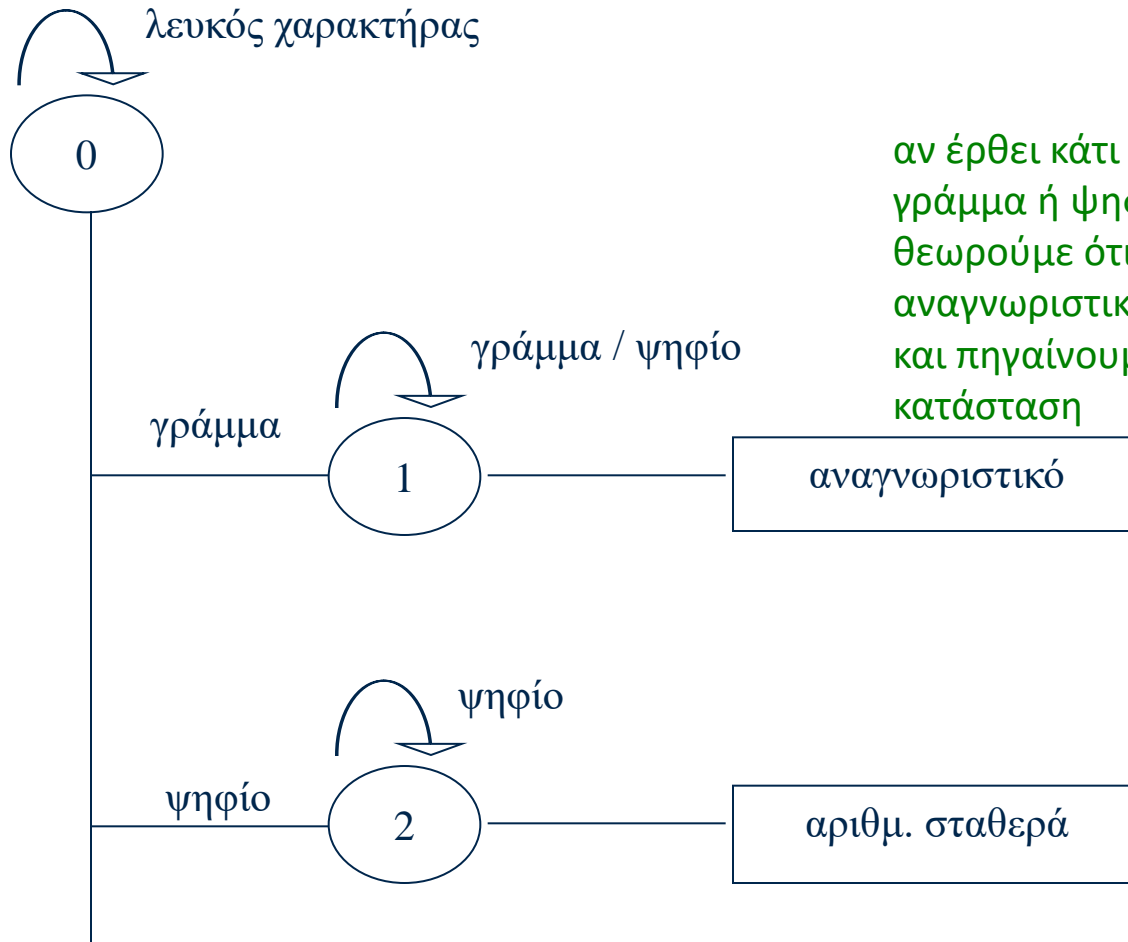
Αυτόματο



Αυτόματο

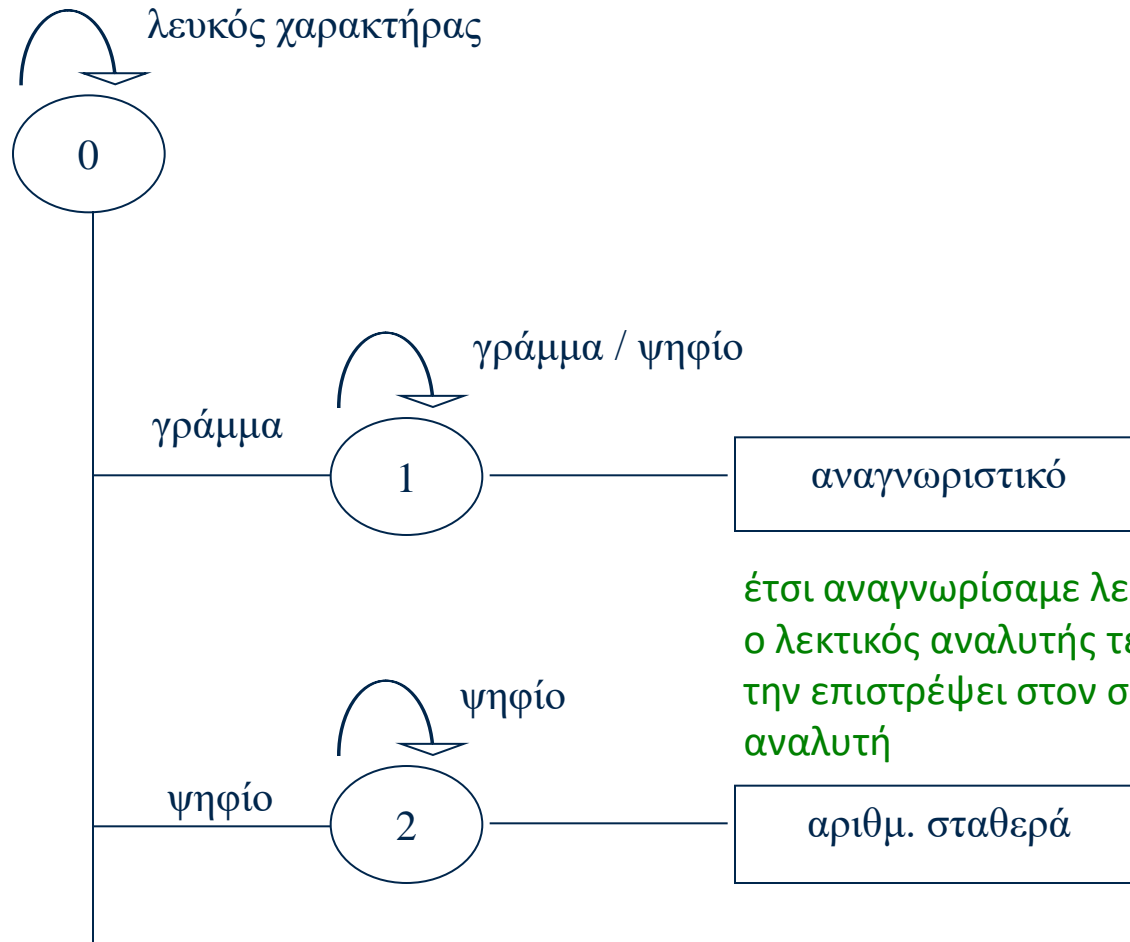


Αυτόματο



αν έρθει κάτι που δεν είναι
γράμμα ή ψηφίο τότε
θεωρούμε ότι το
αναγνωριστικό ολοκληρώθηκε
και πηγαίνουμε σε τελική
κατάσταση

