ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
6° Εξάμηνο

Συμβολικός Προγραμματισμός (Symbolic Programming)

Μάθηση από Παραδείγματα

- **Το Πρόβλημα της Αναλογίας**

Δημοσθένης Σταμάτης
http:/www.iee.ihu.gr/~demos
Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων

Συμβολικός Προγραμματισμός (Symbolic Programming)

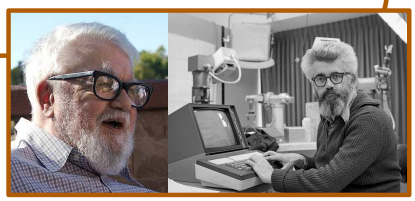
☞ Ο προγραμματισμός εφαρμογών Τεχνητής Νοημοσύνης περιλαμβάνει (κυρίως) **χειρισμό συμβόλων και συμβολικών εκφράσεων** και όχι αριθμών. Τα σύμβολα μπορεί να αντιπροσωπεύουν αντικείμενα (ή οντότητες) και τις σχέσεις μεταξύ αυτών των αντικειμένων.

☞ Γενικά απαιτούνται πολύπλοκες δομές συμβόλων για να αναπαραστήσουν τη γνώση μας για τον κόσμο

Συμβολικός Προγραμματισμός (Symbolic Programming)

Στην ιδέα του συμβολικού προγραμματισμού αναφέρθηκε πρώτος ο **John Mc Arthy**(*) ένας από τους πολύ γνωστούς ανθρώπους της Τεχνητής Νοημοσύνης:

*«μπορούμε να ορίσουμε **συναρτήσεις εκφράσεων**, όχι μόνο συναρτήσεις αριθμών ή συμβολοσειρών ή bits ή άλλων τύπων δεδομένων»*



(*) "*Recursive Functions of Symbolic Expressions*" published in Communications of the ACM, 1960

Συμβολικός Προγραμματισμός στην Prolog

Στην Prolog, δύο είναι τα βασικά χαρακτηριστικά που δίνουν τη δυνατότητα συμβολικού προγραμματισμού:

(α) Η δυνατότητα μίας μεταβλητής να παραμένει μη-δεσμευμένη (ελεύθερη) σε κάποια τιμή κατά τη διάρκεια εκτέλεσης ενός προγράμματος και

(β) Το γεγονός ότι η Prolog (επίσης κατά τη διάρκεια της εκτέλεσης) **δεν υπολογίζει τις εκφράσεις**, δηλαδή τους **σύνθετους όρους** που εμφανίζονται ως παράμετροι των κατηγορημάτων.

Οι Συναρτησιακοί Οροι (funtional terms) ή αλλιώς σύνθετοι (compound) όροι δεν είναι εκτελέσιμοι!

```
lives(name(nick, antoniou), address(kriezotou, 15)).
lives(name(john,stratakis), address(papanikou, 9)).
lives(name(nick, farmakis), address(antheon, 112)).
```

```
?- name(nick, X).
    no
```

```
?- lives(name(nick, X), address(Y, Z)).
   X = antoniou, Y = kriezotou, Z = 15  ->;
   X = farmakis, Y = antheon, Z = 112  ->;
  no
```


ΤΜΗΜΑ ΜΗΧ.ΠΛΗΡΟΦΟΡΙΚΗΣ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΔΙΠΑΕ

Οι Συναρτησιακοί Οροι (funtional terms) ή αλλιώς σύνθετοι (compound) όροι δεν είναι εκτελέσιμοι!

lives(name(nick, antoniou), address(kriezotou, 15)).
lives(name(john,stratakis), address(papanikou, 9)).
lives(name(nick, farmakis), address(antheon, 112)).

```
?- lives(name(nick, X), Y).
   X = antoniou, Y = address(kriezotou, 15) ->;
   X = farmakis, Y = address(antheon, 112) ->;
   no
```

```
?- lives(name(nick, X), _).
   X = antoniou ->;
   X = farmakis ->;
   no
```

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 6ο ΕΞΑΜΗΝΟ

Οι σύνθετοι όροι στους Κανόνες !

```
lives(name(nick, antoniou), address(kriezotou, 15)).
lives(name(john,stratakis), address(papanikou, 9)).
lives(name(nick, farmakis), address(antheon, 112)).
```

```
findname(X,Y) :- lives(name(X,Y), _ ).
```

```
?- findname(nick, X).
   X = antoniou ->;
   X = farmakis ->;
   no
```


ΤΜΗΜΑ ΜΗΧ.ΠΛΗΡΟΦΟΡΙΚΗΣ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΔΙΠΑΕ

Οι σύνθετοι όροι στους Κανόνες !

lives(name(nick, antoniou), address(kriezotou, 15)).
lives(name(john,stratakis), address(papanikou, 9)).
lives(name(nick, farmakis), address(antheon, 112)).

findname(X,Y) :- lives(name(X,Y), _).

```
?- findname(X, antoniou).
   X = nick ->;
   no
```

