

# ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

## 6<sup>ο</sup> Εξάμηνο

### - Ενότητα 3 -

#### Το Συντακτικό της PROLOG Δομές Δεδομένων στην PROLOG

Δημοσθένης Σταμάτης

<http://www.iee.ihu.gr/~demos>

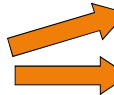
Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων

### Ανακοινώσεις του μαθήματος

- ➔ Επισκέπτομαι την σελίδα <https://apps.it.teithe.gr/> και αφού κάνω “login” βλέπω αν υπάρχει νέα ανακοίνωση του μαθήματος στον πίνακα «6<sup>ο</sup> εξάμηνο»
- ➔ Γενικά μπορώ να ορίσω ποιους πίνακες θέλω να παρακολουθώ (οδηγίες στο: [https://www.iee.ihu.gr/support/dept\\_services/dept\\_account\\_ aboard/](https://www.iee.ihu.gr/support/dept_services/dept_account_ aboard/)) και μου έρχεται email όταν βγαίνει μια νέα ανακοίνωση.
- ➔ Κατεβάζω για Android την εφαρμογή IEEapps (ITapps) που υπάρχει στο [https://www.iee.ihu.gr/support/dept\\_services/](https://www.iee.ihu.gr/support/dept_services/) και την εγκαθιστώ στο κινητό μου. Κάθε φορά που βγαίνει μια νέα ανακοίνωση στους πίνακες που έχω κάνει εγγραφή, λαμβάνω ειδοποίηση στο κινητό.

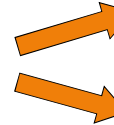
### ΣΥΝΤΑΚΤΙΚΟ της PROLOG (1/2)

**Γεγονότα  
(Facts)**



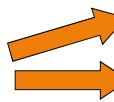
```
father(uranus, kronos).
derivative(X, X, 1).
```

**Κανόνες  
(Rules)**



```
grandfather(X,Y) :-
    father(X,Z),
    parent(Z,Y).
derivative(U+V, X, A+B) :-
    derivative(U, X, A),
    derivative(V, X, B).
```

**Ερωτήσεις  
(Queries)**



```
?- father(X, Y).
?- derivative(2*x*x, x, A).
```

### ΣΥΝΤΑΚΤΙΚΟ της PROLOG (2/2)

**Κανόνες (Rules):**

**Εάν (If)**

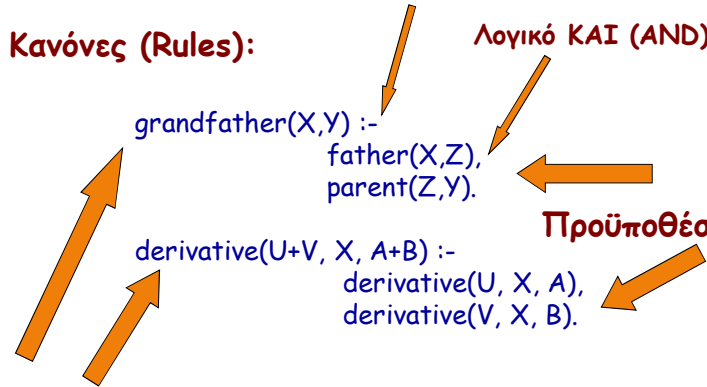
**Λογικό ΚΑΙ (AND)**

```
grandfather(X,Y) :-
    father(X,Z),
    parent(Z,Y).
```

```
derivative(U+V, X, A+B) :-
    derivative(U, X, A),
    derivative(V, X, B).
```

**Προϋποθέσεις**

**Συμπέρασμα**



## ΣΥΝΤΑΚΤΙΚΟ της PROLOG (Τυπικός Ορισμός) 1/4



Ένα πρόγραμμα στην PROLOG είναι ένα πεπερασμένο σύνολο από φράσεις (clauses).

Οι φράσεις έχουν την παρακάτω γενική μορφή:

$$B :- A_1, A_2, \dots, A_n. \quad \text{με } n \geq 0$$

όπου τα  $B, A_1, A_2, \dots, A_n$  είναι ατομικοί τύποι (atomic formulas)



Ένας ατομικός τύπος είναι μια έκφραση της μορφής:

$$p(t_1, t_2, \dots, t_m) \quad \text{με } m \geq 0$$

Το  $p$  ονομάζεται  $m$ -θέσιο κατηγορήμα (predicate) και συμβολίζεται

$p/m$  και τα  $t_1, t_2, \dots, t_m$  είναι οι όροι (terms) του κατηγορήματος.

## ΣΥΝΤΑΚΤΙΚΟ της PROLOG (Τυπικός Ορισμός) 2/4



Ο όρος (term) μπορεί να είναι:

1. μια σταθερά (constant),
2. μια μεταβλητή (variable) ή
3. μια έκφραση της μορφής:  $f(t_1, t_2, \dots, t_k)$  με  $k > 0$



Η έκφραση  $f(t_1, t_2, \dots, t_k)$  ονομάζεται **συναρτησιακός όρος** (functional term)

Τα ορίσματα  $t_1, t_2, \dots, t_k$  του συναρτησιακού όρου είναι με τη σειρά τους όροι.