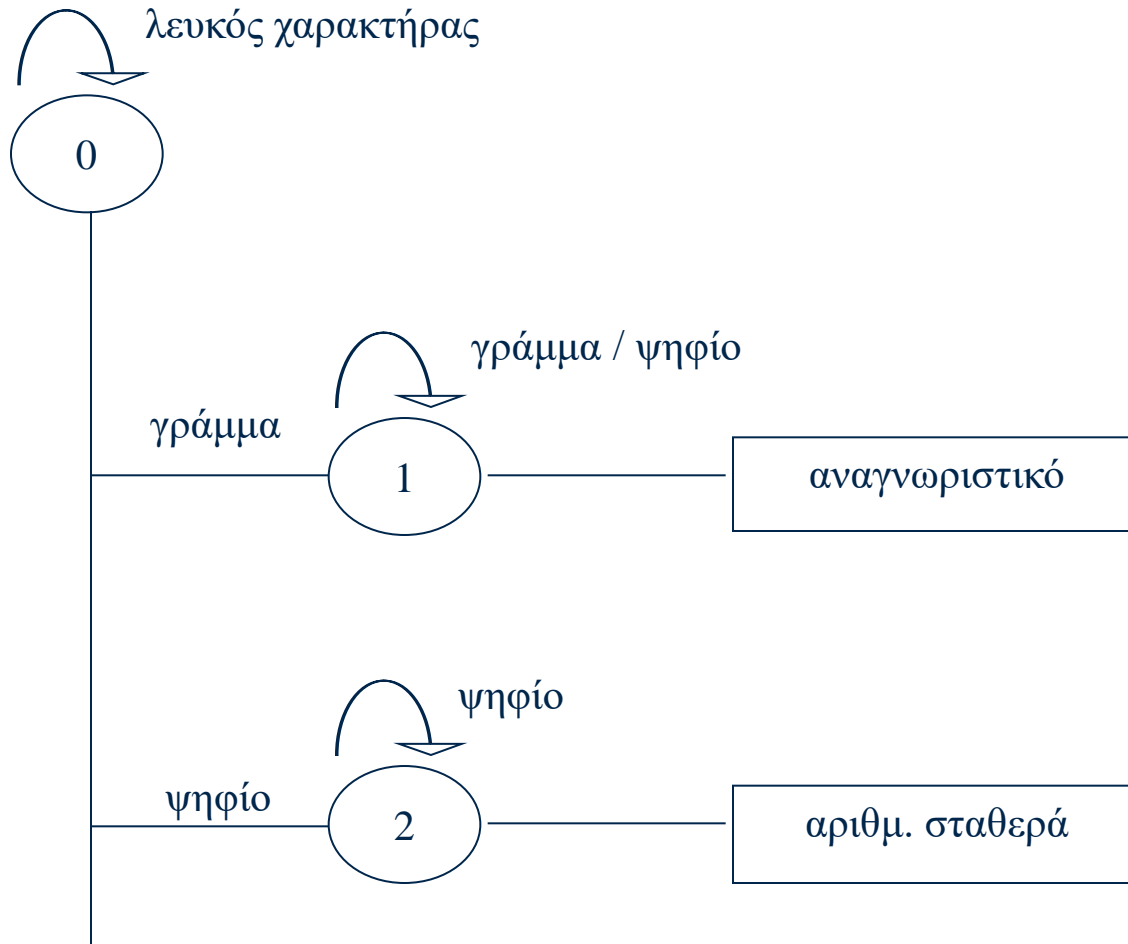
Αυτόματο



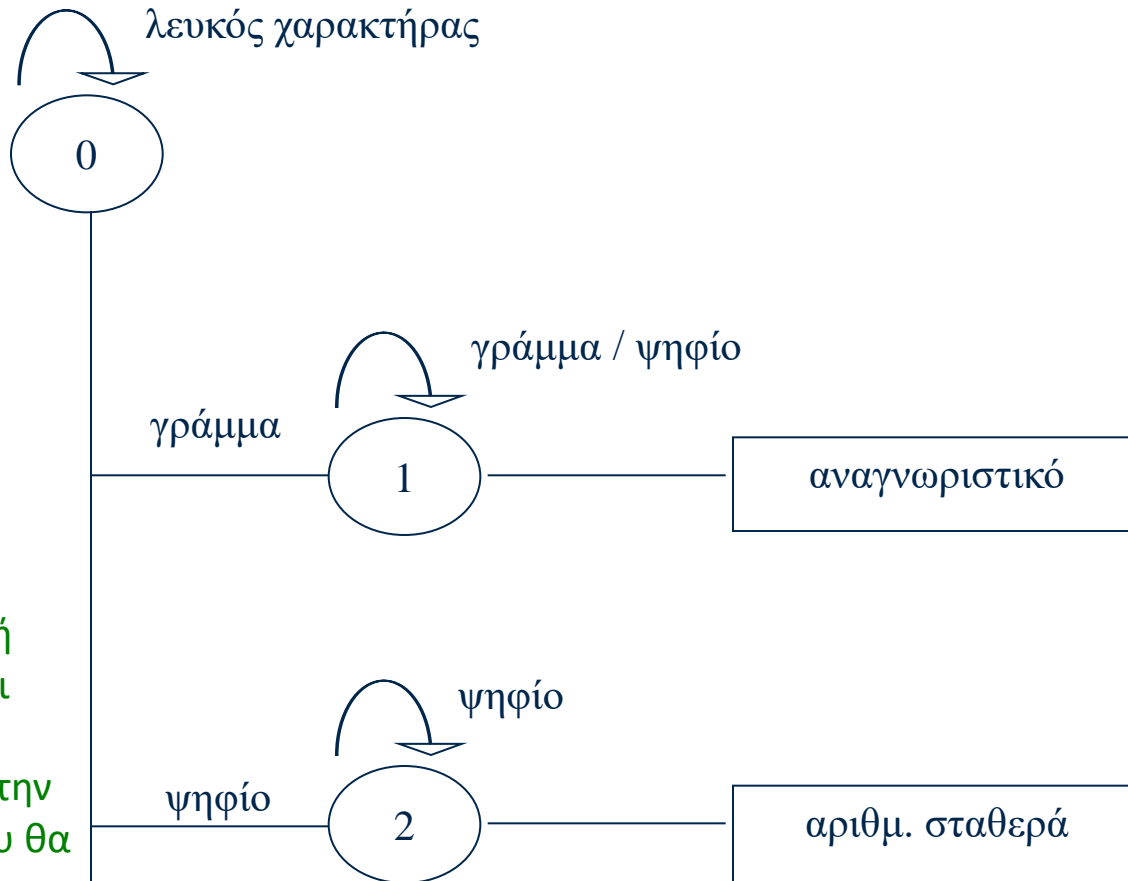
έτσι αναγνωρίσαμε λεκτική μονάδα και
ο λεκτικός αναλυτής τερματίζει αφού
την επιστρέψει στον συντακτικό
αναλυτή

Αυτόματο

όταν ξανακληθεί
ο λεκτικός
αναλυτής θα
ξεκινήσει πάλι
από την αρχική
κατάσταση,
αλλά θα
διαβάζει τους
χαρακτήρες από
εκεί που
σταμάτησε

