



Συμβολικές Γλώσσες Προγραμματισμού

Ενότητα 8: Συναρτησιακός Προγραμματισμός

Νικόλαος Καραμπετάκης

Τμήμα Μαθηματικών



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 Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την 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.



Περιεχόμενα

1. Εφαρμογή συναρτήσεων σε τιμές.
2. Επίπεδα μιας έκφρασης.
3. Οι εντολές Map και Through.
4. Η εντολή Apply.
5. Η εντολή Inner.
6. Η εντολή Outer.
7. Επαναλήψεις μέσω της Nest και Fold.



Σκοποί Ενότητας

- Μελέτη του συναρτησιακού προγραμματισμού.



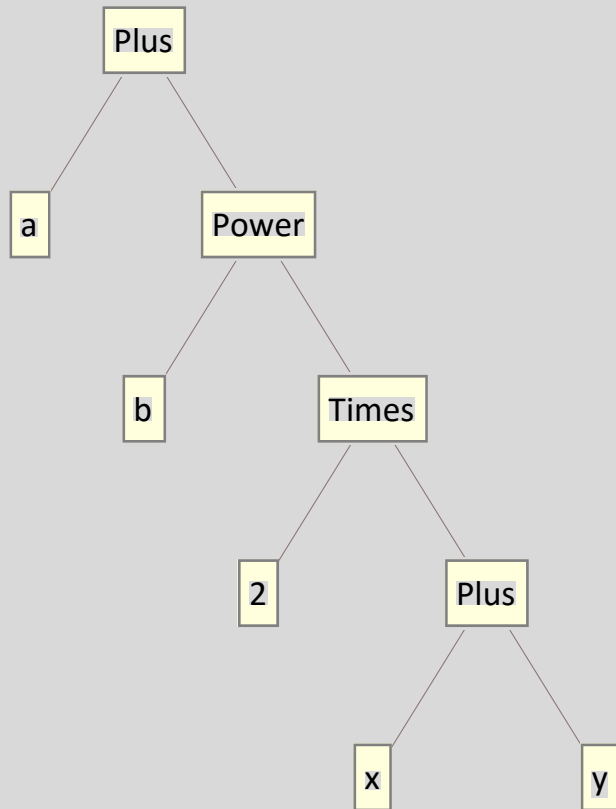
Συναρτησιακός προγραμματισμός

Με τον όρο γλώσσα συναρτησιακού προγραμματισμού εννοούμε την γλώσσα στην οποία συναρτήσεις μπορούν να οριστούν και να εφαρμοστούν σε ορίσματα, τα οποία με τη σειρά τους μπορεί να είναι και αυτά συναρτήσεις Κ.Ο.Κ.



Επίπεδα μιας έκφρασης (1)

TreeForm [a + b² (x+y)]



Επίπεδα μιας έκφρασης (2)

Στο επίπεδο 1 έχουμε

$$\text{Level}[a + b^{2(x+y)}, \{1\}]$$

$$\{a, b^{2(x+y)}\}$$

Στο επίπεδο 2 έχουμε

$$\text{Level}[a + b^{2(x+y)}, \{2\}]$$

$$\{b, 2(x+y)\}$$

ενώ στα επίπεδα 1 και 2

$$\text{Level}[a + b^{2(x+y)}, 2]$$

$$\{a, b, 2(x+y), b^{2(x+y)}\}$$

Κ.Ο.Κ.



Άσκηση 1.1 (Επίπεδα μιας έκφρασης)

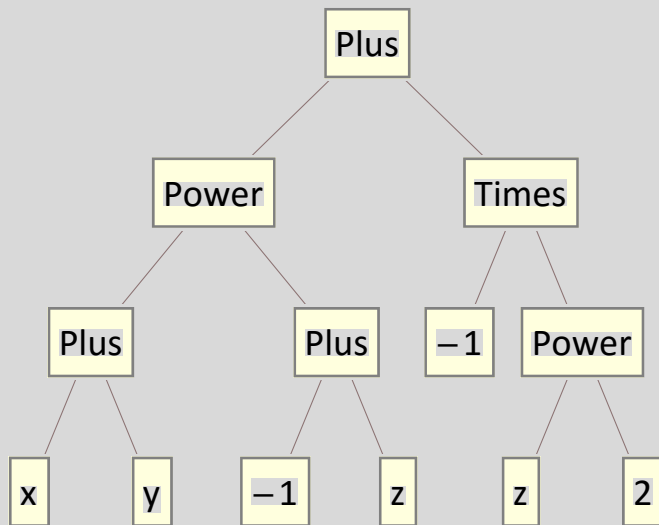
Άσκηση 1. Προσδιορίστε το δέντρο της έκφρασης

$$(x + y)^{z-1} - z^2$$

καθώς και τα στοιχεία του επιπέδου 2.

