



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Διακριτά Μαθηματικά

Ενότητα 8: Υπολογισιμότητα & Γλώσσες

Αν. Καθηγητής Κ. Στεργίου

e-mail: kstergiou@uowm.gr

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

- Υπολογισιμότητα:
 - Το “πρόβλημα τερματισμού” του Turing.
- Διατεταγμένα Σύνολα.
- Γλώσσες:
 - Γραμματικές Δομής Φράσεων.
 - Δημιουργία Προτάσεων.
 - Τύποι Γραμματικών και Γλωσσών.



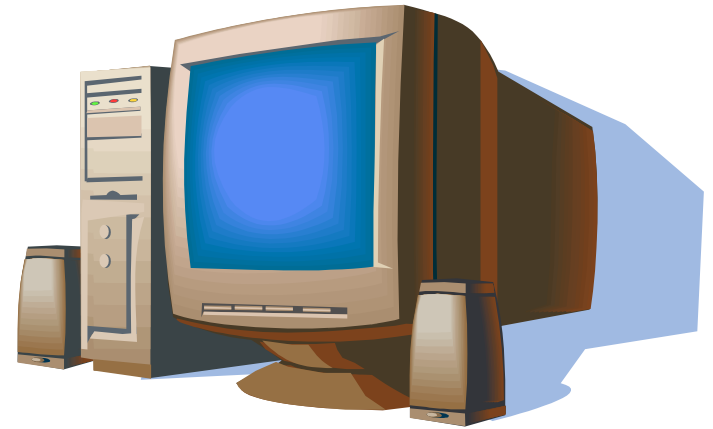
Στόχοι

- Κατανόηση της έννοιας της υπολογισιμότητας.
- Εισαγωγή στη θεωρία τυπικών γλωσσών.
- Κατανόηση των βασικών εργαλείων προσδιορισμού γλωσσών.



Υπολογισιμότητα και Τυπικές Γλώσσες

- Υπάρχουν προβλήματα που δεν μπορούν να λυθούν από υπολογιστή.
- Τι ακριβώς σημαίνει αυτό;;;



Υπολογισσιμότητα (1/12)



Λέω πάντα
ψέματα!

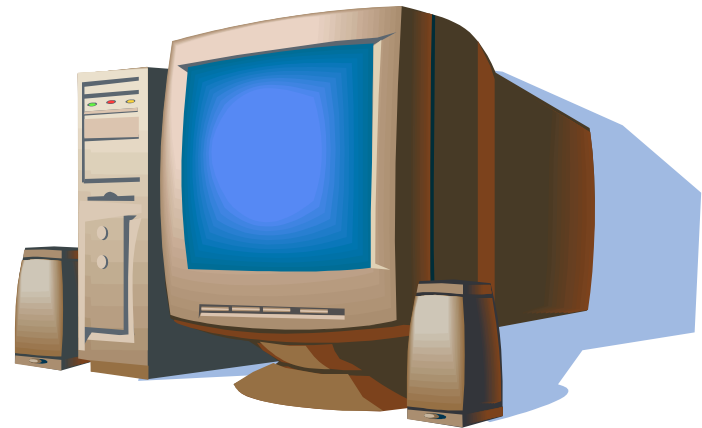
Υπολογισιμότητα (2/12)

Κουρεύω
όσους δεν
κουρεύονται
μόνοι τους!



Υπολογισσιμότητα (3/12)

- Υπάρχουν προβλήματα που δεν μπορούν να λυθούν από υπολογιστή.
- Μπορούμε να κατασκευάσουμε πρόγραμμα που να ανακαλύπτει αν ένα οποιοδήποτε άλλο πρόγραμμα θα τερματίσει;;;



Υπολογισιμότητα (4/12)

πρόγραμμα P

```
01022 Private Function CharByLine(ByVal Line As String) As String
01023 Dim iQuoteChar As Long
01024 Dim sChar As String
01025 Dim sQuoteChar As String
01026 Dim sQuoteChar As String
01027
01028 ' Starts with Row if it is a comment
01029 sLine = Trim(sLine)
01030 If Left(sLine, 1) = " " Then
01031 CharByLine = ""
01032 Exit Function
01033 End If
01034
01035 ' Starts with ' if it is a comment
01036 If Left(sLine, 1) = "'" Then
01037 CharByLine = ""
01038 Exit Function
01039 End If
01040
01041 ' Otherwise ' may not be a comment, so test if it is a comment or so the
01042 ' body of a string
01043 If Left(sLine, 1) = " " Then
01044 sQuoteChar = " "
01045 iQuoteChar = 0
01046
01047 For iQuote = 1 To Len(sLine)
01048 sChar = Mid(sLine, iQuote, 1)
01049
01050 ' If an quote ' ' then an even number of ' characters is found
01051 ' means it is the start of a comment, and odd number means it is
01052 ' part of a string
01053 If sChar = "'" And sQuoteChar = "" Then
01054 If iQuoteChar Mod 2 = 0 Then
01055 sLine = Trim(Left(sLine, iQuote - 1))
01056 Exit For
01057 End If
01058 ElseIf sChar = " " Then
01059 iQuoteChar = iQuoteChar + 1
01060 End If
01061 sQuoteChar = sChar
01062 Next iQuote
01063
01064 CharByLine = sLine
01065 End Function
```

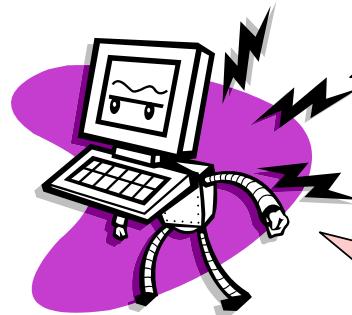
δεδομένα D

ID	Object	Object2	Quantity	Traverse	SPACBOLM	TYPE	SURFACE
1	27	105	1	1	2	part	smooth
2	31	104	1	1	1.5	part	smooth
4	26	102	1	1	2	part	rough
5	40	103	1	1	3	part	rough
6	29	106	2	1	0.5	part	rough
7	47	102	2	1	1	shell	skelens
8	44	107	1	1	0.3	part	rough
9	33	104	1	1	0.75	part	rough
10	44	106	1	1	1.5	part	smooth
11	36	104	4	1	1	leading	strough
12	60	106	1	1	3	shell	skelens
13	49	100	3	1	1.5	leading	strough
14	41	101	1	1	1	leading	strough
15	330	104	3	1	0.2	part	rough
16	265	108	1	1	0.5	part	rough
17	100	175	1	1	0.5	part	strough
18	245	110	1	1	1	part	smooth
19	315	105	2	1	1	part	smooth
20	100	108	1	1	2	shell	skelens
21	212	112	2	2	0.4	part	smooth
22	345	108	3	2	0.7	part	rough
23	245	105	1	2	1	stron	teaser
24	310	106	1	2	1.5	part	smooth
25	310	104	4	2	0.3	part	strough
26	100	103	1	3	1	shell	skelens

αν P τερματίζει με είσοδο D

τυπώνει "P+D
θα τερματίσει"
και τερματίζει

πρόγραμμα T



τυπώνει "P+D δε
θα τερματίσει" και
τερματίζει

αν P δεν τερματίζει με είσοδο D

Υπάρχει τέτοιο πρόγραμμα T;



Υπολογισιμότητα (5/12)

Έστω ότι το πρόγραμμα T υπάρχει...

πρόγραμμα P

```
01522 Private Function CleanUpLib(ByVal s As String) As String
01523 Dim iQuoteCount As Long
01524 Dim iSpace As Long
01525 Dim sChar As String
01526 Dim sPrevChar As String
01527
01528 ' Starts with space if it is a comment
01529 sLine = Trim(sLine)
01530 ' If left(sLine, 1) = "rem" Then
01531 '   CleanUpLine = ""
01532 ' End Function
01533
01534
01535 ' Starts with ' if it is a comment
01536 ' If left(sLine, 1) = "'" Then
01537 '   CleanUpLine = ""
01538 ' End Function
01539
01540
01541 ' Contains ' say and in a comment, so test if it is a comment or in the
01542 ' body of a string.
01543 ' If InStr(sLine, "'") > 0 Then
01544 '   sPrevChar = " "
01545 '   iQuoteCount = 0
01546
01547 ' For iQuoteCount = 1 To Len(sLine)
01548 '   sChar = Mid(sLine, iQuoteCount, 1)
01549 '   ' If we count "' then an even number of " characters in front
01550 '   ' means it is the start of a comment, and odd number means it is
01551 '   ' part of a string.
01552 '   If sChar = "" And sPrevChar = "" Then
01553 '     ' If iQuoteCount Mod 2 = 0 Then
01554 '       sLine = Trim(Left(sLine, iQuoteCount - 1))
01555 '     End If
01556 '   End If
01557 '   If sChar = "'" And sPrevChar = "" Then
01558 '     iQuoteCount = iQuoteCount + 1
01559 '   End If
01560 '   sPrevChar = sChar
01561 ' End For
01562 ' CleanUpLine = sLine
01563 End Function
```