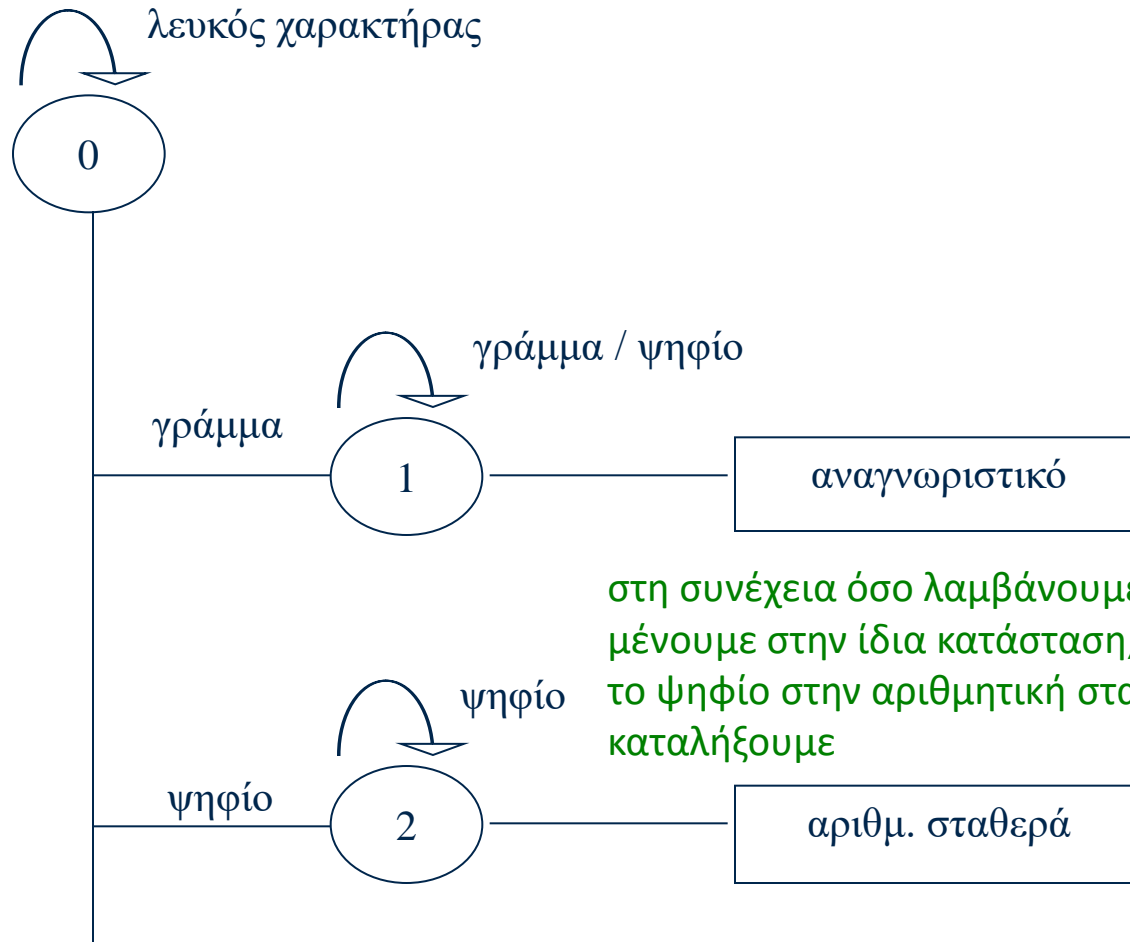


Αυτόματο



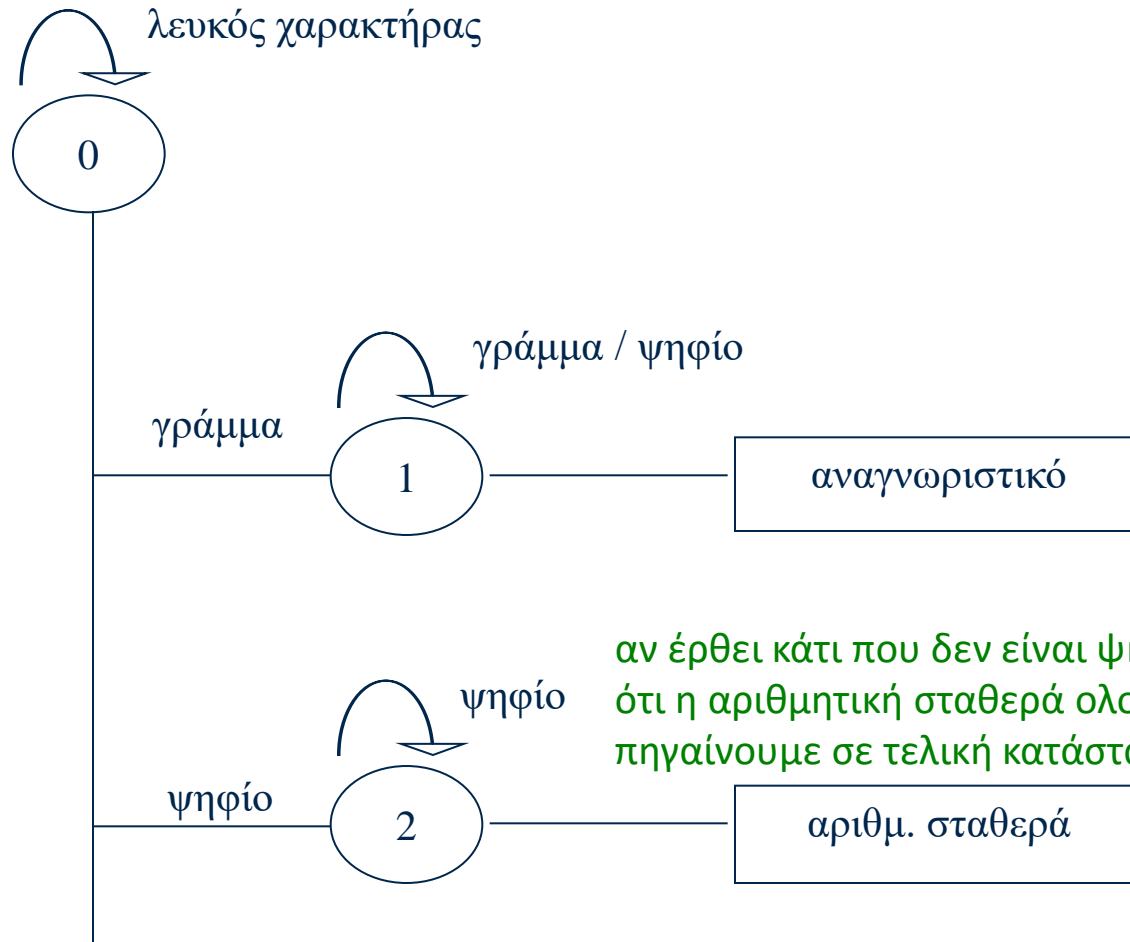
όταν στην αρχική κατάσταση έρθει ψηφίο, τότε μεταβαίνουμε στην κατάσταση 2 που θα μας οδηγήσει σε αριθμητική σταθερά

Αυτόματο



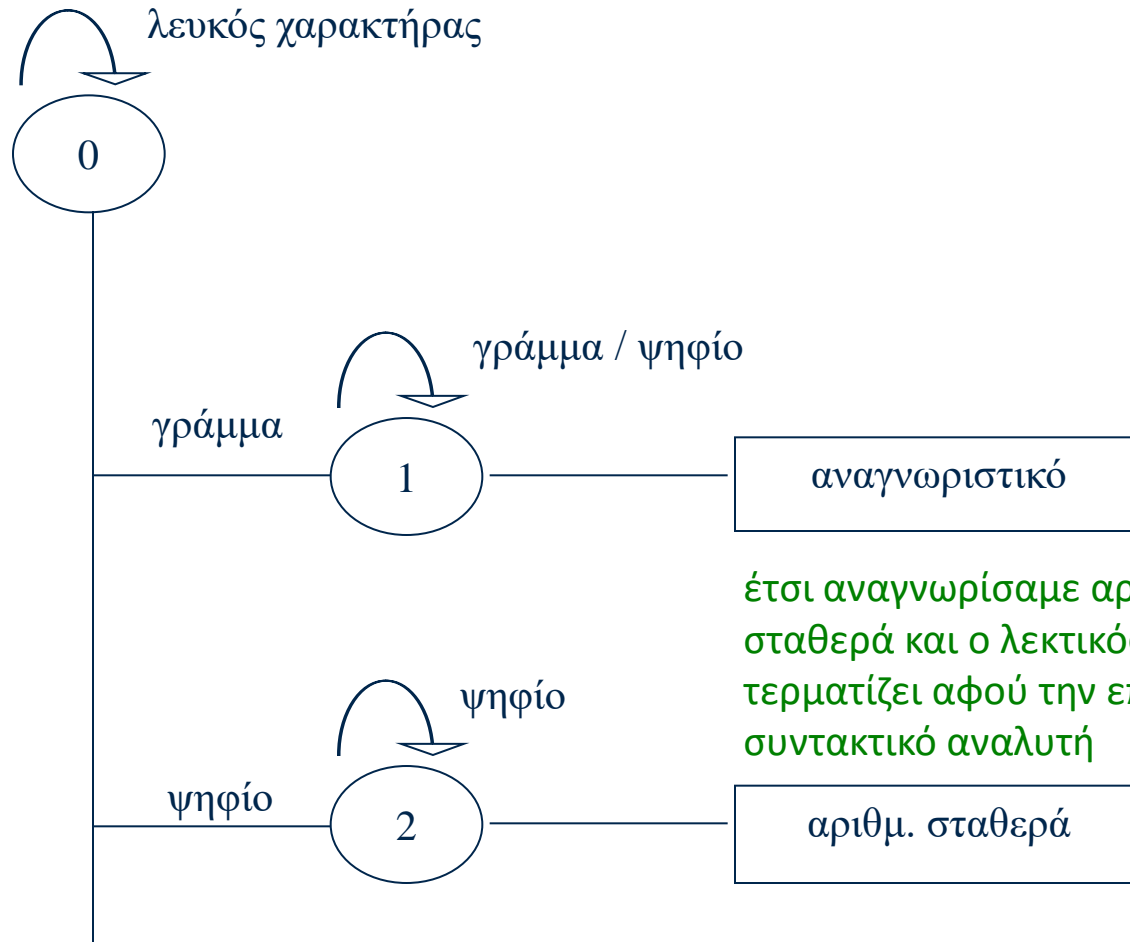
στη συνέχεια όσο λαμβάνουμε ψηφίο
μένουμε στην ίδια κατάσταση, προσθέτοντας
το ψηφίο στην αριθμητική σταθερά που θα
καταλήξουμε