Το  $f$  ονομάζεται  $k$ -θέσιο **συναρτησιακό σύμβολο** (functional symbol)

**Οι Συναρτησιακοί Όροι (functional terms)****δεν είναι εκτελέσιμοι!**

(1/2)

```
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
```

**Οι Συναρτησιακοί Όροι (functional terms)****δεν είναι εκτελέσιμοι!**

(2/2)

```
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
```

**Από τα Γεγονότα στους Κανόνες! (1/3)**

```
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
```

**Από τα Γεγονότα στους Κανόνες! (2/3)**

```
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
```

## Από τα Γεγονότα στους Κανόνες! (3/3)

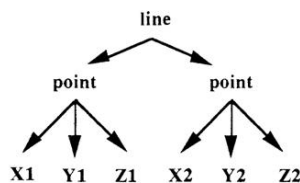
```
lives(name(nick, antoniou), address(kriezotou, 15)).
lives(name(john, stratakis), address(papanikou, 9)).
lives(name(nick, farmakis), address(antheon, 112)).
```

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

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

## Σύνθετοι Οροι = Σύνθετες Δομές Δεδομένων

```
Ένα σημείο στο επίπεδο -> point(X, Y)
Ένα σημείο στο χώρο -> point(X, Y, Z)
Ευθεία στο χώρο -> line(point(X1, Y1, Z1), point(X2, Y2, Z2))
Ένα τρίγωνο στο επίπεδο -> triangle(point(X1, Y1), point(X2, Y2), point(X3, Y3))
```

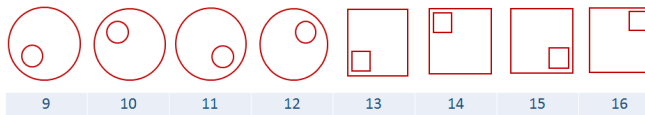
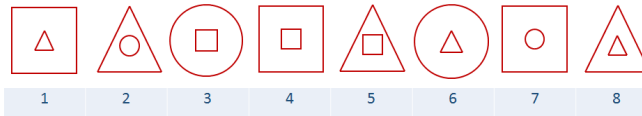


Πότε δύο ευθείες είναι παράλληλες (στο επίπεδο)?

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

...

## Σύνθετοι Όροι και το πρόβλημα της αναλογίας



```
figure(1,middle(triangle,square)).
```

```
figure(5,middle(square,triangle)).
```

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

Σύνθετοι (συναρτησιακοί) όροι

Μεταβλητές

Σταθερά

## ΣΥΝΤΑΚΤΙΚΟ της PROLOG (Τυπικός Ορισμός) 3/4



Όπως έχει αναφερθεί φράση είναι μία έκφραση της μορφής

$$B :- A_1, A_2, \dots, A_n. \quad \text{με } n \geq 0$$

Η παραπάνω έκφραση ονομάζεται και **κανόνας (rule)**.

Το  $B$  αποτελεί την **κεφαλή** της φράσης ή το **συμπέρασμα** του κανόνα

Τα  $\{A_1, A_2, \dots, A_n\}$  αποτελούν το **σώμα** της φράσης ή τις **προϋποθέσεις** του κανόνα.



Εάν  $n = 0$  η παραπάνω φράση παίρνει τη μορφή:

$$B :- \text{true}. \quad \text{ή ισοδύναμα } B.$$

και ονομάζεται **γεγονός (fact)**



Εάν λείπει η κεφαλή  $B$  της φράσης αυτή παίρνει τη μορφή:

$$:- A_1, A_2, \dots, A_n.$$

και ονομάζεται **ερώτηση (query)** ή **στόχος (goal)**

**ΣΥΝΤΑΚΤΙΚΟ της PROLOG (Τυπικός Ορισμός) 4/4**

Το σύνολο των φράσεων (γεγονότων και κανόνων) που έχουν ως κεφαλή τους το ίδιο κατηγορημα ορίζει για την PROLOG την έννοια του υποπρογράμματος ή διαδικασίας (*procedure*)



Στην PROLOG ο τρόπος κατασκευής των ατομικών τύπων (κατηγορημάτων), από συντακτική άποψη, είναι ακριβώς ίδιος με αυτόν των όρων. Στην ουσία όμως, η PROLOG χειρίζεται διαφορετικά τις δύο αυτές οντότητες.

Στην διάρκεια της εκτέλεσης τα κατηγορήματα αντιστοιχούν σε κλήσεις διαδικασιών (*procedures*), ενώ οι όροι αντιστοιχούν σε δομές δεδομένων που μεταφέρονται ως παράμετροι στις κλήσεις των κατηγορημάτων.