Λύση άσκησης

TreeForm $[(x + y)^{z-1} - z^2]$



Άσκηση 1.2 (Επίπεδα μιας έκφρασης)

$\text{Level}[(x + y)^{z-1} - z^2, \{2\}]$

$\{x + y, -1 + z, -1, z^2\}$



Οι εντολές Map και Through (1)

? Map

Η συνάρτηση Map συνήθως συντάσσεται με μια συνάρτηση συνοδευόμενη από μια λίστα. Η Map εφαρμόζει την συνάρτηση f σε κάθε στοιχείο της λίστας $expr$.

Map[Cos, {0, Pi/4, Pi/3, Pi/2}]

$\{1, \frac{1}{\sqrt{2}}, \frac{1}{2}, 0\}$

Παρατηρήστε ότι δεν γράφω $f[x]$ αλλά σκέτο f . Η λίστα $expr$ μπορεί να αντικατασταθεί και από μια έκφραση όπως $a + b^{2(x+y)}$:

Map[Cos, $a + b^{2(x+y)}$, {1}]

$\text{Cos}[a] + \text{Cos}[b^{2(x+y)}]$



Οι εντολές Map και Through (2)

οπότε η Map εφαρμόζει την συνάρτηση στο επίπεδο 1 της έκφρασης ή

$$\text{Map}[\text{Cos}, a + b^{2(x+y)}, \{2\}]$$
$$a + \text{Cos}[b]^{\text{Cos}[2(x+y)]}$$

που εφαρμόζει την συνάρτηση στο επίπεδο 2 της έκφρασης ή

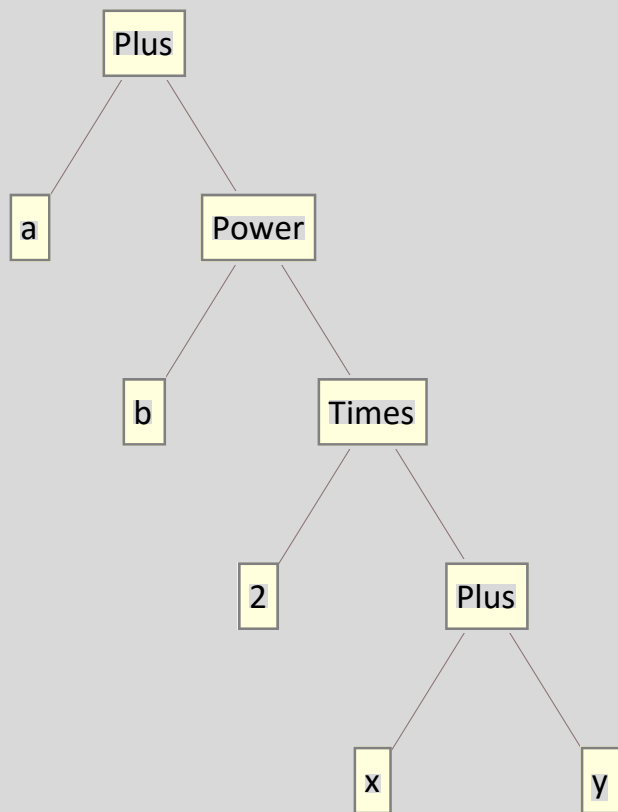
$$\text{Map}[\text{Cos}, a + b^{2(x+y)}, \{3\}]$$
$$a + b^{\text{Cos}[2] \text{Cos}[x+y]}$$

που εφαρμόζει την συνάρτηση στο επίπεδο 2 της έκφρασης. Παρακάτω βλέπουμε το δένδρο της έκφρασης



Οι εντολές Map και Through (3)

TreeForm[$a + b^{2(x+y)}$]



Οι εντολές Map και Through (4)

Η Map μπορεί να εφαρμοσθεί σε κάθε υποέκφραση της $a + b^{2(x+y)}$

```
MapAll[Cos, a + b2(x+y)]
```

```
Cos[Cos[a] + Cos[Cos[b]Cos[Cos[2] Cos[Cos[x]+Cos[y]]]]]
```



Οι εντολές Map και Through (5)

Αν θέλω αντίθετα από την Map να εφαρμόσω παραπάνω από μια συναρτήσεις σε μια τιμή

```
Through[{Sin, Cos, Tan} [Pi/2]]
```

```
{1, 0, ComplexInfinity}
```

? Through

```
Through[p[Subscript[f, 1], Subscript[f, 2]][x]] gives  
p[Subscript[f, 1][x], Subscript[f, 2][x]].Through[expr, h] performs the transformation wherever h occurs in the head of expr. >>
```

Δες επίσης MapAt, MapThread, MapIndexed



Άσκηση 2.1 (Οι εντολές Map και Through)

Άσκηση 2.

- α) Η εντολή `CharactersRange` μας βοηθάει στο να δημιουργήσουμε μια λίστα με τα γράμματα του αγγλικού αλφαβήτου. Δημιουργήστε μια λίστα `a` με όλους τους χαρακτήρες του αγγλικού αλφαβήτου.
- β) Με τη βοήθεια της `Map` και της `ToCharacterCode` να υπολογίσετε τους κωδικούς αριθμούς ASCII των παραπάνω γραμμάτων.
- γ) Στη συνέχεια να ορίσετε την συνάρτηση $f(x) = [(5x+4) \bmod 26] + 65$ και να την εφαρμόσετε στα στοιχεία της τελευταίας λίστας.
- δ) Τέλος να μετατρέψετε τους αριθμούς που προέκυψαν σε χαρακτήρες με χρήση της συνάρτησης `FromCharacterCode`.