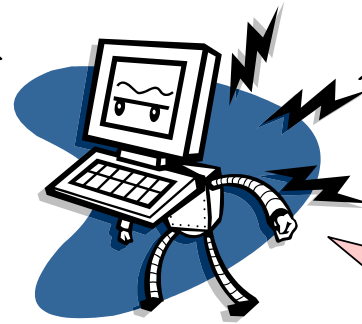
δεδομένα D

ID	Orient1	Orient2	Quantity	Traversal	SPACING	TYPE	SURFACE
1	03	06	1	1	2	joint	smooth
2	42	03	1	1	1	joint	smooth
3	01	04	1	1	1.5	joint	smooth
4	90	02	1	1	2	joint	smooth
5	40	03	1	3	joint	rough	
6	23	06	2	1	0.5	joint	rough
7	47	02	1	1	1	shew	stikens
8	244	07	1	1	0.3	joint	rough
9	04	04	1	1	0.75	joint	rough
10	44	06	1	1	1.5	joint	smooth
11	60	04	4	1	1	bedding	smooth
12	90	06	1	1	3	shew	stikens
13	69	00	3	1	1.5	bedding	smooth
14	69	01	1	1	1	joint	rough
15	30	04	3	1	0.2	joint	rough
16	265	08	1	1	0.5	joint	rough
17	200	75	1	1	0.5	joint	rough
18	316	70	1	1	1	joint	smooth
19	245	05	1	1	1	joint	smooth
20	200	08	1	1	2	shew	stikens
21	372	72	2	2	0.4	joint	smooth
22	246	08	2	0.2	joint	rough	
23	308	05	1	2	1	shew	stikens
24	240	06	2	1.4	joint	smooth	
25	710	04	4	2	0.3	joint	smooth
26	100	07	1	1	1	shew	stikens

αν P τερματίζει με είσοδο D

T' δεν τερματίζει

πρόγραμμα T'



T' τερματίζει

αν P δεν τερματίζει με είσοδο D



Υπολογισιμότητα (6/12)

πρόγραμμα P

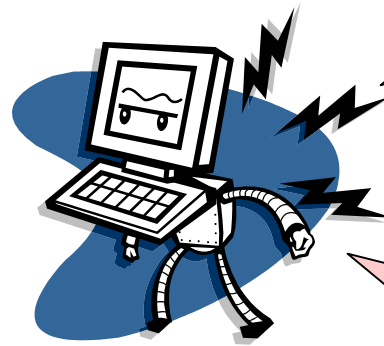
```
01222 Private Function CleanLine(ByVal Line As String) As String
01223 Dim iQuoteCount As Long
01224 Dim iCount As Long
01225 Dim sChar As String
01226 Dim sPrevChar As String
01227
01228 ' Starts with Has it is a comment
01229 sLine = Trim(sLine)
01230 If Left(sLine, 1) = "Has" Then
01231   CleanByLine = ""
01232   Exit Function
01233 End If
01234
01235 ' Starts with ' it is a comment
01236 If Left(sLine, 1) = "'" Then
01237   CleanByLine = ""
01238   Exit Function
01239 End If
01240
01241 ' Contains ' any and in a comment, so test if it is a comment or in the
01242 ' body of a string
01243 If InStr(sLine, "'") > 0 Then
01244   sPrevChar = ""
01245   iQuoteCount = 0
01246   For iCount = 1 To Len(sLine)
01247     sChar = Mid(sLine, iCount, 1)
01248     ' If we found "" then an even number of " characters in front
01249     ' means it is the start of a comment, and odd number means it is
01250     ' part of a string
01251     If sChar = "" And sPrevChar = "" Then
01252       If iQuoteCount Mod 2 = 0 Then
01253         sLine = Trim(Left(sLine, iCount - 1))
01254         Exit For
01255       End If
01256     ElseIf sChar = "" Then
01257       iQuoteCount = iQuoteCount + 1
01258     End If
01259     sPrevChar = sChar
01260   Next iCount
01261 End If
01262 CleanByLine = sLine
01263 End Function
```

δεδομένα P

```
01222 Private Function CleanLine(ByVal Line As String) As String
01223 Dim iQuoteCount As Long
01224 Dim iCount As Long
01225 Dim sChar As String
01226 Dim sPrevChar As String
01227
01228 ' Starts with Has it is a comment
01229 sLine = Trim(sLine)
01230 If Left(sLine, 1) = "Has" Then
01231   CleanByLine = ""
01232   Exit Function
01233 End If
01234
01235 ' Starts with ' it is a comment
01236 If Left(sLine, 1) = "'" Then
01237   CleanByLine = ""
01238   Exit Function
01239 End If
01240
01241 ' Contains ' any and in a comment, so test if it is a comment or in the
01242 ' body of a string
01243 If InStr(sLine, "'") > 0 Then
01244   sPrevChar = ""
01245   iQuoteCount = 0
01246   For iCount = 1 To Len(sLine)
01247     sChar = Mid(sLine, iCount, 1)
01248     ' If we found "" then an even number of " characters in front
01249     ' means it is the start of a comment, and odd number means it is
01250     ' part of a string
01251     If sChar = "" And sPrevChar = "" Then
01252       If iQuoteCount Mod 2 = 0 Then
01253         sLine = Trim(Left(sLine, iCount - 1))
01254         Exit For
01255       End If
01256     ElseIf sChar = "" Then
01257       iQuoteCount = iQuoteCount + 1
01258     End If
01259     sPrevChar = sChar
01260   Next iCount
01261 End If
01262 CleanByLine = sLine
01263 End Function
```

αν P τερματίζει με είσοδο P

πρόγραμμα T'