Αυτόματο



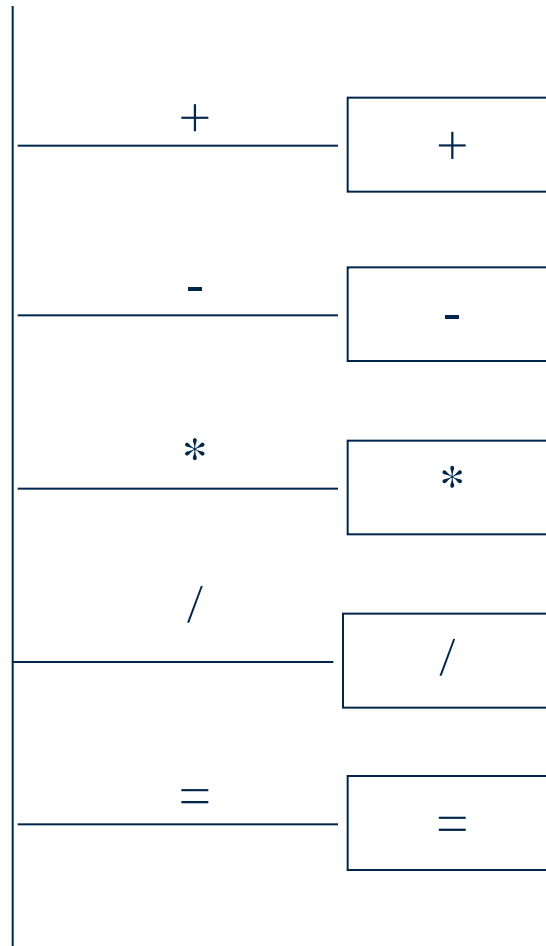
αν έρθει κάτι που δεν είναι ψηφίο τότε θεωρούμε
ότι η αριθμητική σταθερά ολοκληρώθηκε και
πηγαίνουμε σε τελική κατάσταση

Αυτόματο



έτσι αναγνωρίσαμε αριθμητική
σταθερά και ο λεκτικός αναλυτής
τερματίζει αφού την επιστρέψει στον
συντακτικό αναλυτή

Αυτόματο



όταν από την αρχική κατάσταση έρθει το σύμβολο «+» το αναγνωρίζουμε

όταν από την αρχική κατάσταση έρθει το σύμβολο «-» το αναγνωρίζουμε

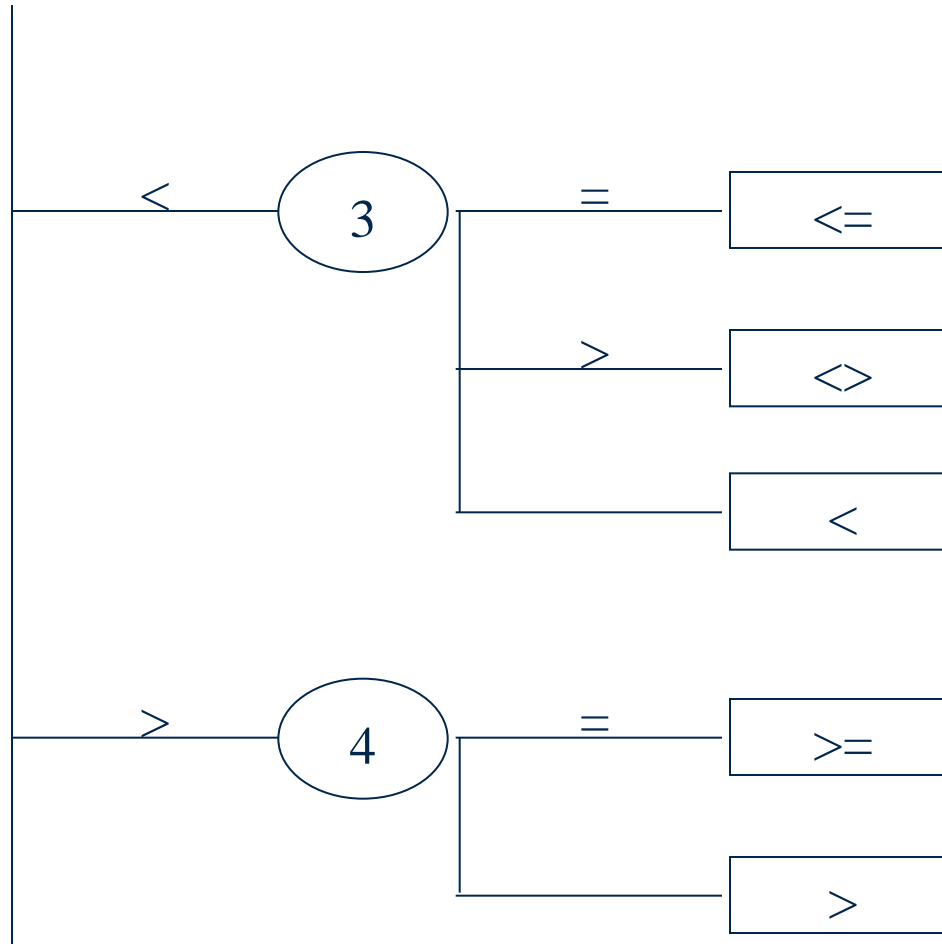
όταν από την αρχική κατάσταση έρθει το σύμβολο «*» το αναγνωρίζουμε