Άσκηση 2.2 (Οι εντολές Map και Through)

```
a = CharacterRange["A", "Z"]
```

```
{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}
```

```
a = Map[ToCharacterCode, a]
```

```
{{65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75}, {76}, {77},  
{78}, {79}, {80}, {81}, {82}, {83}, {84}, {85}, {86}, {87}, {88}, {89}, {90}}
```

και αν χρησιμοποιήσουμε την Flatten

```
f[x_] := Mod[5 x + 4, 26] + 65
```

```
a = Map[f, a]
```

```
{{82}, {87}, {66}, {71}, {76}, {81}, {86}, {65}, {70}, {75}, {80}, {85}, {90},  
{69}, {74}, {79}, {84}, {89}, {68}, {73}, {78}, {83}, {88}, {67}, {72}, {77}}
```



Άσκηση 2.3 (Οι εντολές Map και Through)

```
a = Map[FromCharCode, a]
```

```
{R, W, B, G, L, Q, V, A, F, K, P, U, Z, E, J, O, T, Y, D, I, N, S, X, C, H, M}
```

Συνοπτικά θα μπορούσαμε να γράψουμε

```
Map [ FromCharCode , Map [ Mod [ 5 #1 + 4 , 26 ] + 65 & ,  
      Map [ ToCharCode , CharacterRange [ "A" , "Z" ] ] ] ]
```

```
{R, W, B, G, L, Q, V, A, F, K, P, U,  
Z, E, J, O, T, Y, D, I, N, S, X, C, H, M}
```

```
Map [ FromCharCode ,  
      Map [ Function [ x , Mod [ 5 x + 4 , 26 ] + 65 ] ,  
      Map [ ToCharCode , CharacterRange [ "A" , "Z" ] ] ] ]
```

```
{R, W, B, G, L, Q, V, A, F, K, P, U,  
Z, E, J, O, T, Y, D, I, N, S, X, C, H, M}
```



Η εντολή Apply (1)

? Apply

Η Apply αντικαθιστά την κεφαλή της έκφρασης expr με την συνάρτηση f.

Η κεφαλή της έκφρασης x+y είναι

Head[x + y + z]

Plus

η οποία μπορεί να αντικατασταθεί με την Times όπως φαίνεται παρακάτω

Apply[Times, x + y + z]

x y z



Η εντολή Apply (2)

Αν θέλουμε να υπολογίσουμε το άθροισμα $1+2+3+\dots+10$, πρώτα δημιουργούμε την λίστα

```
Table[i, {i, 1, 10}]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

και στη συνέχεια εφαρμόζουμε την Apply με πρώτο όρισμα το Plus

```
Apply[Plus, %]
```

```
55
```

ή αν θέλουμε να υπολογίσουμε το $10!=1*2*3*\dots*10$ όμοια έχουμε

```
Apply[Times, Table[i, {i, 1, 10}]]
```

```
3 628 800
```



Η εντολή Apply (3)

Ας δούμε και ένα παράδειγμα με περισσότερα επίπεδα

```
FullForm[x + y2(z-1)]
```

```
Plus[x, Power[y, Times[2, Plus[-1, z]]]]
```

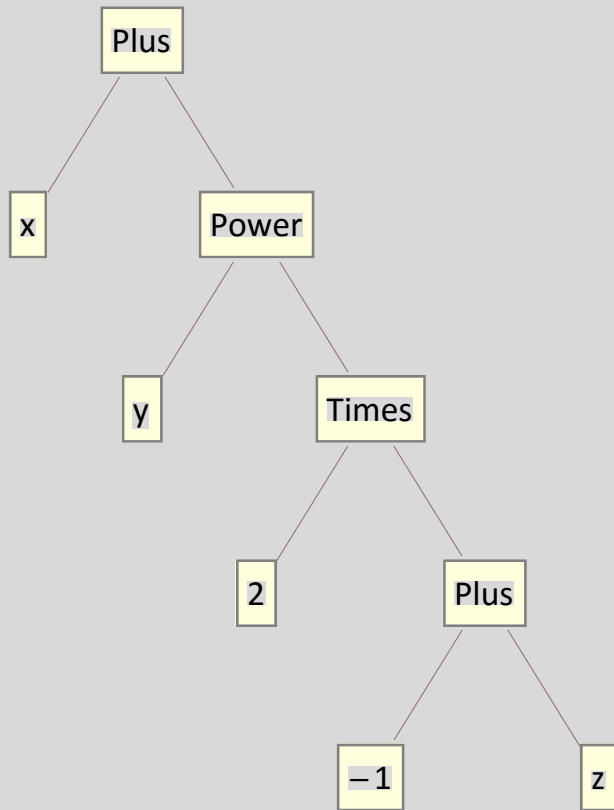
```
Apply[Plus, x + y2(z-1), {1}]
```

```
x + y + 2(-1 + z)
```



Η εντολή Apply (4)

TreeForm[$x + y^2(z-1)$]



Η εντολή Apply (5)

Άσκηση. Να δημιουργήσετε μια λίστα με τους 10 πρώτους αριθμούς. Στη συνέχεια κάνοντας χρήση της Apply να υπολογίσετε το άθροισμα των αριθμών αυτών.