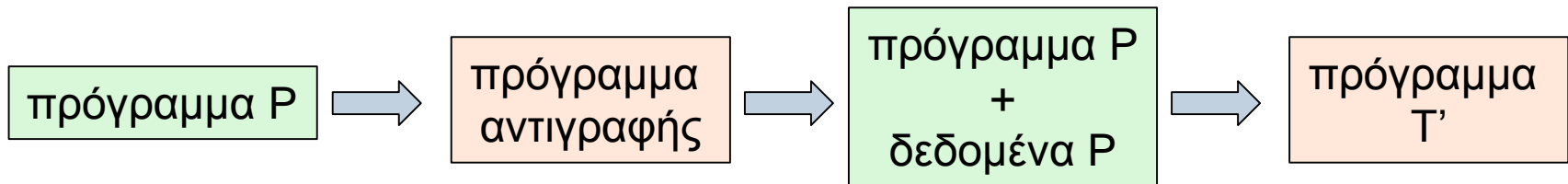
T' δεν τερματίζει

T' τερματίζει

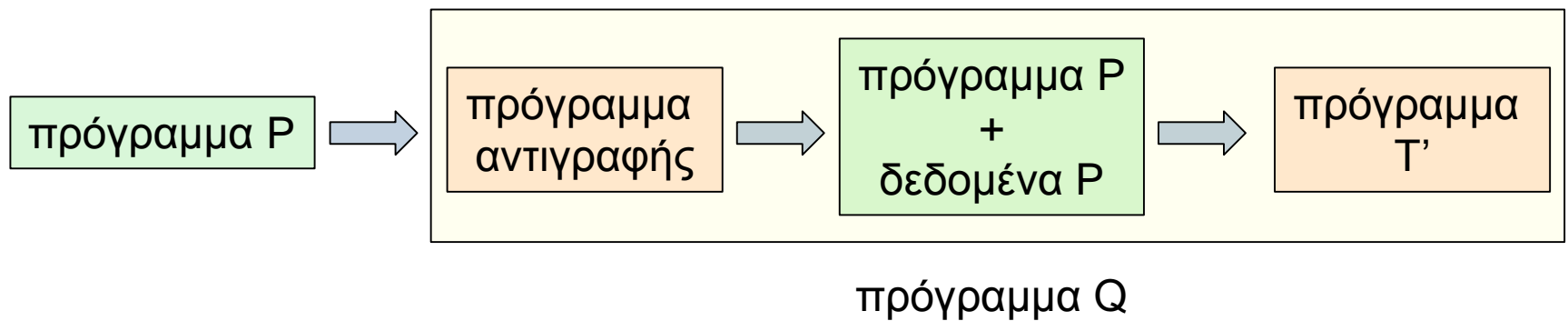
αν P δεν τερματίζει με είσοδο P



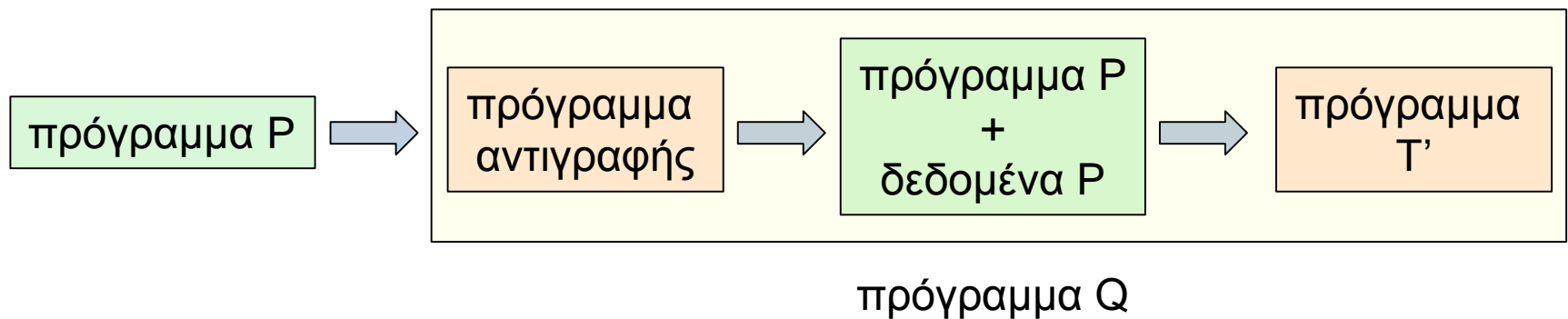
Υπολογισιμότητα (7/12)



Υπολογισιμότητα (8/12)



Υπολογισιμότητα (9/12)



Τι κάνει το Q με είσοδο Q;



Υπολογισιμότητα (10/12)

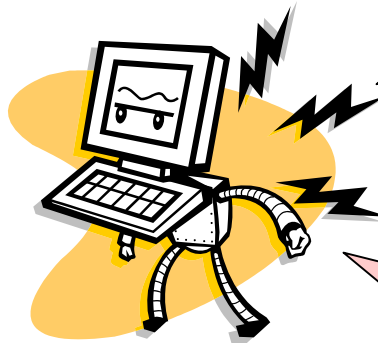
αν Q τερματίζει με είσοδο Q

Q δεν τερματίζει

πρόγραμμα Q

```
01522 Private Function CheckLine(ByVal sLine As String) As String
01523
01524 Dim iQuoteCount As Long
01525 Dim iLocust As Long
01526 Dim sChar As String
01527 Dim sPrevChar As String
01528
01529 ' Starts with Qm it is a comment
01530 sLine = Trim(sLine)
01531 If Left(sLine, 1) = "Qm" Then
01532   ClearLine = ""
01533   Exit Function
01534 End If
01535
01536 ' Starts with ' if it is a comment
01537 If Left(sLine, 1) = "'" Then
01538   ClearLine = ""
01539   Exit Function
01540 End If
01541
01542 ' Contains ' say and in a comment, so test if it is a comment or in the
01543 ' body of a string
01544 If InStr(sLine, "'") > 0 Then
01545   sPrevChar = ""
01546   iQuoteCount = 0
01547
01548   For iLocust = 1 To Len(sLine)
01549     sChar = Mid(sLine, iLocust, 1)
01550     ' If we found "" then an even number of " characters in front
01551     ' means it is the start of a comment, and odd number means it is
01552     ' part of a string
01553     If sChar = "" And sPrevChar = "" Then
01554       If iQuoteCount Mod 2 = 0 Then
01555         sLine = Trim(Left(sLine, iLocust - 1))
01556         Exit For
01557       End If
01558     ElseIf sChar = "" Then
01559       iQuoteCount = iQuoteCount + 1
01560     End If
01561     sPrevChar = sChar
01562   Next iLocust
01563 End If
01564
01565 ClearLine = sLine
01566 End Function
```