όταν από την αρχική κατάσταση έρθει το σύμβολο «/» το αναγνωρίζουμε

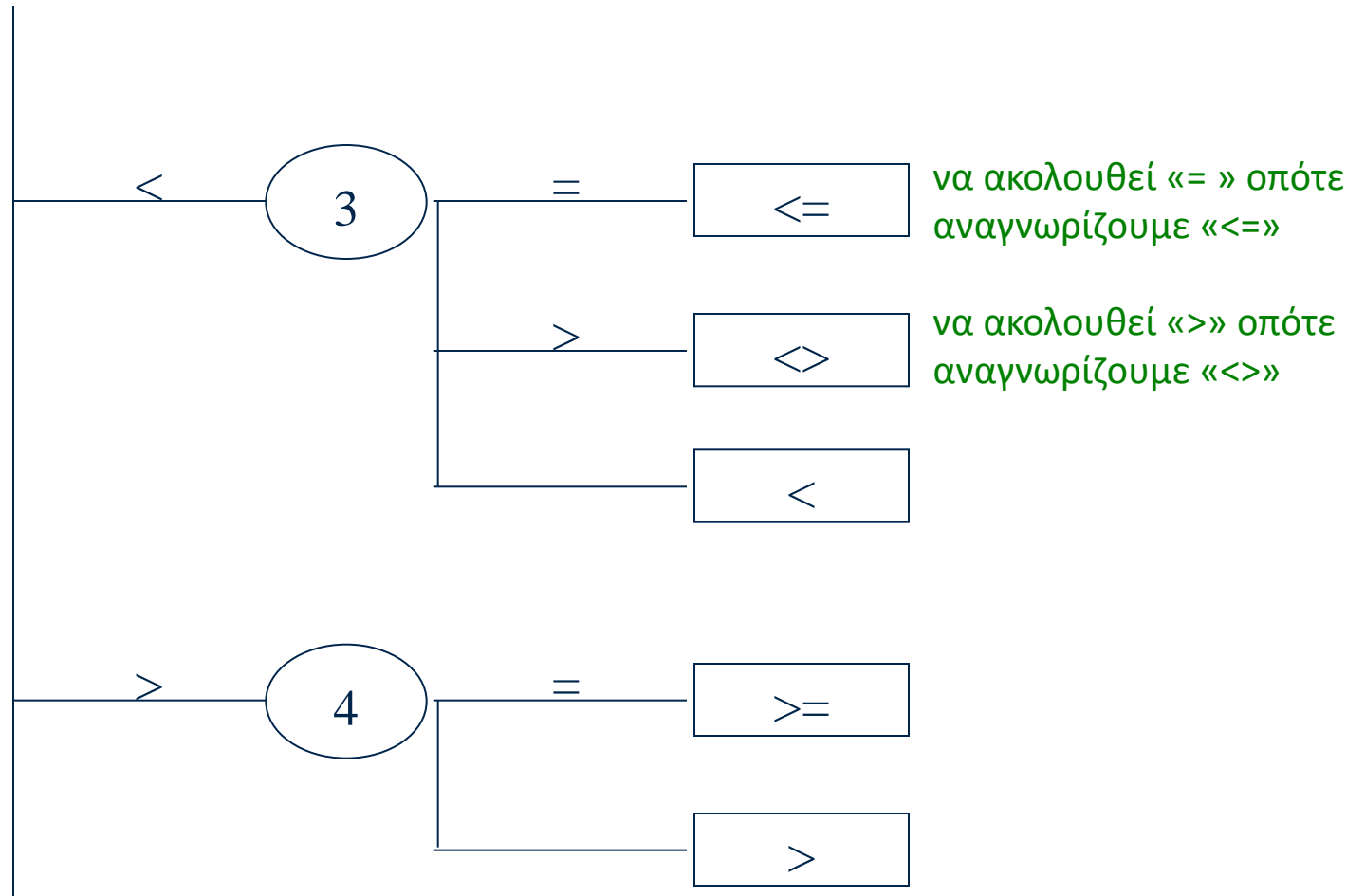
όταν από την αρχική κατάσταση έρθει το σύμβολο «=» το αναγνωρίζουμε

Αυτόματο

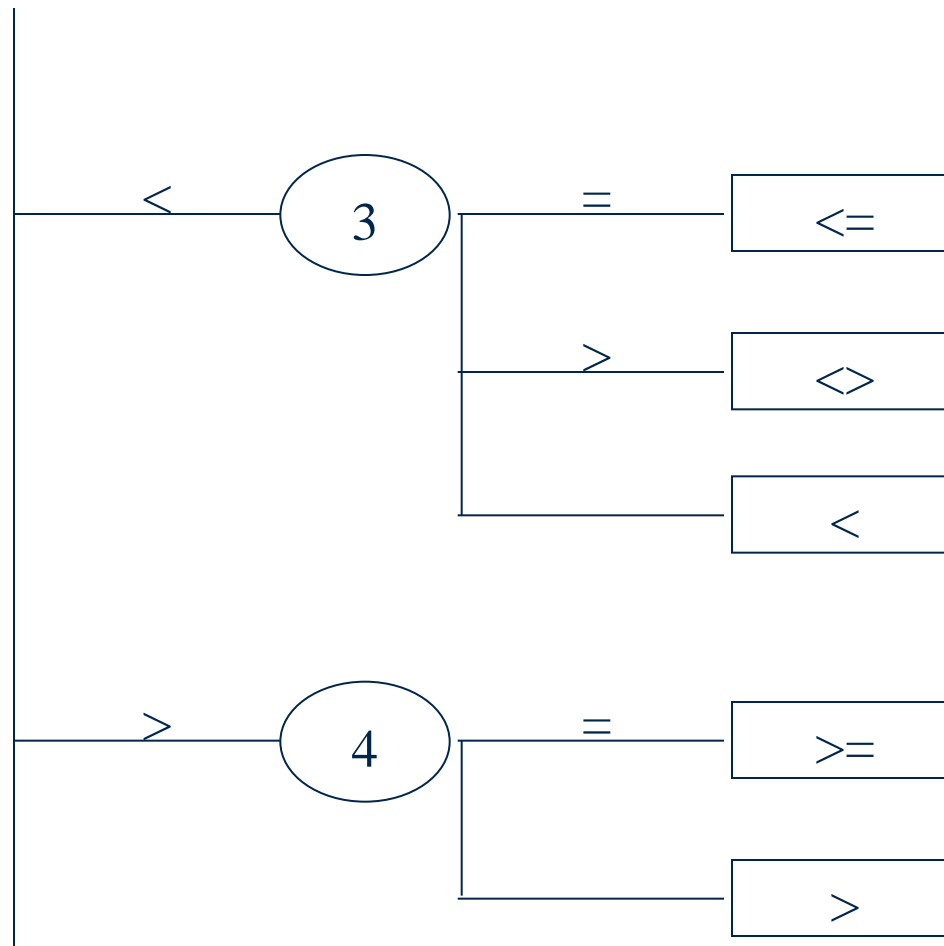
όταν από την αρχική
κατάσταση έρθει το
σύμβολο «<» τότε
υπάρχουν τρεις
περιπτώσεις