```
a = Table[Prime[i], {i, 1, 10}]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
```

```
Apply[Plus, a]
```

```
129
```

ή αν συνθέσω τις δύο συναρτήσεις

```
Apply[Plus, Table[Prime[i], {i, 1, 10}]]
```

```
129
```



Η εντολή Dot - Η εντολή Inner

? Dot

a.b.c or Dot[a, b, c] gives products of vectors, matrices and tensors. >>

Dot[{x, y}, {z, w}]

w y + x z

Η Dot όπως βλέπουμε παρακάτω μας δίνει το εσωτερικό γινόμενο δύο διανυσμάτων. Πολλαπλασιάζει δηλαδή τις αντίστοιχες συντεταγμένες και στην συνέχεια τις προσθέτει. Αν αντί για τον πολλαπλασιασμό και την πρόσθεση θέλαμε δύο άλλες πράξεις οι οποίες δηλώνονται μέσω συναρτήσεων f και g θα είχαμε την συνάρτηση Inner

? Inner

Inner[f, Subscript[list, 1], Subscript[list, 2], g] is a generalization of Dot in which f plays the role of multiplication and g of addition. >>



Η εντολή Inner

```
Inner[f, {x, y}, {z, w}, g]
```

```
g[f[x, z], f[y, w]]
```

Η παρακάτω εντολή είναι όμοια με την Dot

```
Inner[Times, {x, y}, {z, w}, Plus]
```

```
w y + x z
```

Η παρακάτω πράξη θυμίζει πολλαπλασιασμό πινάκων, μόνο που ο πολλαπλασιασμός έχει αντικατασταθεί από την f και η πρόσθεση από την g.

```
Inner[f, {{x11, x12}, {x21, x22}}, {{y11, y12}, {y21, y22}}, g] // MatrixForm
```

```
( g[f[x11, y11], f[x12, y21]]  g[f[x11, y12], f[x12, y22]] )  
 ( g[f[x21, y11], f[x22, y21]]  g[f[x21, y12], f[x22, y22]] )
```



Άσκηση 3.1

Άσκηση 3. Να ορίσετε την συνάρτηση g για την πρόσθεση δύο δυαδικών μεταβλητών $\{0+0=0,0+1=1,1+0=0,1+1=1\}$ και την συνάρτηση f για τον πολλαπλασιασμό δυαδικών μεταβλητών $\{0*0=0,0*1=0,1*0=0,1*1=1\}$ και στη συνέχεια να υπολογίσετε με την βοήθεια της `Inner` το γινόμενο δύο δυαδικών πινάκων.

```
f[x_, y_] := Switch[{x, y},  
  {0, 0}, 0,  
  {0, 1}, 0,  
  {1, 0}, 0,  
  {1, 1}, 1]
```



Άσκηση 3.2

```
g[x_, y_] := Switch[{x, y},  
  {0, 0}, 0,  
  {0, 1}, 1,  
  {1, 0}, 1,  
  {1, 1}, 1]
```

```
Inner[f, {{0, 1}, {1, 0}}, {{1, 1}, {0, 0}}, g] // MatrixForm
```

$$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$

ή τα παραπάνω μπορώ να τα συνοψίσω σε μια συνάρτηση



Άσκηση 3.3

```
matrixProduct[x_, y_] := Inner[
  Function[{x1, y1}, Switch[{x1, y1}, {0, 0}, 0, {0, 1}, 0, {1, 0}, 0, {1, 1}, 1]],
  x, y,
  Function[{z1, w1}, Switch[{z1, w1}, {0, 0}, 0, {0, 1}, 1, {1, 0}, 1, {1, 1}, 1]] //
  MatrixForm
```

```
matrixProduct[{{0, 1}, {1, 0}}, {{1, 1}, {0, 0}}]
```

$$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$



Η εντολή Outer (1)

? Outer

$$w \otimes v = wv^T = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} w_1 v_1 & w_1 v_2 & w_1 v_3 \\ w_2 v_1 & w_2 v_2 & w_2 v_3 \\ w_3 v_1 & w_3 v_2 & w_3 v_3 \\ w_4 v_1 & w_4 v_2 & w_4 v_3 \end{bmatrix}$$

Η Outer αποτελεί μια γενίκευση της Map για συναρτήσεις με περισσότερες από μια μεταβλητές, παίρνει δηλαδή όλους τους δυνατούς συνδυασμούς των δύο λιστών από τις οποίες συνοδεύεται και εφαρμόζει την συνάρτηση f στα ορίσματα που προκύπτουν.