πρόγραμμα Q

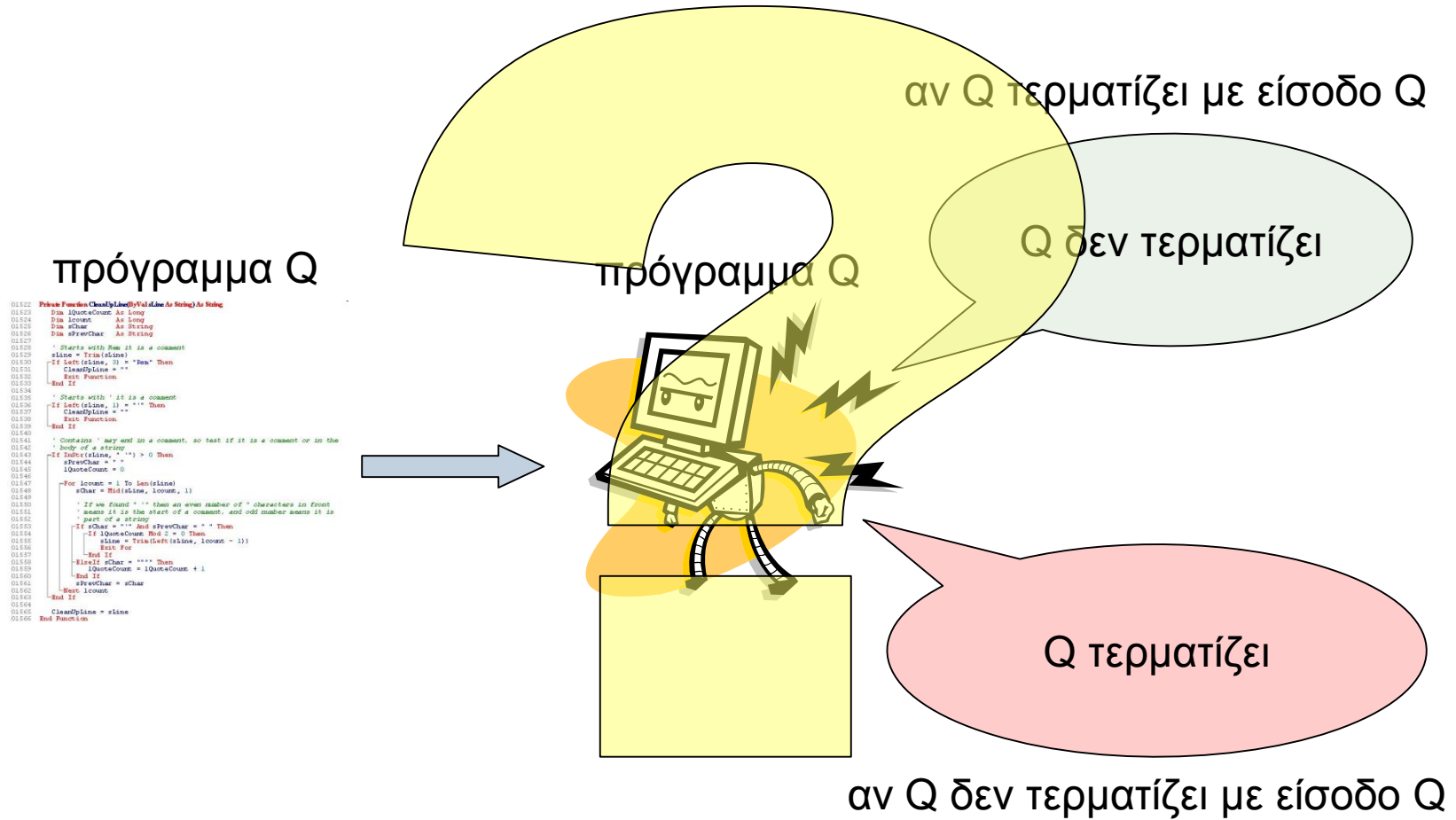


Q τερματίζει

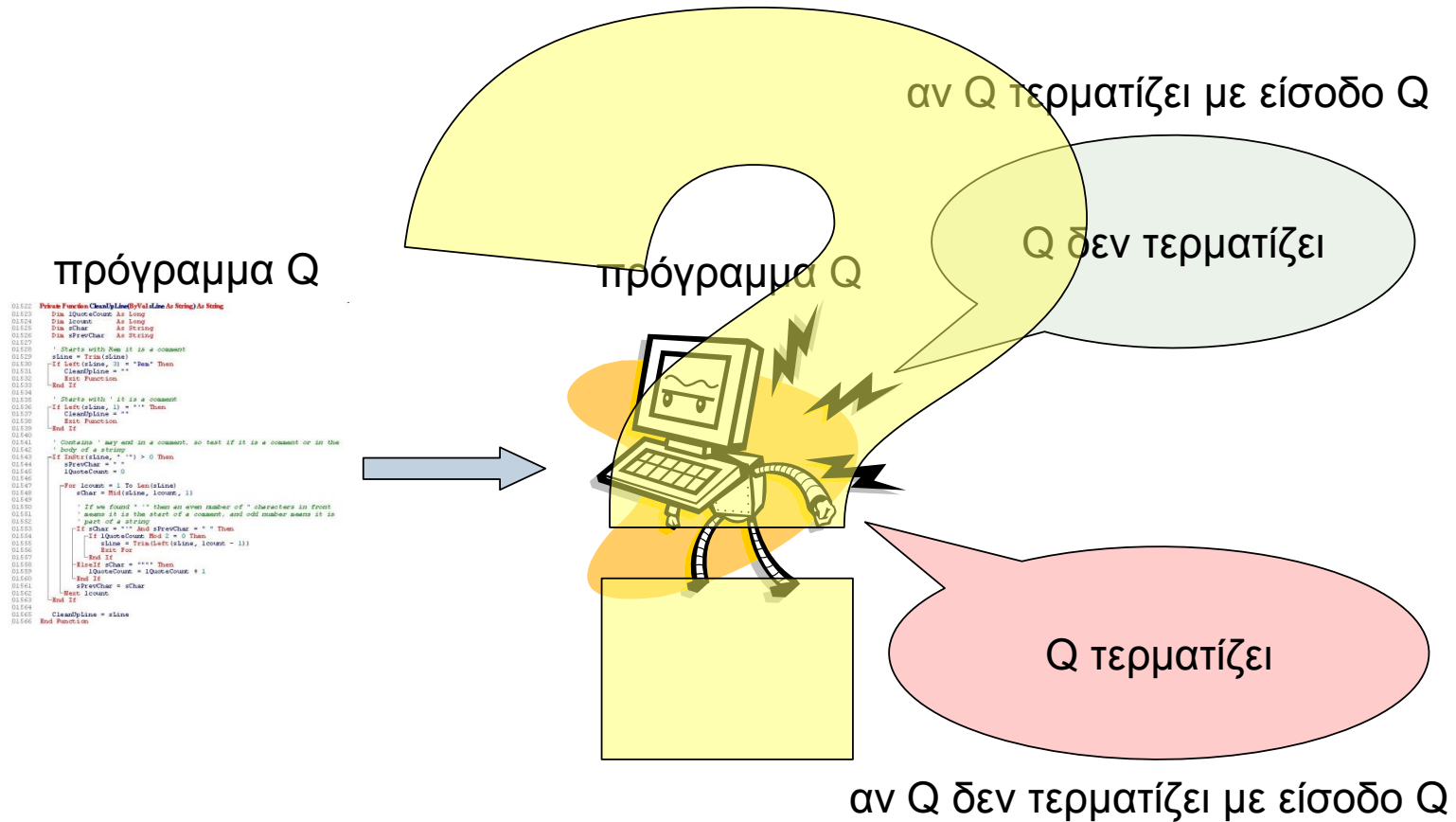
αν Q δεν τερματίζει με είσοδο Q



Υπολογισιμότητα (11/12)



Υπολογισιμότητα (12/12)



Η υπόθεση ότι το πρόγραμμα T υπάρχει οδηγεί σε αντίφαση!



Διατεταγμένα Σύνολα (1/2)

- Ένα σύνολο είναι μια **μη διατεταγμένη** συλλογή αντικειμένων.
- Ένα **διατεταγμένο ζεύγος** είναι ένα ζεύγος αντικειμένων τοποθετημένων με συγκεκριμένη σειρά.
 - π.χ. (a,b)
- Ένα διατεταγμένο ζεύγος διαφέρει από ένα σύνολο δύο αντικειμένων κατά δύο τρόπους.
 - Η σειρά των αντικειμένων είναι σημαντική. Δηλ. τα (a,b) και (b,a) είναι διαφορετικά.
 - Μπορούμε να έχουμε το ίδιο αντικείμενο και στις δύο θέσεις. Δηλ. το (a,a) είναι αποδεκτό διατεταγμένο ζεύγος.
 - Μπορεί π.χ. να συμβολίζει τους φοιτητές που πήραν την καλύτερη βαθμολογία στα Μαθηματικά και τη Φυσική.



Διατεταγμένα Σύνολα (2/2)

- Μπορούμε να επεκτείνουμε την έννοια του διατεταγμένου ζεύγους σε δυάδες, τριάδες, κτλ.
 - π.χ. (a,b,c) είναι μια διατεταγμένη τριάδα.
- Μια διατεταγμένη τριάδα αντικειμένων μπορεί να οριστεί ως ένα διατεταγμένο ζεύγος $((a,b),c)$ όπου το πρώτο στοιχείο είναι και αυτό ένα διατεταγμένο ζεύγος.
 - Ο ορισμός αυτός μπορεί να επεκταθεί σε n -άδες.
 - Μια διατεταγμένη n -άδα είναι ένα διατεταγμένο ζεύγος όπου το πρώτο συστατικό του είναι μια διατεταγμένη $(n-1)$ -άδα .
 - Στην ουσία μια διατεταγμένη n -άδα είναι ένα διατεταγμένο σύνολο από n στοιχεία.



Γλώσσες (1/7)

- Έστω $A = \{a,b,\dots,x,y,z\}$ το αγγλικό αλφάβητο. Μια λέξη n γραμμάτων είναι μια διατεταγμένη n -άδα των γραμμάτων του αλφαβήτου.
 - π.χ. $(((((s,o),l),v),e),r)$.
 - Μια διατεταγμένη n -άδα εναλλακτικά ονομάζεται ακολουθία, συμβολοσειρά, ή πρόταση.
 - A^n είναι το σύνολο όλων των ακολουθιών n γραμμάτων από το σύνολο A .
 - A^* είναι το σύνολο όλων των ακολουθιών από το σύνολο A .
- Αντί για το αγγλικό αλφάβητο μπορούμε να θεωρήσουμε το αλφάβητο οποιασδήποτε γλώσσας (είτε φυσικής είτε γλώσσας προγραμματισμού).



Γλώσσες (2/7)

- Αλφάβητο $A = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- Λέξη με n γράμματα = διατεταγμένη n -άδα γραμμάτων του A .
- A^n = σύνολο όλων των ακολουθιών n γραμμάτων του A .
- A^* = σύνολο όλων των ακολουθιών n γραμμάτων του A για κάθε $n=1,2,\dots$

Παραδείγματα

- Προτάσεις στην ελληνική γλώσσα

$$B = \{\alpha, \beta, \dots, \omega\} \cup \{A, B, \dots, \Omega\} \cup \{., ', ,, :, !, ;, _\}$$

$$T\iota_ώρα_είναι; \in B^*$$

- Εντολές σε γλώσσα προγραμματισμού

$$C = \{A, B, \dots, Y, Z\} \cup \{0, 1, 2, \dots, 8, 9\} \cup \{+, -, *, /, ;, ,, =\}$$

$$X=2*T*Z+10; \in C^*$$



Γλώσσες (3/7)

- Αλφάβητο $A = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- Λέξη με n γράμματα = διατεταγμένη n -άδα γραμμάτων του A .
- A^n = σύνολο όλων των ακολουθιών n γραμμάτων του A .
- A^* = σύνολο όλων των ακολουθιών n γραμμάτων του A για κάθε $n=1,2,\dots$

Γενικά μία γλώσσα (επί του αλφαβήτου A) είναι ένα υποσύνολο του A^*

Επομένως ορίζουμε μία γλώσσα ως ένα σύνολο συμβολοσειρών

Π.χ. $A=\{\alpha, \beta, \gamma\}$

$L_1 = \{\alpha, \alpha\beta, \beta\gamma, \alpha\beta\gamma\}$

$L_2 = \{\alpha\beta^i\gamma^i \mid i \geq 1\}$



Γλώσσες (4/7)

- Έστω A ένα πεπερασμένο σύνολο που αποτελεί το αλφάβητο μιας γλώσσας.
- Μια γλώσσα επί του αλφάβητου A είναι ένα υποσύνολο του A^* .
- Π.χ. Έστω $A = \{a, b, c\}$. Τα παρακάτω σύνολα είναι γλώσσες επί του αλφαβήτου A .
 - $L1 = \{a, aa, ab, ac, abc, cab\}$,
 - $L2 = \{aba, aabaa\}$,
 - $L3 = \{ \}$,
 - $L4 = \{a^i c b^i \mid i \geq 1\}$.
 - το a^i συμβολίζει μια ακολουθία από i το πλήθος a .