Αυτόματο

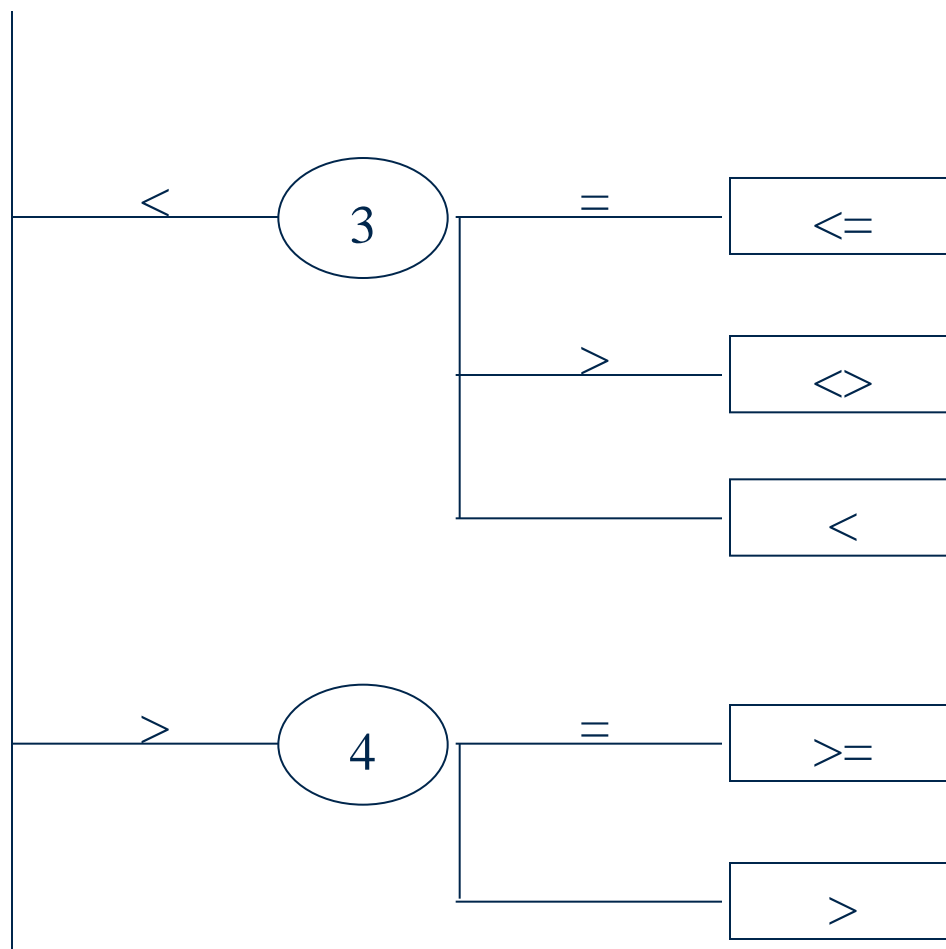


Αυτόματο



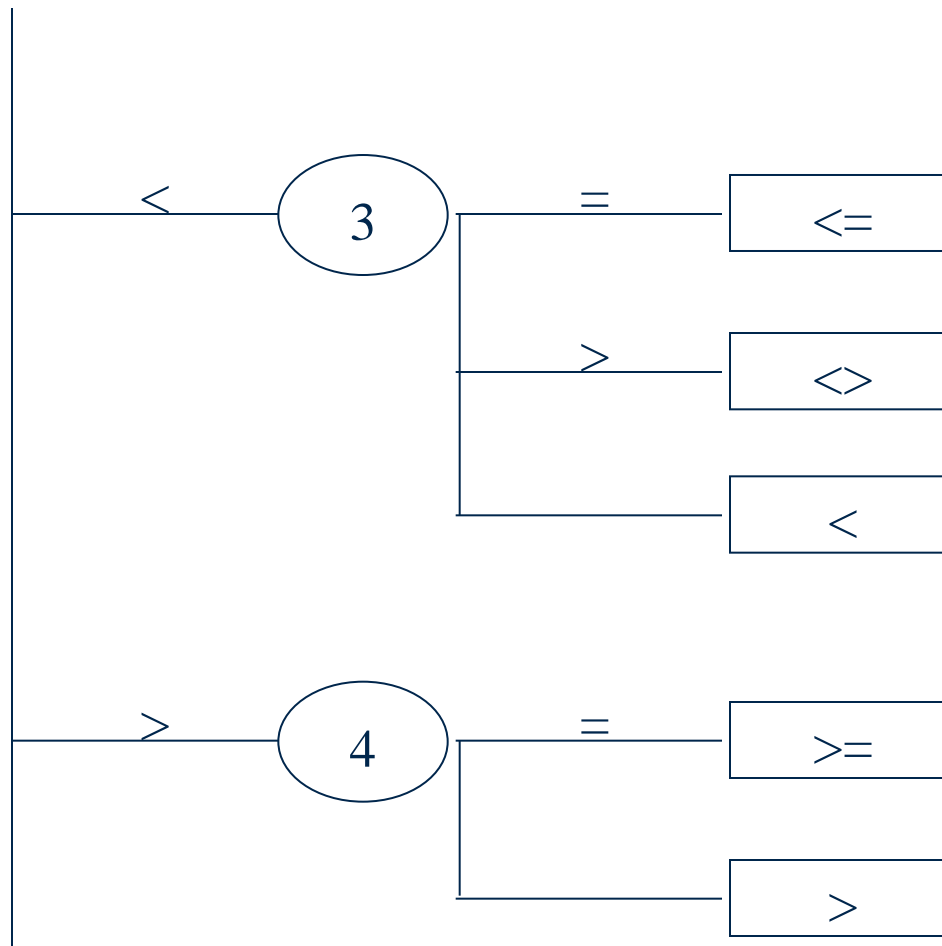
οτιδήποτε άλλο και να
ακολουθεί
αναγνωρίζουμε «<»

Αυτόματο



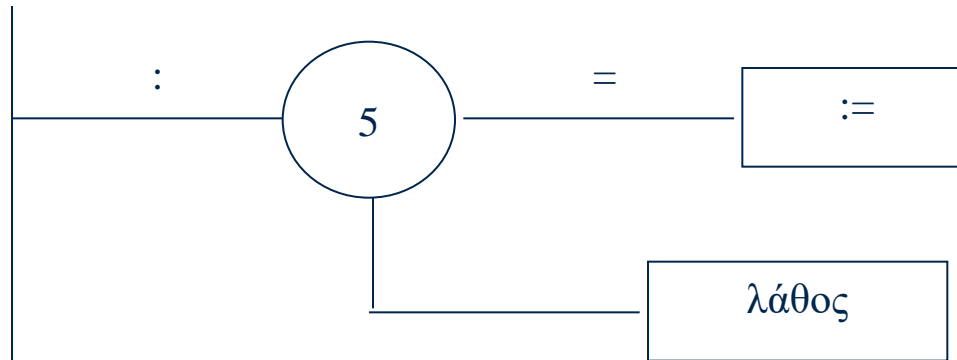
όμοια και εδώ, αν το «>»
το ακολουθεί «=»
αναγνωρίζουμε «<=»

Αυτόματο



οτιδήποτε άλλο και να
ακολουθεί
αναγνωρίζουμε «>»

Αυτόματο



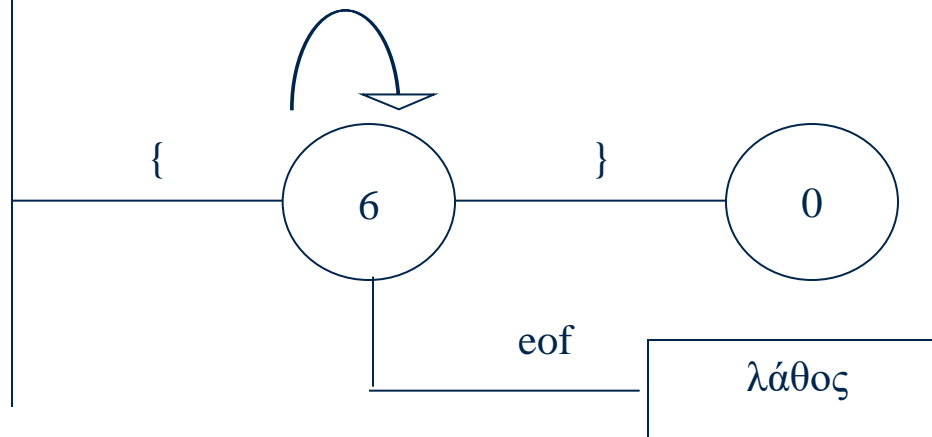
αν από την αρχική κατάσταση συναντήσουμε μια «:»,
τότε υπάρχει μόνο μία επιλογή να καταλήξουμε σε «:=»

έτσι αν μετά την «:» έρθει «=» αναγνωρίζουμε «:=»,
ενώ σε κάθε άλλη περίπτωση αναγνωρίζουμε λάθος

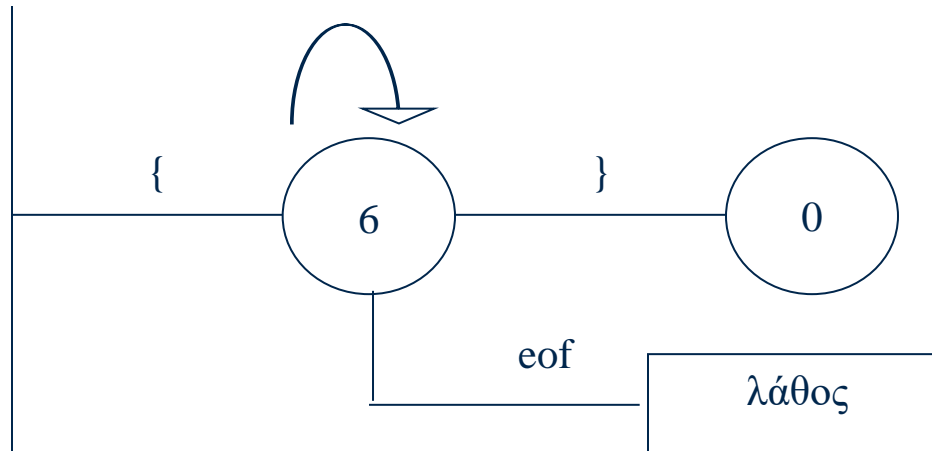
εδώ δεν μπορούμε να αναγνωρίσουμε την «:» ως
λεκτική μονάδα όπως κάναμε για το «<» και το «>»
αφού το «:» δεν αποτελεί σύμβολο της γλώσσας

Αυτόματο

αν από την αρχική κατάσταση συναντήσουμε άνοιγμα σχολίων τότε ό,τι ακολουθήσει αγνοείται έως ότου βρεθεί κλείσιμο σχολίων οπότε μεταβαίνουμε στην κατάσταση 0, που είναι η αρχική κατάσταση