```
Clear[f, a, b, c, d]
```

```
Outer[f, {a, b}, {c, d}]
```

```
{{f[a, c], f[a, d]}, {f[b, c], f[b, d]}}
```



Η εντολή Outer (2)

```
Outer[Integrate, {Cos[x], Sin[y]}, {x, y}]
```

```
{{Sin[x], y Cos[x]}, {x Sin[y], -Cos[y]}}
```

Άσκηση. Να εφαρμόσετε την Outer στην συνάρτηση της παραγώγου για μια συνάρτηση με δύο μεταβλητές π.χ. $f(x, y) = x^2 + y^2$,

```
Outer[D, {x^2 + y^2, x^2 - y^2}, {x, y}]
```

```
{{2 x, 2 y}, {2 x, -2 y}}
```

Αν ψάχναμε τον Hessian πίνακα της συνάρτησης $x^2 + y^2$ τότε θα ακολουθούσαμε την παρακάτω μέθοδο

```
Outer[D, {x^2 + y^2}, {x, y}]
```



Η εντολή Outer (3)

```
Outer[D, {2 x, 2 y}, {x, y}]
```

```
{{2, 0}, {0, 2}}
```

ή ισοδύναμα

```
Outer[D, Outer[D, {x2 + y2}, {x, y}][[1]], {x, y}]
```

```
{{2, 0}, {0, 2}}
```



Επαναλήψεις μέσω της Nest και Fold (1)

?Nest

Nest[Cos, Pi/2, 2]

1

Στην παραπάνω πρόταση πρώτα υπολογίζεται το $\text{Cos}[\text{Pi}/2] \rightarrow 0$ και στη συνέχεια υπολογίζεται το $\text{Cos}[0] \rightarrow 1$ δηλ. $\text{Cos}[\text{Cos}[\text{Pi}/2]]$. Η Nest λοιπόν εφαρμόζει την συνάρτηση f , n φορές στην έκφραση expr .

Η Nest είναι πολύ χρήσιμη στις αναδρομικές ακολουθίες. Ας υποθέσουμε ότι έχουμε να υπολογίσουμε τον 5ο όρο της ακολουθίας

$$a_n = 1/2 * (a_{n-1} + 2/a_{n-1}), a_0 = 1$$

Η παραπάνω αναδρομική ακολουθία αποτελεί τον τύπο του Newton για τον υπολογισμό της ρίζας του 2.



Επαναλήψεις μέσω της Nest και Fold (2)

```
f[x_] := (1/2) * (x + 2/x)
```

```
Nest[f, 1, 5] // N
```

```
1.41421
```

Αν θέλουμε μπορούμε να δούμε και όλους τους ενδιάμεσα όρους της ακολουθίας με την NestList

```
NestList[f, 1, 5] // N
```

```
{1., 1.5, 1.41667, 1.41422, 1.41421, 1.41421}
```

Ο γενικός τύπος της προσέγγισης ρίζας μιας συνάρτησης f είναι

$$x_n = x_{n-1} - f(x_{n-1}) / f'(x_{n-1})$$



Επαναλήψεις μέσω της Nest και Fold (3)

Συνεπώς θα μπορούσαμε να έχουμε

$$f[x_] := x^2 - 5x + 6$$

$$g[x_] := x - f[x]/f'[x]$$

```
NestList[g, 1, 5] // N
```

```
{1., 1.66667, 1.93333, 1.99608, 1.99998, 2.}
```

```
NestList[g, 3, 5] // N
```

```
{3., 3., 3., 3., 3., 3.}
```



Επαναλήψεις μέσω της Nest και Fold (4)

Αν πάλι θέλουμε να εφαρμόσουμε την Nest έως ότου επέλθει σύγκλιση θα χρησιμοποιήσουμε την FixedPoint

```
FixedPoint[g, 3]
```

```
3
```

που θα μπορούσε να γίνει παρόμοια με την εξής εντολή

```
k = 2;
```

```
While[Abs[Nest[g, 1, k] - Nest[g, 1, k - 1]] > E-12, k = k + 1];
```

```
Nest[g, 3, k]
```

```
3
```



Άσκηση 4.1

Άσκηση 4. Προσπαθήστε να υπολογίσετε τον 20ο όρο της ακολουθίας

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

ή

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}, \begin{pmatrix} F_1 \\ F_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Λύση άσκησης.