Γλώσσες (5/7)

- Αλφάβητο $A = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- Λέξη με n γράμματα = διατεταγμένη n -άδα γραμμάτων του A .
- A^n = σύνολο όλων των ακολουθιών n γραμμάτων του A .
- A^* = σύνολο όλων των ακολουθιών n γραμμάτων του A για κάθε $n=1,2,\dots$

Γενικά μία γλώσσα (επί του αλφαβήτου A) είναι ένα υποσύνολο του A^*

Επομένως ορίζουμε μία γλώσσα ως ένα σύνολο συμβολοσειρών

$L_1 \cup L_2$ = περιέχει προτάσεις που ανήκουν στην L_1 ή στην L_2

$L_1 \cap L_2$ = περιέχει προτάσεις που ανήκουν και στην L_1 και στην L_2



Γλώσσες (6/7)

- Αλφάβητο $A = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- Λέξη με n γράμματα = διατεταγμένη n -άδα γραμμάτων του A .
- A^n = σύνολο όλων των ακολουθιών n γραμμάτων του A .
- A^* = σύνολο όλων των ακολουθιών n γραμμάτων του A για κάθε $n=1,2,\dots$

Γενικά μία γλώσσα (επί του αλφαβήτου A) είναι ένα υποσύνολο του A^*

Επομένως ορίζουμε μία γλώσσα ως ένα σύνολο συμβολοσειρών

$$\{\alpha^i \beta \gamma^j \mid 1 \leq i < j\} \cup \{\alpha^i \beta \gamma^j \mid i > j \geq 1\} = \text{[Blank Box]}$$

$$\{\alpha^i \beta^i \gamma^j \mid i, j \geq 1\} \cap \{\alpha^i \beta^j \gamma^j \mid i, j \geq 1\} = \text{[Blank Box]}$$

$$\{\alpha^i \beta \gamma^j \mid i \geq j \geq 1\} \oplus \{\alpha^i \beta \gamma^j \mid 1 \leq i \leq j\} = \text{[Blank Box]}$$

$$\{\alpha^i \beta \gamma^j \mid i, j \geq 1\} - \{\alpha^i \beta \gamma^i \mid i \geq 1\} = \text{[Blank Box]}$$



Γλώσσες (7/7)

- Αλφάβητο $A = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- Λέξη με n γράμματα = διατεταγμένη n -άδα γραμμάτων του A .
- A^n = σύνολο όλων των ακολουθιών n γραμμάτων του A .
- A^* = σύνολο όλων των ακολουθιών n γραμμάτων του A για κάθε $n=1,2,\dots$
- Γενικά μία γλώσσα (επί του αλφαβήτου A) είναι ένα υποσύνολο του A^* .
- Επομένως ορίζουμε μία γλώσσα ως ένα σύνολο συμβολοσειρών.

- **Ωστόσο οι τρόποι περιγραφής συνόλων που έχουμε δει δεν είναι κατάλληλοι για την περιγραφή γλωσσών.**

- Επιπλέον, δεδομένης της περιγραφής μίας γλώσσας, θέλουμε να εκτελούνται αυτόματα οι ακόλουθες λειτουργίες:
 - παραγωγή συμβολοσειρών της γλώσσας,
 - καθορισμός του αν μία δεδομένη λέξη ανήκει στη γλώσσα ή όχι.



Γραμματικές Δομής Φράσεως (1/2)

- Ξέροντας ότι μια γλώσσα είναι ένα σύνολο συμβολοσειρών, πως μπορούμε να την ορίσουμε;
 - Η πλήρης παράθεση των στοιχείων της απορρίπτεται μια και οι περισσότερες γλώσσες περιέχουν άπειρο πλήθος συμβολοσειρών.
 - Επίσης, για μια μη τετριμμένη γλώσσα είναι εξαιρετικά περίπλοκο να περιγραφούν οι ιδιότητες που κατά μοναδικό τρόπο χαρακτηρίζουν τις συμβολοσειρές της.
- Για τις περισσότερες εφαρμογές μας ενδιαφέρουν τα εξής:
 - Δεδομένης της περιγραφής μιας γλώσσας, αυτομάτως να δημιουργήσουμε συμβολοσειρές της.
 - Δεδομένης της περιγραφής μιας γλώσσας, να μπορούμε να καθορίσουμε αν κάποια συμβολοσειρά ανήκει στη γλώσσα.



Γραμματικές Δομής Φράσεως (2/2)

- Για να απαντήσουμε σε τέτοια ερωτήματα θα περιγράψουμε γλώσσες χρησιμοποιώντας γραμματικές. Συγκεκριμένα θα μελετήσουμε μια κλάση γραμματικών που ονομάζονται *γραμματικές δομής φράσεως*.
- Ας υποθέσουμε ότι περιοριζόμαστε σε ένα υποσύνολο των προτάσεων στα Ελληνικά. Αρχικά θα ορίσουμε τι είναι πρόταση:
 - Μια πρόταση είναι μια **φράση-ουσιαστικό** ακολουθούμενη από μια **φράση-μεταβατικό-ρήμα** και από μια ακόμα **φράση-ουσιαστικό**.
 - Μια πρόταση είναι μια **φράση-ουσιαστικό** ακολουθούμενη από μια **φράση-αμετάβατο-ρήμα**.



Περιγραφή

γλώσσας μέσω γραμματικής (1/9)

Παραγωγή προτάσεων στα ελληνικά

πρόταση → **φράση-ουσιαστικό** που ακολουθείται από **φράση-μεταβατικό-ρήμα** και ακόμα μία **φράση-ουσιαστικό**

πρόταση → **φράση-ουσιαστικό** που ακολουθείται από **φράση-αμετάβατο-ρήμα**

φράση-ουσιαστικό → **άρθρο** που ακολουθείται από **ουσιαστικό**

φράση-ουσιαστικό → **ουσιαστικό**

φράση-μεταβατικό-ρήμα → **μεταβατικό ρήμα**

φράση-αμετάβατο-ρήμα → **αμετάβατο ρήμα** που ακολουθείται από **επίρρημα**

φράση-αμετάβατο-ρήμα → **αμετάβατο ρήμα**



Περιγραφή

γλώσσας μέσω γραμματικής (2/9)

Παραγωγή προτάσεων στα ελληνικά

ουσιαστικό → πρόγραμμα ή υπολογιστής ή αρχείο

άρθρο → ο ή το ή ένα

μεταβατικό ρήμα → διαβάζει

αμετάβατο ρήμα → τρέχει

επίρρημα → γρήγορα



Περιγραφή

γλώσσας μέσω γραμματικής (3/9)

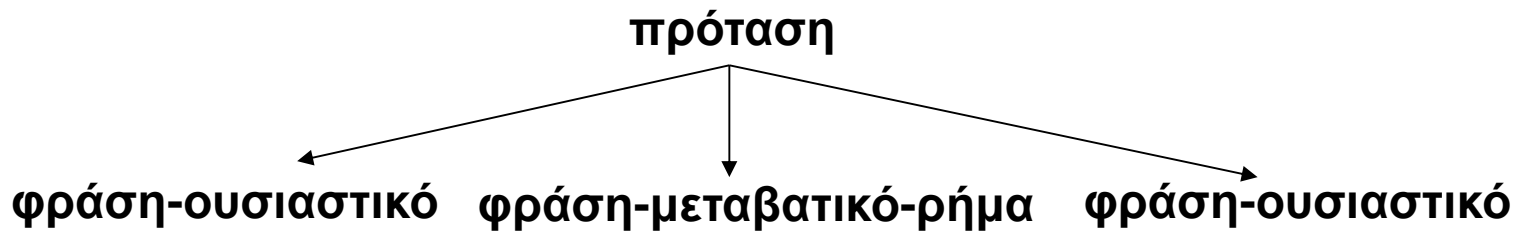
- Παραγωγή προτάσεων στα ελληνικά.
 - Παράδειγμα: «το πρόγραμμα διαβάζει ένα αρχείο».