Αυτόματο



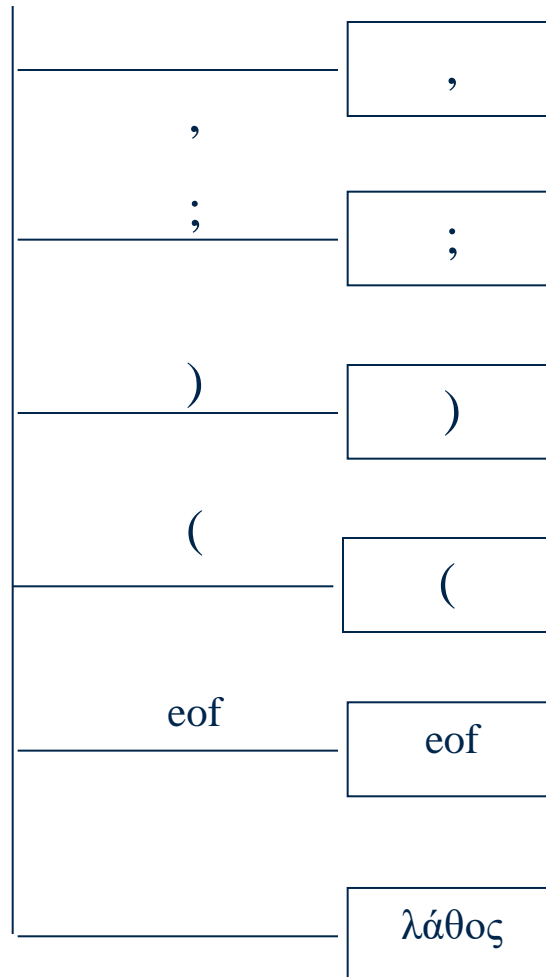
αν πριν βρεθεί κλείσιμο σχολίων βρεθεί τέλος αρχείου
σημαίνει ότι άνοιξαν σχόλια χωρίς να κλείσουν. Προφανώς
έχουμε εδώ λάθος

θα μπορούσε να τροποποιηθεί το αυτόματο αν επιθυμούμε
να θεωρούνται διαφορετικά πράγματα λάθος, για παράδειγμα
αν θέλουμε το {...} να μην είναι επιτρεπτό

Αυτόματο

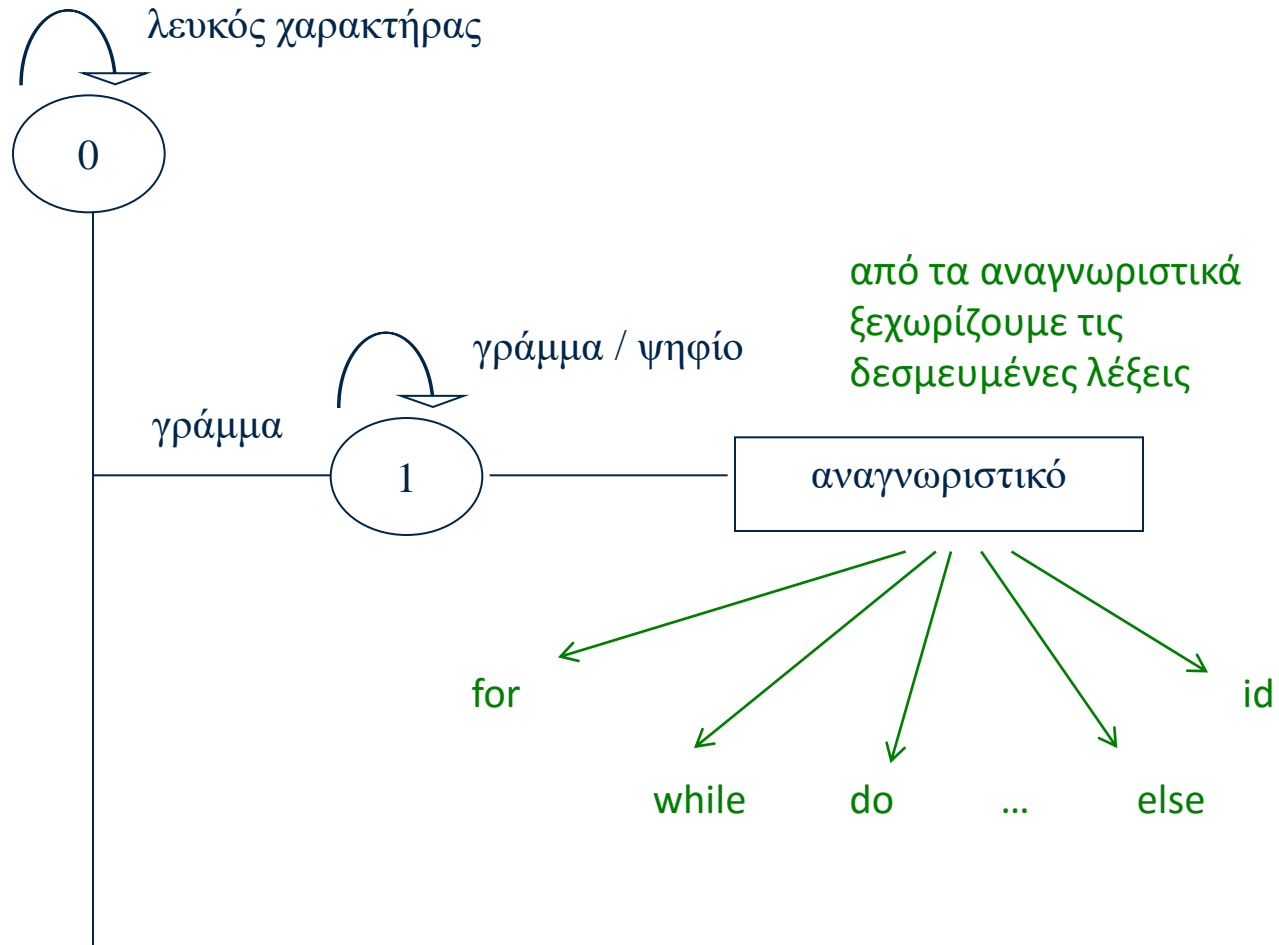
	,	όταν από την αρχική κατάσταση έρθει το σύμβολο «,» το αναγνωρίζουμε
,	;	όταν από την αρχική κατάσταση έρθει το σύμβολο «;» το αναγνωρίζουμε
))	όταν από την αρχική κατάσταση έρθει το σύμβολο «)» το αναγνωρίζουμε
((όταν από την αρχική κατάσταση έρθει το σύμβολο «(» το αναγνωρίζουμε
eof	eof	όταν από την αρχική κατάσταση έρθει το τέλος αρχείου το αναγνωρίζουμε
	λάθος	

Αυτόματο



όταν από την αρχική κατάσταση έρθει
οτιδήποτε άλλο, τότε προφανώς αυτό δε
μπορεί να μας οδηγήσει σε τελική κατάσταση,
οπότε αναγνωρίζουμε την ύπαρξη λάθους