```
f[x_] := {{1, 1}, {1, 0}}.x
```

```
Nest[f, {{1}, {0}}, 20][[2]]
```

```
{6765}
```



Άσκηση 4.2

```
Fibonacci[20]
```

```
6765
```

Θα μπορούσαμε να γράψουμε και συνάρτηση για τον υπολογισμό του n-οστού όρου Fibonacci

```
fibonacci[n_Integer] := Nest[Function[{x}, {{1, 1}, {1, 0}}.x], {{1}, {0}}, n][[2, 1]]
```

```
fibonacci[20]
```

```
6765
```

ή αλλιώς



Άσκηση 4.3

```
g[{x_, y_}] := {x + y, x}
```

```
Nest[g, {{1}, {0}}, 20][[2]]
```

```
{6765}
```

```
g1[n_Integer] := Nest[Function[{x}, {x[[1]] + x[[2]], x[[1]]}], {{1}, {0}}, n][[2, 1]]
```

```
g1[20]
```

```
6765
```



Άσκηση 4.4

Άλλος τρόπος μέσω διαδικασιακού προγραμματισμού

```
fibon[n_Integer] :=  
  (s1 = 1;  
   s2 = 1;  
   Do[s = s1 + s2; s2 = s1; s1 = s, {i, 1, n - 2}];  
   s)
```

```
fibon[20]
```

```
6765
```



Άσκηση 4.5

Σύγκριση χρόνων

Timing[fibonacci[1000]]

{0.03125,

43 466 557 686 937 456 435 688 527 675 040 625 802 564 660 517 371 780 402 481 729 089 536 555 ∴
417 949 051 890 403 879 840 079 255 169 295 922 593 080 322 634 775 209 689 623 239 873 322 ∴
471 161 642 996 440 906 533 187 938 298 969 649 928 516 003 704 476 137 795 166 849 228 875 }

Timing[g1[1000]]

{0.015625,

43 466 557 686 937 456 435 688 527 675 040 625 802 564 660 517 371 780 402 481 729 089 536 555 ∴
417 949 051 890 403 879 840 079 255 169 295 922 593 080 322 634 775 209 689 623 239 873 322 ∴
471 161 642 996 440 906 533 187 938 298 969 649 928 516 003 704 476 137 795 166 849 228 875 }



Άσκηση 4.6

Timing[Fibonacci[1000]]

```
{0.,  
43 466 557 686 937 456 435 688 527 675 040 625 802 564 660 517 371 780 402 481 729 089 536 555`·:  
417 949 051 890 403 879 840 079 255 169 295 922 593 080 322 634 775 209 689 623 239 873 322`·:  
471 161 642 996 440 906 533 187 938 298 969 649 928 516 003 704 476 137 795 166 849 228 875}
```

Timing[fibon[1000]]

```
{0.,  
43 466 557 686 937 456 435 688 527 675 040 625 802 564 660 517 371 780 402 481 729 089 536 555`·:  
417 949 051 890 403 879 840 079 255 169 295 922 593 080 322 634 775 209 689 623 239 873 322`·:  
471 161 642 996 440 906 533 187 938 298 969 649 928 516 003 704 476 137 795 166 849 228 875}
```



Η εντολή Fold (1)

?Fold

```
Fold[f, x, list] gives the last element of FoldList[f, x, list]. >>
```

Ας δούμε με ένα παράδειγμα πως δουλεύει η Fold

```
Fold[f, a, {b, c, d}]
```

```
f[f[f[a, b], c], d]
```

Την χρησιμοποιούμε για συναρτήσεις με δύο μεταβλητές. Δέχεται όπως φαίνεται παραπάνω μια αρχική τιμή για το x (a) της συνάρτησης $f[x,y]$ και υπολογίζει το $f[a,b]$ όπου b το πρώτο στοιχείο της λίστας $\{b,c,d\}$. Στη συνέχεια τοποθετεί την τιμή αυτή ως όρισμα για το x και συνεχίζει με δεύτερη τιμή του y την c που είναι η δεύτερη τιμή της λίστας $f[f[a,b],c]$. Στη συνέχεια τοποθετεί την τιμή αυτή ως όρισμα για το x και συνεχίζει με τρίτη τιμή του y την d που είναι η τρίτη τιμή της λίστας $f[f[f[a,b],c], d]$.



Η εντολή Fold (2)

Η παρακάτω συνάρτηση υπολογίζει το άθροισμα $1+2+3+\dots+10$

```
Fold[Plus, 0, Range[10]]
```

```
55
```

Η FoldList μας δίνει τα ενδιάμεσα αποτελέσματα

```
FoldList[Plus, 0, Range[10]]
```

```
{0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55}
```



Άσκηση 5

Άσκηση 5. Να υπολογίσετε το μέγιστο στοιχείο της λίστας x κάνοντας χρήση των συναρτήσεων `Max`, `Fold`.

Λύση της άσκησης