Περιγραφή γλώσσας μέσω γραμματικής (4/9)

Παραγωγή προτάσεων στα ελληνικά

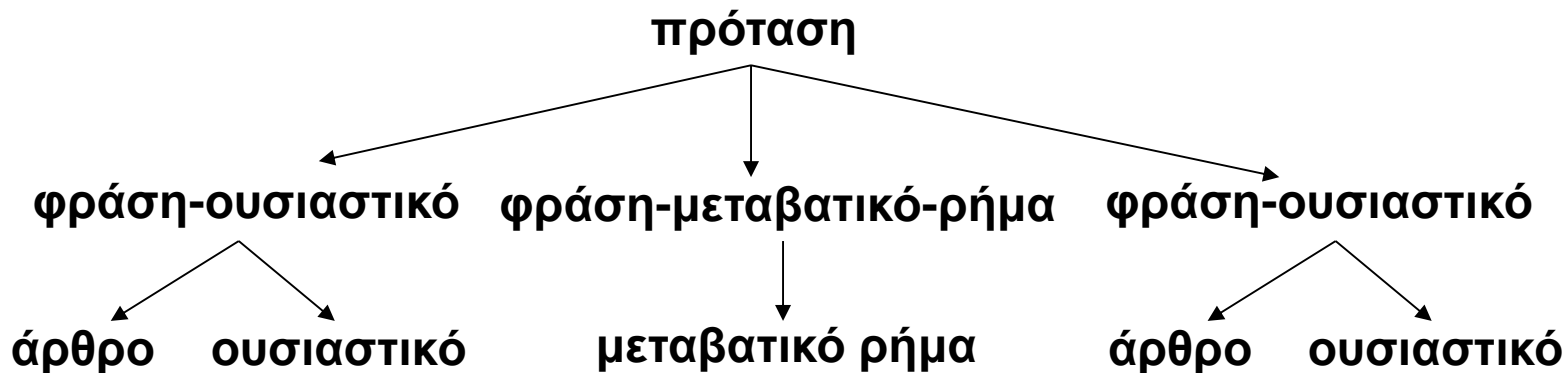
Παράδειγμα : «το πρόγραμμα διαβάζει ένα αρχείο»



Περιγραφή γλώσσας μέσω γραμματικής (5/9)

Παραγωγή προτάσεων στα ελληνικά

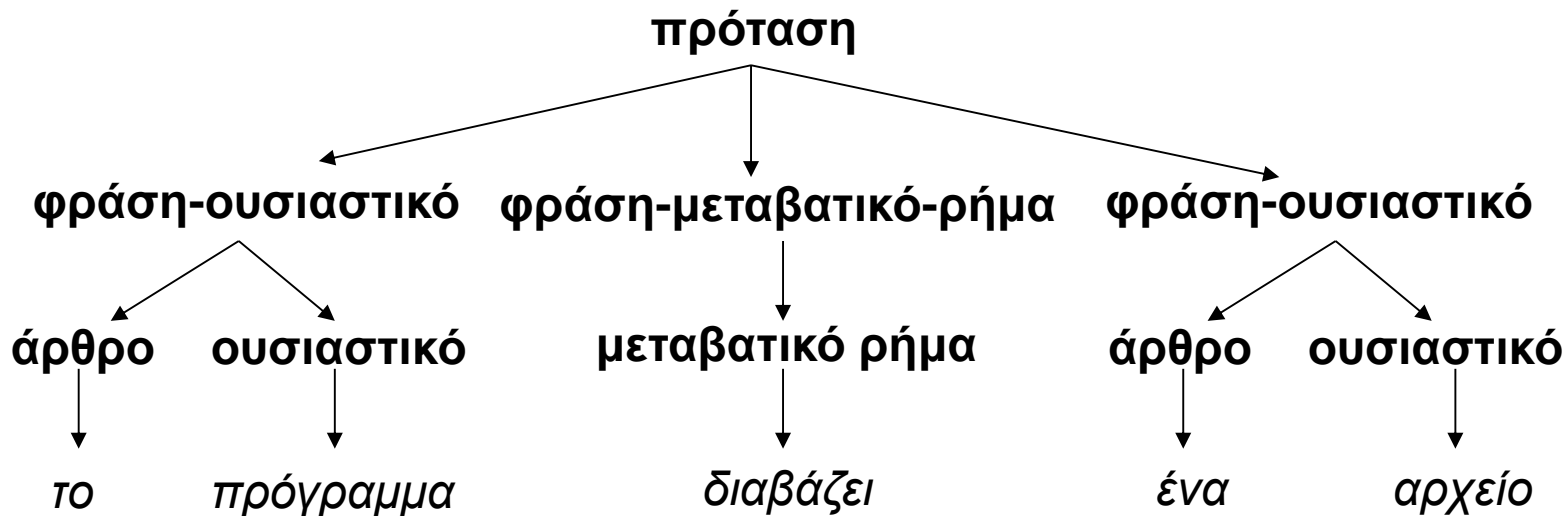
Παράδειγμα : «το πρόγραμμα διαβάζει ένα αρχείο»



Περιγραφή γλώσσας μέσω γραμματικής (6/9)

Παραγωγή προτάσεων στα ελληνικά

Παράδειγμα : «το πρόγραμμα διαβάζει ένα αρχείο»



Περιγραφή

γλώσσας μέσω γραμματικής (7/9)

- Παραγωγή προτάσεων στα ελληνικά.
 - Παράδειγμα: «ο υπολογιστής τρέχει γρήγορα».



Περιγραφή γλώσσας μέσω γραμματικής (8/9)

Παραγωγή προτάσεων στα ελληνικά

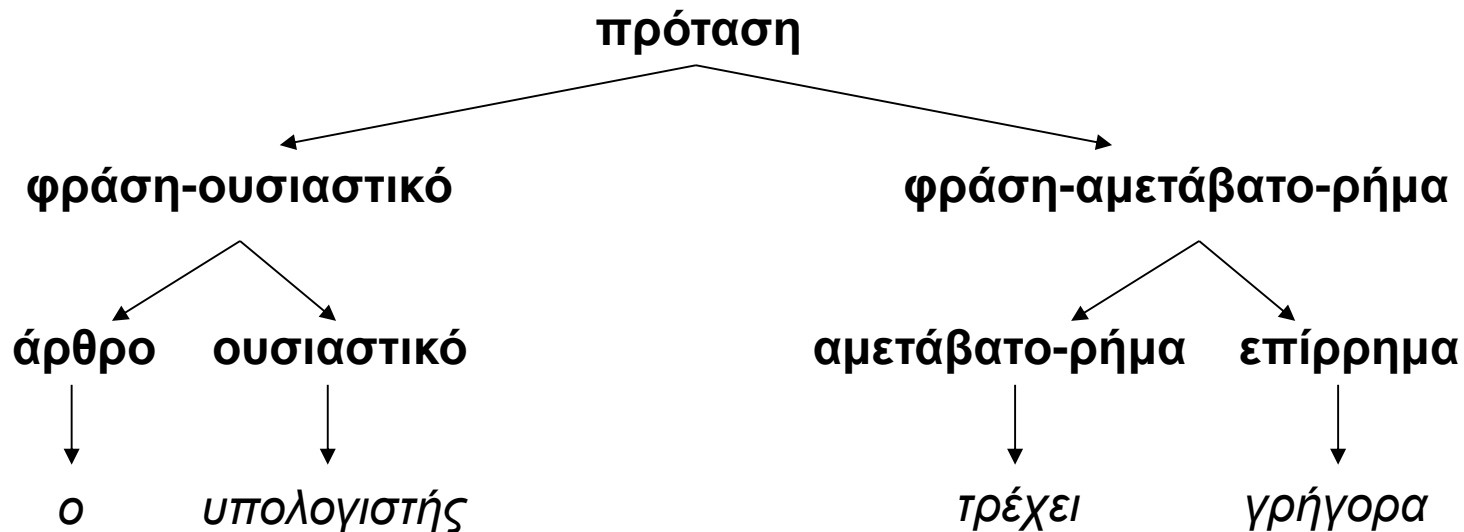
Παράδειγμα : «ο υπολογιστής τρέχει γρήγορα»



Περιγραφή γλώσσας μέσω γραμματικής (9/9)

Παραγωγή προτάσεων στα ελληνικά

Παράδειγμα : «ο υπολογιστής τρέχει γρήγορα»



Γραμματικές Δομής Φράσεως (1/4)

Έστω ο ακόλουθος συμβολισμός:

πρόταση → φράση-ουσιαστικό φράση-μεταβατικό-ρήμα φράση-ουσιαστικό

πρόταση → φράση-ουσιαστικό φράση-αμετάβατο-ρήμα

φράση-ουσιαστικό → άρθρο ουσιαστικό

φράση-ουσιαστικό → ουσιαστικό

φράση-μεταβατικό-ρήμα → μεταβατικό-ρήμα

φράση-αμετάβατο-ρήμα → φράση-αμετάβατο-ρήμα επίρρημα

φράση-αμετάβατο-ρήμα → αμετάβατο-ρήμα

άρθρο → *μια*

άρθρο → *ο*

ουσιαστικό → *σκύλος*

ουσιαστικό → *γάτα*

μεταβατικό-ρήμα → *κυνηγά*

μεταβατικό-ρήμα → *συναντά*

αμετάβατο-ρήμα → *τρέχει*

επίρρημα → *αργά*

επίρρημα → *γρήγορα*



Γραμματικές Δομής Φράσεως (2/4)

- Μερικές προτάσεις της γλώσσας είναι:
 - ο σκύλος συναντά μια γάτα,
 - σκύλος κυνηγά γάτα,
 - μια γάτα τρέχει αργά,
 - **ο γάτα κυνηγά μια σκύλος.**
- Μια **γραμματική δομής φράσεων** αποτελείται από 4 στοιχεία:
 - Ένα σύνολο **τερματικών** συμβόλων T .
 - Ένα σύνολο **μη-τερματικών** συμβόλων N .
 - Ένα σύνολο **παραγωγών** P .
 - Ανάμεσα σε όλα τα μη-τερματικά σύμβολα του N , υπάρχει ένα ειδικό σύμβολο που ονομάζεται **αρχικό σύμβολο**.