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 6ο ΕΞΑΜΗΝΟ

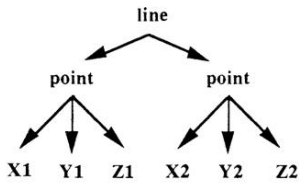
Σύνθετοι Οροι = Σύνθετες Δομές Δεδομένων

Ενα σημείο στο επίπεδο -> **point(X, Y)**

Ενα σημείο στο χώρο -> **point(X, Y, Z)**

Ευθεία στο χώρο -> **line(point(X1, Y1, Z1), point(X2, Y2, Z2))**

Ενα τρίγωνο στο επίπεδο -> **triangle(point(X1,Y1),point(X2,Y2),point(X3,Y3))**



Πότε δύο ευθείες είναι παράλληλες (στο επίπεδο)?

parallel(line(point(X1,Y1),point(X2,Y2)), line(point(X3,Y3),point(X4,Y4))) :-

. . .

Συμβολικός Προγραμματισμός στην Prolog

Η δυνατότητα χρήσης σύνθετων όρων που περιλαμβάνουν ελεύθερες μεταβλητές μας δίνει τη δυνατότητα αναπαράστασης, με γενικό τρόπο, σύνθετων δομών δεδομένων με τη συμβολική τους μορφή.

Ένα τέτοιο παράδειγμα, που έχουμε ήδη δει, είναι αυτό της αναπαράστασης της μερικής λύσης στο πρόβλημα των 8-Βασιλισσών με τη μορφή της παρακάτω λίστας:

template([[1,Y1],[2,Y2],[3,Y3],[4,Y4],[5,Y5],[6,Y6],[7,Y7],[8,Y8]]).

Ο Συμβολικός τρόπος υπολογισμού του αθροίσματος

```
symbolic_sum([],0).
symbolic_sum([H|T],S):-
        symbolic_sum(T,ST),
        S=H+ST.          % το άθροισμα δεν υπολογίζεται
```

ή καλύτερα:

```
symbolic_sum([],0).
symbolic_sum([H|T],H+ST):-    % το = ως τελεστής ενοποίησης μας επέτρεψε
        symbolic_sum(T,ST).  % να μεταφέρουμε το H+ST ως όρισμα
```


Δοκιμές Εκτέλεσης του συμβολικού αθροίσματος

```
?- symbolic_sum([8,1,2,4],S).
   S = 8+1+2+4+0

?- symbolic_sum([8,X,2,Y,4]).
   S = 8+X+2+Y+4+0

?- symbolic_sum([8,1,2,4],S), SA is S.
   S = 8+1+2+4+0,
   SA = 15.
```

Υπολογισμός Αριθμητικού αθροίσματος με τη βοήθεια του Συμβολικού

```
arithmetic_sum(L, AS):-
        symbolic_sum(L,S),
        AS is S.
```

```
?- symbolic_sum([8,1,2,X,4],S), guess(X), SA is S.
   S = 8+1+2+9+4+0,
   X = 9,
   SA = 24.
```


Συμβολικός Υπολογισμός Παραγώγου

```
derivative(X, X, 1).
derivative(C, X, 0) :- atomic(C), C \=X.
derivative(sin(X), X, cos(X)).
derivative(cos(X), X, -sin(X)).
derivative(U+V, X, A+B) :- derivative(U, X, A), derivative(V, X, B).
derivative(U*V, X, A*V+B*U) :- derivative(U, X, A), derivative(V, X, B).
```

Παραδείγματα εκτέλεσης:

```
?- derivative(x*x + 2, x, P).
   P = 1*x+1*x+0
?- derivative(x*x*x + 2, x, P).
   P = (1*x+1*x)*x + 1*(x*x)+0
```

Το Πρόβλημα της Αναλογίας

(Μάθηση από παραδείγματα)

2	→	4
3	→	6
5	→	10
10	→	???

Αντιστοιχίες αριθμών

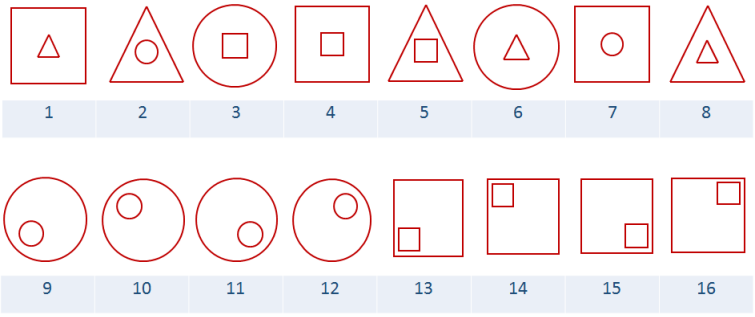
Μαρούλι	→	Λαχανικό
Μήλο	→	???

Το μήλο είναι φρούτο! Το μαρούλι τι είναι?