```
f[x_] := Fold[Max, -Infinity, x]
```

```
f[{2, 3, -9, 10}]
```

```
10
```



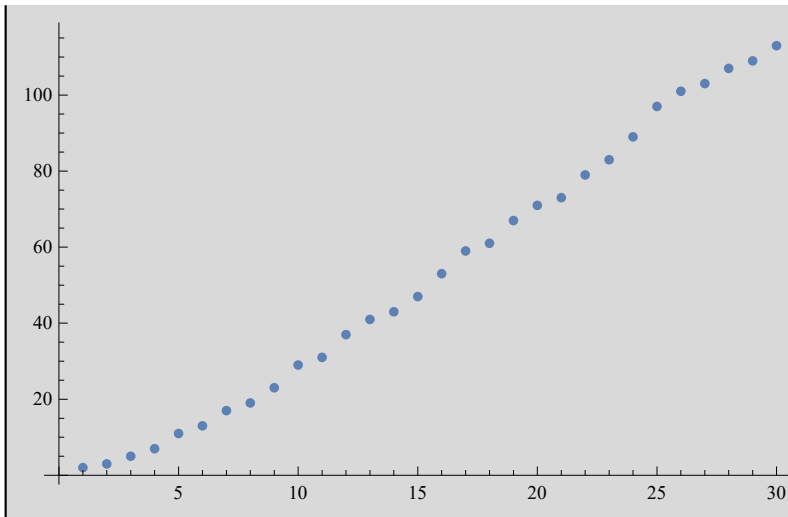
Παραδείγματα (1)

Άσκηση 1. Να γίνει η γραφική παράσταση των πρώτων αριθμών από 1-30.

```
a = Table[Prime[i], {i, 1, 30}]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,  
53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113}
```

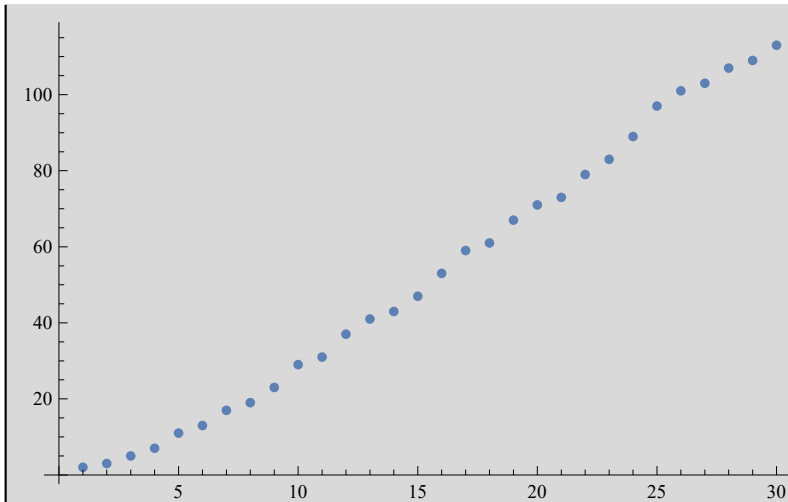
```
ListPlot[a]
```



Παραδείγματα (2)

ή

```
ListPlot[Table[Prime[i], {i, 1, 30}]]
```



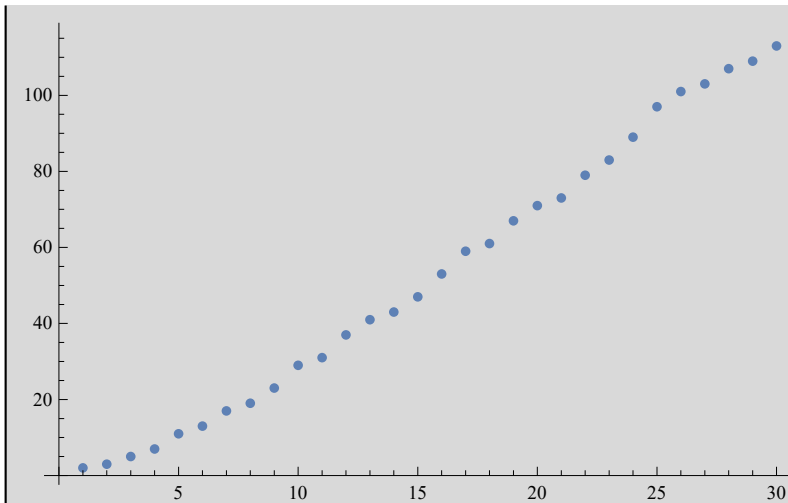
Παραδείγματα (3)

Θα μπορούσαμε όμως να παραλείψουμε την μεταβλητή i όπως φαίνεται παρακάτω

```
Map[Prime[#] &, Range[30]]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,  
53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113}
```

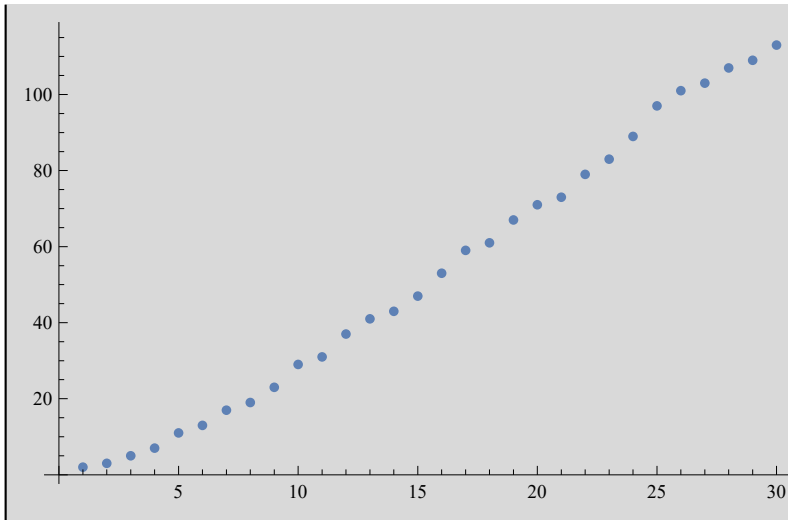
```
ListPlot[%]
```



Παραδείγματα (4)

ή μέσω μιας συνάρτησης