Γραμματικές Δομής Φράσεως (3/4)

- Τα τερματικά σύμβολα στο T είναι σύμβολα που χρησιμοποιούνται για να δημιουργηθούν οι προτάσεις της γλώσσας.
 - {μια,ο,σκύλος,γάτα,κυνηγά,συναντά,τρέχει,αργά,γρήγορα}
- Τα μη-τερματικά σύμβολα στο N είναι ενδιάμεσα σύμβολα που χρησιμοποιούνται για να περιγραφεί η δομή των προτάσεων:
 - {πρόταση,φράση-ουσιαστικό,φράση-μεταβατικό-ρήμα,φράση-αμετάβατο-ρήμα,άρθρο,ουσιαστικό, μεταβατικό-ρήμα, αμετάβατο,επίρρημα}
- Οι παραγωγές είναι γραμματικοί κανόνες που καθορίζουν πως μπορούν να σχηματιστούν προτάσεις της γλώσσας.



Γραμματικές Δομής Φράσεως (4/4)

- Μια παραγωγή είναι της μορφής $\alpha \rightarrow \beta$, όπου τα α και β είναι συμβολοσειρές τερματικών και μη-τερματικών συμβόλων. Μια παραγωγή καθορίζει ότι η συμβολοσειρά α μπορεί να μετατραπεί στη συμβολοσειρά β .
 - πρόταση \rightarrow φράση-ουσιαστικό φράση-μεταβατικό-ρήμα φράση-ουσιαστικό
πρόταση \rightarrow φράση-ουσιαστικό φράση-αμετάβατο-ρήμα
κτλ.
- Το αρχικό σύμβολο είναι ένα ειδικό μη τερματικό σύμβολο το οποίο αρχίζει τη δημιουργία οποιασδήποτε πρότασης στη γλώσσα.
 - στο παράδειγμα είναι η **πρόταση**.



Δημιουργία Προτάσεων (1/3)

- Εφόσον έχει δοθεί μια γραμματική μπορούμε να δημιουργήσουμε προτάσεις της γλώσσας ως εξής:
 - Αρχίζουμε με το αρχικό σύμβολο ως **τρέχουσα συμβολοσειρά** τερματικών και μη τερματικών συμβόλων.
 - Αν κάποιο τμήμα της τρέχουσας συμβολοσειράς τερματικών και μη τερματικών συμβόλων ταιριάζει με το αριστερό τμήμα μιας παραγωγής, αντικαθιστούμε το τμήμα αυτό της συμβολοσειράς με το δεξιό τμήμα της παραγωγής.

Έστω η τρέχουσα συμβολοσειρά **a** που αποτελείται από τα τμήματα **abc** και η παραγωγή **b** → **d**
Μπορούμε να δημιουργήσουμε τη συμβολοσειρά **adc**



Δημιουργία Προτάσεων (2/3)

- Ο συμβολισμός $\alpha \Rightarrow \text{adc}$ υποδηλώνει ότι μπορεί η συμβολοσειρά α να μετασχηματιστεί στην adc χρησιμοποιώντας παραγωγές της γραμματικής.
- Οποιαδήποτε συμβολοσειρά τερματικών συμβόλων που λαμβάνεται μετά από έναν αριθμό επαναλήψεων της παραπάνω διαδικασίας είναι μια πρόταση της γλώσσας.
- Υπάρχει περίπτωση στην τρέχουσα συμβολοσειρά να μπορούμε να εφαρμόσουμε πάνω από μια παραγωγές.
- Επίσης μπορεί να καταλήξουμε σε συμβολοσειρά η οποία περιέχει μη τερματικά σύμβολα και στην οποία δε μπορεί να εφαρμοστεί καμία παραγωγή.
 - **αδιέξοδο.**



Δημιουργία Προτάσεων (3/3)

- Η πρόταση “ο σκύλος τρέχει γρήγορα” παράγεται ως εξής:

πρόταση ⇒ φράση-ουσιαστικό φράση-αμετάβατο-ρήμα
⇒ φράση-ουσιαστικό αμετάβατο-ρήμα επίρρημα
⇒ φράση-ουσιαστικό αμετάβατο-ρήμα *γρήγορα*
⇒ φράση-ουσιαστικό *τρέχει γρήγορα*
⇒ άρθρο ουσιαστικό *τρέχει γρήγορα*
⇒ άρθρο σκύλος *τρέχει γρήγορα*
⇒ ο σκύλος *τρέχει γρήγορα*



Γραμματικές Δομής Φράσεως - Παράδειγμα

- Θέλουμε να κατασκευάσουμε μια γραμματική για τη γλώσσα:
 $L = \{aaaa, aabb, bbaa, bbbb\}$.
- Μια και υπάρχει πεπερασμένο πλήθος συμβολοσειρών μπορούμε απλά να τις καταγράψουμε:
 - $T = \{a,b\}$ το σύνολο των τερματικών συμβόλων.
 - $N = \{S\}$ το σύνολο των μη τερματικών συμβόλων.
 - Το σύνολο των παραγωγών: $\{S \rightarrow aaaa, S \rightarrow aabb, S \rightarrow bbaa, S \rightarrow bbbb\}$.
- Μπορούμε να απλουστεύσουμε τη γραμματική:
 - $N = \{S,A\}$ το σύνολο των μη τερματικών συμβόλων όπου S το αρχικό σύμβολο.
 - Το σύνολο των παραγωγών: $\{S \rightarrow AA, A \rightarrow aa, A \rightarrow bb\}$.



Γλώσσες – Παράδειγμα 1 (1/2)

Παράδειγμα $L = \{\alpha^i \beta^{2i} \mid i \geq 1\}$

Έχουμε $T = \{\alpha, \beta\}$
 $N = \{S\}$

και τις παραγωγές

$$S \rightarrow \alpha S \beta \beta$$

$$S \rightarrow \alpha \beta \beta$$

Παράδειγμα

$$S \Rightarrow \alpha S \beta \beta \Rightarrow \alpha \alpha S \beta \beta \beta \beta \Rightarrow \alpha \alpha \alpha \beta \beta \beta \beta \beta \beta$$



Γλώσσες – Παράδειγμα 1 (2/2)

Παράδειγμα $L = \{\alpha^i \beta^j \mid 1 \leq i < j\}$

Έχουμε $T = \{\alpha, \beta\}$

$N = \{S, B\}$

και τις παραγωγές

$S \rightarrow \alpha S \beta$

$S \rightarrow \alpha B \beta$

$B \rightarrow \beta B$

$B \rightarrow \beta$

Παράδειγμα

$S \Rightarrow \alpha S \beta \Rightarrow \alpha \alpha B \beta \beta \Rightarrow \alpha \alpha \beta B \beta \beta \Rightarrow \alpha \alpha \beta \beta B \beta \beta \Rightarrow \alpha \alpha \beta \beta \beta \beta \beta$

Πως θα δημιουργήσουμε τις συμβολοσειρές $\alpha\beta\beta\beta$, $\alpha\alpha\beta\beta\beta\beta$, $\alpha\alpha\alpha\beta\beta\beta\beta$;



Γραμματικές δομής φράσεως – παράδειγμα

- Θέλουμε να κατασκευάσουμε μια γραμματική για τη γλώσσα:
 $L = \{x \mid x \in \{a,b\}^*, \text{ ο αριθμός των } a \text{ στο } x \text{ είναι πολλαπλάσιος του } 3\}$.
- $T = \{a,b\}$ το σύνολο των τερματικών συμβόλων.
- $N = \{S,A,B\}$ το σύνολο των μη τερματικών συμβόλων όπου S το αρχικό σύμβολο.
- Το σύνολο των παραγωγών είναι:
 - $S \rightarrow bS$
 - $S \rightarrow b$
 - $S \rightarrow aA$
 - $A \rightarrow bA$
 - $A \rightarrow aB$
 - $A \rightarrow bB$
 - $B \rightarrow aS$
 - $B \rightarrow a$

• $S \Rightarrow bS \Rightarrow bbS \Rightarrow bbaA \Rightarrow bbabA \Rightarrow$
 $bbabaB \Rightarrow bbababB \Rightarrow bbababbB \Rightarrow$
 $bbababbaS \Rightarrow bbababbab$