ΤΜΗΜΑ ΜΗΧ.ΠΛΗΡΟΦΟΡΙΚΗΣ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΔΙΠΑΕ

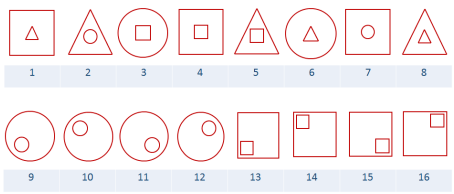
Το Πρόβλημα της Αναλογίας

(Μάθηση από παραδείγματα)



ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 6ο ΕΞΑΜΗΝΟ

Το Πρόβλημα της Αναλογίας
(Μάθηση από παραδείγματα)

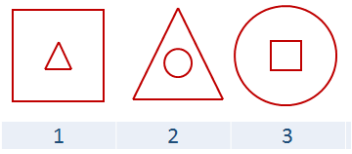


- Αν το σχήμα 1 σχετίζεται με το σχήμα 5, τότε ποιο σχήμα σχετίζεται με το σχήμα 3 **?**
- Η εικόνα στο 1 είναι ένα τρίγωνο μέσα σε ένα τετράγωνο, ενώ η εικόνα στο 5 είναι ένα τετράγωνο μέσα σε ένα τρίγωνο, δηλαδή τα δύο σχήματα είναι «αντίστροφα».
- Έτσι θα πρέπει να ψάξουμε για το αντίστροφο της εικόνας 3 που είναι ένα τετράγωνο μέσα σε ένα κύκλο, δηλαδή η εικόνα 7 η οποία έχει ένα κύκλο μέσα σε ένα τετράγωνο.

Το Πρόβλημα της Αναλογίας
(Μάθηση από παραδείγματα)

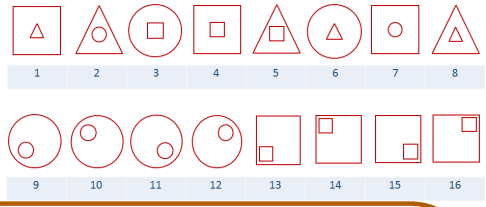
Θα μπορούσαμε να αναπαραστήσουμε τις εικόνες με τη μορφή γεγονότων ως εξής:

figure(1, middle(triangle, square)).
figure(2, middle(circle, triangle)).
figure(3, middle(square, circle)).



- Ο όρος **middle(triangle,square)** αποτελεί μία **συμβολική περιγραφή του σχήματος**, όπου απλά τονίζονται τα ποιοτικά χαρακτηριστικά της αναπαράστασης «ένα σχήμα, το τρίγωνο, είναι μέσα σε ένα άλλο σχήμα, στο τετράγωνο».

Το Πρόβλημα της Αναλογίας
(Μάθηση από παραδείγματα)



Οι σχέσεις μεταξύ των εικόνων θα μπορούσαν να αναπαρασταθούν ως εξής:

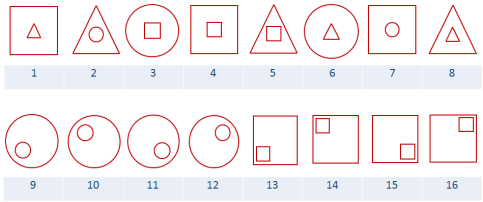
relation(middle(S1,S2), middle(S2,S1), inverse).

Το παραπάνω γεγονός δηλώνει ότι υπάρχει σχέση αντιστροφής (inverse) μεταξύ της πρώτης εικόνας η οποία περιέχει το σχήμα **S1** μέσα στο σχήμα **S2** και της δεύτερης εικόνας με το σχήμα **S2** μέσα στο σχήμα **S1**.

Προσέξτε επίσης τη συμβολική μορφή της αναπαράστασης της σχέσης, καθώς και το γεγονός ότι αυτή ισχύει για οποιαδήποτε δύο σχήματα καθώς οι μεταβλητές **S1** και **S2** είναι μη-τοποθετημένες.

ΤΜΗΜΑ ΜΗΧ.ΠΛΗΡΟΦΟΡΙΚΗΣ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΔΙΠΑΕ

Το Πρόβλημα της Αναλογίας
(Μάθηση από παραδείγματα)



Αντίστοιχα:

figure(9, bottomleft(circle,circle)).
figure(10, topleft(circle,circle)).
figure(11, bottomright(circle,circle)).

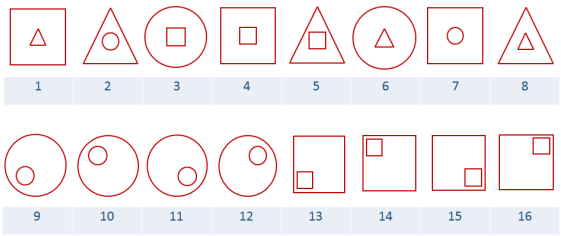
και

relation(middle(S1,S2),middle(S3,S4),changeout):- S2\=S4.
relation(middle(S1,S2),middle(S3,S4),changein):- S1\=S3.

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 6ο ΕΞΑΜΗΝΟ

Το Πρόβλημα της Αναλογίας

(Μάθηση από παραδείγματα)




```
find_relation_between(N1, N2, Relation) :-
        figure(N1, Dercription1),
        figure(N2, Dercription2),
        relation(Dercription1, Dercription2,  Relation).
```