```
ListPlot[Map[Prime[#] &, Range[30]]]
```



Συναρτήσεις όπως οι παραπάνω ονομάζονται one-liners και αποτελούνται μόνο από συναρτήσεις, σταθερές αλλά και ορίσματα που βρίσκονται ως ορίσματα σε περίπτωση συναρτήσεων. Στην πρώτη περίπτωση επίλυσης της παραπάνω άσκησης έχουμε την εξάρτηση της εντολής από την μεταβλητή i σε αντίθεση με την τελευταία μορφή εντολής.



Παραδείγματα (5)

Άσκηση 2. Να γραφεί συνάρτηση που υπολογίζει τα γράμματα που εμφανίζονται σε ένα string καθώς και την συχνότητα που εμφανίζονται.

Έστω ότι έχουμε το string

```
a = "Nikos Karampetakis"
```

```
Nikos Karampetakis
```

Τα γράμματα από τα οποία αποτελείται βρίσκονται στην παρακάτω λίστα

```
a = Characters[a]
```

```
{N, i, k, o, s, , K, a, r, a, m, p, e, t, a, k, i, s}
```

Χρησιμοποιώντας την εντολή `Union` παίρνουμε τα διαφορετικά γράμματα που εμφανίζονται στην λίστα

```
c = Union[a]
```

```
{ , a, e, i, k, K, m, N, o, p, r, s, t}
```



Παραδείγματα (6)

Με την `Count` μπορώ να υπολογίσω πόσες φορές εμφανίζεται κάθε γράμμα στην παραπάνω λίστα π.χ.

```
Count[c, "a"]
```

```
1
```

ή καλύτερα παίρνουμε το ζεύγος (χαρακτήρα, πλήθος φορών που εμφανίσθηκε)

```
{"a", Count[c, "a"]}
```

```
{a, 1}
```

ή αλλιώς

```
{#, Count[c, #]} &["a"]
```

```
{a, 1}
```



Παραδείγματα (7)

Η σύνθεση των παραπάνω εντολών μας οδηγεί στην συνάρτηση (σε μορφή one-liner)

```
frequency[list_] := Map[{#, Count[Characters[list], #]} &, Union[Characters[list]]]
```

```
frequency["Nikos Karampetakis"]
```

```
{{ , 1}, {a, 3}, {e, 1}, {i, 2}, {k, 2}, {K, 1}, {m, 1}, {N, 1}, {o, 1}, {p, 1}, {r, 1}, {s, 2}, {t, 1}}
```



Ασκήσεις για σπίτι

Άσκηση 1. Να γραφεί συνάρτηση που θα κωδικοποιεί ένα string σύμφωνα με τον τύπο $f(x)=[(5x+4)\bmod 26]+65$ (δες παρόμοια άσκηση στην ενότητα με την εντολή Map). Χρησιμοποίησε την εντολή Characters που επιστρέφει τα γράμματα ενός string καθώς και την StringJoin που συνδέει τους χαρακτήρες μιας λίστας. Η συνάρτηση σου θα δέχεται ως όρισμα το string.

Άσκηση 2. Προσπαθήστε να λύσετε την άσκηση 2 της τελευταίας ενότητας με διαδικασιακό προγραμματισμό.

Άσκηση 3. Να γραφεί συνάρτηση (one-liner) $f[n_Integer]$ που θα κάνει τα εξής:

- α) θα δημιουργεί μια λίστα $a1$ με τους φυσικούς αριθμούς $1,2,3,\dots,n$
- β) θα δημιουργεί μια λίστα $a2$ που θα περιέχει τα στοιχεία της $a1$ εκτός από το $1o, 3o, 5o,\dots,2*[n/2]$.
- γ) θα δημιουργεί μια λίστα $a3$ που θα περιέχει τα στοιχεία της $a2$ εκτός από το $1o, 3o, 5o,\dots$
- δ) θα συνεχίσετε την διαδικασία αυτή εως ότου πάρετε μια λίστα με ένα μόνο στοιχείο.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Νικόλαος Καραμπετάκης.
«Συμβολικές Γλώσσες Προγραμματισμού. Ενότητα 8: Συναρτησιακός
προγραμματισμός». Έκδοση: 1.0. Θεσσαλονίκη 2014.

Διαθέσιμο από τη δικτυακή διεύθυνση:

```
http://eclass.auth.gr/courses/OCRS430/
```



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [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».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

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

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.





Τέλος ενότητας

Επεξεργασία: Αναστασία Γ. Γρηγοριάδου
Θεσσαλονίκη, Εαρινό εξάμηνο 2014-2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