Γλώσσες – Παράδειγμα 2 (1/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$



Γλώσσες – Παράδειγμα 2 (2/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

$\text{asgn_stat} \rightarrow \text{id} = \text{exp}$

$\text{id} \rightarrow A$

$\text{exp} \rightarrow \text{exp} + \text{term}$

$\text{id} \rightarrow B$

$\text{exp} \rightarrow \text{term}$

$\text{id} \rightarrow C$

$\text{term} \rightarrow \text{term} * \text{factor}$

$\text{id} \rightarrow D$

$\text{term} \rightarrow \text{factor}$

$\text{factor} \rightarrow (\text{exp})$

$\text{factor} \rightarrow \text{id}$



Γλώσσες – Παράδειγμα 2 (3/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

$\text{asgn_stat} \rightarrow \text{id} = \text{exp}$

$\text{id} \rightarrow A$

$\text{exp} \rightarrow \text{exp} + \text{term}$

$\text{id} \rightarrow B$

$\text{exp} \rightarrow \text{term}$

$\text{id} \rightarrow C$

$\text{term} \rightarrow \text{term} * \text{factor}$

$\text{id} \rightarrow D$

$\text{term} \rightarrow \text{factor}$

$\text{factor} \rightarrow (\text{exp})$

π.χ. $A = (B + D) * (C + D)$

$\text{factor} \rightarrow \text{id}$



Γλώσσες – Παράδειγμα 2 (4/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

$\text{asgn_stat} \rightarrow \text{id} = \text{exp}$

$\text{id} \rightarrow A$

$\text{exp} \rightarrow \text{exp} + \text{term}$

$\text{id} \rightarrow B$

$\text{exp} \rightarrow \text{term}$

$\text{id} \rightarrow C$

$\text{term} \rightarrow \text{term} * \text{factor}$

$\text{id} \rightarrow D$

$\text{term} \rightarrow \text{factor}$

π.χ. $A = (B + D) * (C + D)$

$\text{factor} \rightarrow (\text{exp})$

$\text{factor} \rightarrow \text{id}$

$\text{asgn_stat} \rightarrow \text{id} = \text{exp}$



Γλώσσες – Παράδειγμα 2 (5/7)

Παράδειγμα Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

π.χ. $A = (B + D) * (C + D)$

asgn_stat \rightarrow **id = exp**

\rightarrow **id = term** \rightarrow **id = term * factor** \rightarrow **id = term * (exp)**

\rightarrow **id = term * (exp + term)**

\rightarrow **id = term * (exp + factor)**

\rightarrow **id = term * (exp + id)**

\rightarrow **id = term * (exp + D)**

\rightarrow **id = term * (term + D)**



Γλώσσες – Παράδειγμα 2 (6/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

π.χ. $A = (B + D) * (C + D)$

→ **id** = **term** * (**term** + D)

→ **id** = **term** * (**factor** + D)

→ **id** = **term** * (**id** + D)

→ **id** = **term** * ($C + D$)

→ **id** = **factor** * ($C + D$)

→ **id** = (**exp**) * ($C + D$)



Γλώσσες – Παράδειγμα 2 (7/7)

Θέλουμε να κατασκευάσουμε αριθμητικές παραστάσεις με

σύμβολα $T = \{A, B, C, D, +, *, (,), =\}$

$N = \{\text{asgn_stat}, \text{exp}, \text{term}, \text{factor}, \text{id}\}$

π.χ. $A = (B + D) * (C + D)$

→ **id** = (**exp**) * ($C + D$)

→ **id** = (**exp** + **term**) * ($C + D$)

→ **id** = (**exp** + **factor**) * ($C + D$)

→ **id** = (**exp** + **id**) * ($C + D$)

→ **id** = (**exp** + D) * ($C + D$)

→ ... → $A = (B + D) * (C + D)$



Γλώσσες – Παράδειγμα 3 (1/2)

Παράδειγμα Θέλουμε γραμματική για τη γλώσσα

$$L = \{\{\alpha, \beta\}^i \mid i \geq 1\}$$

Έχουμε $T = \{\alpha, \beta\}$

$$N = \{S\}$$

και τις παραγωγές

$$S \rightarrow \alpha S$$

$$S \rightarrow \beta S$$

$$S \rightarrow \alpha$$

$$S \rightarrow \beta$$

Παράδειγμα

$$S \Rightarrow \beta S \Rightarrow \beta \alpha S \Rightarrow \beta \alpha \alpha S \Rightarrow \beta \alpha \alpha \beta S \Rightarrow \beta \alpha \alpha \beta \beta S \Rightarrow \beta \alpha \alpha \beta \beta \beta$$



Γλώσσες – Παράδειγμα 3 (2/2)

Παράδειγμα Θέλουμε γραμματική για τη γλώσσα

$$L = \{\alpha^i \beta^j \gamma^j \mid i, j \geq 1\}$$

Έχουμε $T = \{\alpha, \beta, \gamma\}$

$$N = \{S, B\}$$

και τις παραγωγές

$$S \rightarrow \alpha S$$

$$S \rightarrow B$$

$$B \rightarrow \beta B \gamma$$

$$B \rightarrow \beta \gamma$$

Παράδειγμα

$$S \Rightarrow \alpha S \Rightarrow \alpha \alpha S \Rightarrow \alpha \alpha B \Rightarrow \alpha \alpha \beta B \gamma \Rightarrow \alpha \alpha \beta \beta B \gamma \gamma \Rightarrow \alpha \alpha \beta \beta \beta \gamma \gamma \gamma$$



Τύποι Γραμματικών και Γλωσσών (1/4)

- Η χρήση γραμματικών για την περιγραφή γλωσσών οδηγεί σε έναν τρόπο κατηγοριοποίησης των γλωσσών. Έστω:
 - A και B μη τερματικά σύμβολα.
 - a και b τερματικά σύμβολα.
 - α και β συμβολοσειρές τερματικών και μη τερματικών συμβόλων.
- Μια γραμματική ονομάζεται **γραμματική τύπου 3**, αν όλες οι παραγωγές της είναι της μορφής.

$$A \rightarrow a$$

$$A \rightarrow aB$$

Ή ισοδύναμα

$$A \rightarrow a$$

$$A \rightarrow Ba$$



Τύποι Γραμματικών και Γλωσσών (2/4)

- Μια γραμματική ονομάζεται **γραμματική τύπου 2**, αν όλες οι παραγωγές της είναι της μορφής.

$$A \rightarrow \alpha$$

- Αυτό σημαίνει ότι οι γραμματικές τύπου 2 έχουν παραγωγές όπου το αριστερό μέρος αποτελείται πάντα από ένα μόνο μη τερματικό σύμβολο.

- μια γραμματική τύπου 3 είναι και τύπου 2.

- Μια γραμματική ονομάζεται **γραμματική τύπου 1**, αν για κάθε παραγωγή της μορφής:

$$\alpha \rightarrow \beta$$

το μήκος του β είναι μεγαλύτερο ή ίσο με το μήκος του α

- π.χ. Οι παραγωγές $ABc \rightarrow bc$ και $Aa \rightarrow a$ δεν ικανοποιούν τον περιορισμό.



Τύποι Γραμματικών και Γλωσσών (3/4)

- Μια γραμματική ονομάζεται **γραμματική τύπου 0**, αν δεν έχει κανέναν περιορισμό
- Μια γλώσσα ονομάζεται **γλώσσα τύπου i** ($i=0,1,2,3$) αν μπορεί να καθοριστεί από μια γραμματική τύπου i , αλλά δε μπορεί να καθοριστεί από μια γραμματική τύπου $i+1$
 - π.χ. Η γλώσσα $L = \{a^k b^k \mid k \geq 1\}$ είναι μια γλώσσα τύπου 2, γιατί μπορεί να καθοριστεί από τη γραμματική τύπου 2.

$$A \rightarrow aAb$$

$$A \rightarrow ab$$

αλλά δε μπορεί να καθοριστεί από μια γραμματική τύπου 3. Αυτό μπορεί να αποδειχθεί χρησιμοποιώντας ΜΠΚ, όπως θα δούμε παρακάτω.



Τύποι Γραμματικών και Γλωσσών (4/4)

- Υπάρχουν γλώσσες που δεν είναι τύπου 0;
 - Ναι, υπάρχουν γλώσσες που δεν μπορούν να καθοριστούν από μια γραμματική δομής φράσεως.
- Οι γλώσσες προγραμματισμού μπορούν να καθοριστούν;
 - Ναι, οι περισσότερες (σχεδόν όλες) είναι τύπου 2.
- Πως μπορούμε να καθορίσουμε αν μια δεδομένη συμβολοσειρά ανήκει σε μια γλώσσα η οποία καθορίζεται από μια γραμματική;
 - Το πρόβλημα αυτό είναι πολύ σημαντικό στην κατασκευή μεταφραστών/μεταγλωττιστών για γλώσσες προγραμματισμού.
 - Είναι δύσκολο υπολογιστικά, αλλά υπάρχουν αποδοτικοί αλγόριθμοι.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Διακριτά Μαθηματικά». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.uowm.gr/courses/ICTE257/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Όχι Παράγωγα Έργα Μη Εμπορική Χρήση 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως Μη Εμπορική ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους
υπερσυνδέσμους.