Αυτόματο

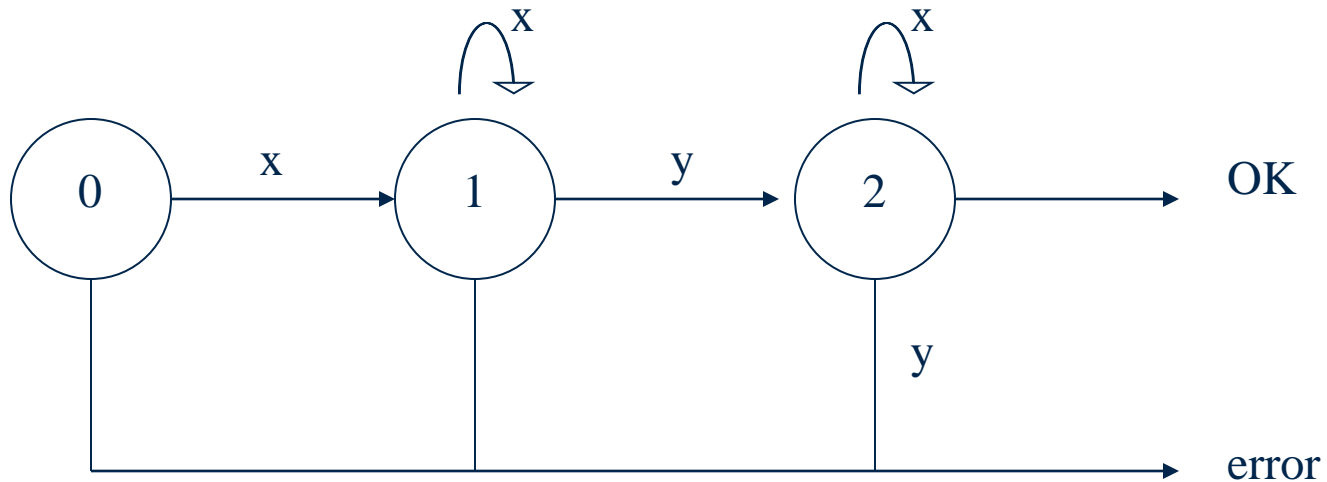


Οπισθοδρόμηση

- # Για να αναγνωρίσουμε μία λεκτική μονάδα μερικές φορές **καταναλώνουμε** ένα χαρακτήρα από την επόμενη λεκτική μονάδα
- # Ο χαρακτήρας αυτός πρέπει να **ενσωματωθεί** στην επόμενη λεκτική μονάδα
- # Αυτό συμβαίνει στο παράδειγμά μας
 - στα αναγνωριστικά και στις αριθμητικές σταθερές
 - στα δεσμευμένα σύμβολα “<” και “>”

Παράδειγμα Υλοποίησης Αυτόματου με Πίνακα

Έστω το αυτόματο το οποίο αναγνωρίζει λεκτικές μονάδες που αποτελούνται από μία σειρά από 1 ή περισσότερα x ακολουθούμενη από ένα y ακριβώς και μετά από μία σειρά από 0 ή περισσότερα x, δηλαδή x^+yx^*



Υλοποίηση με Πίνακα

```
#define x 0  
#define y 1  
#define other 2
```

```
#define state0 0  
#define state1 1  
#define state2 2
```

```
#define error -1  
#define OK -2
```

	x	y	other
state0	state1	error	error
state1	state1	state2	error
state2	state2	error	OK

Υλοποίηση με Πίνακα

```
state = state0;
while (state!=OK && state!=error)
{
    newSymbol(&inp);
    state = Trans_Diagram [state][inp];
}
```

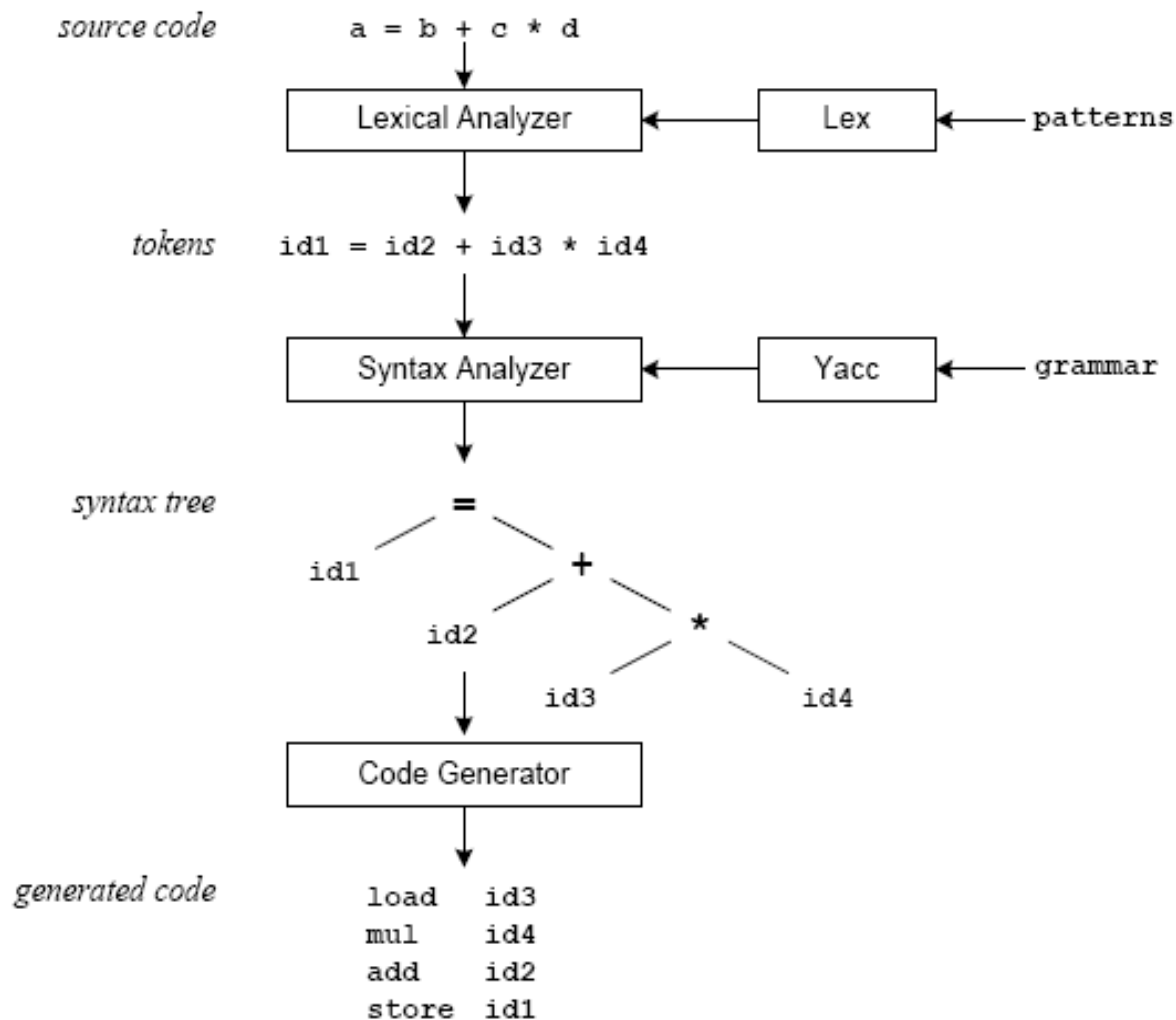
Εναλλακτική Υλοποίηση με Σειρά Εντολών Απόφασης

```
state = state0;
while (state!=OK && state!=error)
{
    if (state==state0 && input==x) state=state1;
    else if (state==state0) state=error;
    else if (state==state1 && input==x) state=state1;
    else if (state==state1 && input==y) state=state2;
    else if (state==state1) state=error;
    else if (state==state2 && input==x) state=state2
    else if (state==state2 && input==y) state=error
    else if (state==state2) state=OK
}
```

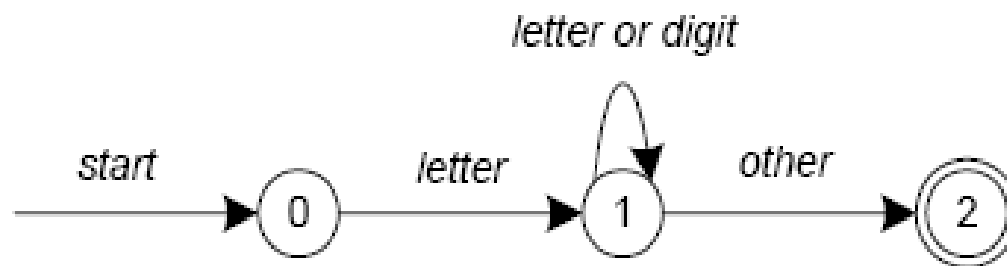
Το Εργαλείο Lex (flex)



Μετάφραση με τα Εργαλεία Lex-Yacc



Αναγνώριση με Κανονικές Εκφράσεις



`letter(letter|digit)*`

Παράδειγμα Αναγνώρισης με το Lex

letter [a-zA-Z]

digit [0-9]

delim [" "\t\n]

id {letter}({letter}|{digit})*

number {digit}+