

# Κεφάλαιο 11 :

## Γλώσσες Λογικού Προγραμματισμού

*Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών*

# Λογικός Προγραμματισμός

- Βασίζεται στον κατηγορηματικό λογισμό
- Κατηγορήματα – δομικά συστατικά  $P(a_1, a_2, \dots, a_k)$
- $P$  από το Predicates (κατηγορήματα)....
- “Logic Programming (LP) is at the nexus of knowledge representation, AI, mathematical logic, databases, and programming languages. It allows programming to be more declarative, by specifying “**what**” to do instead of “**how**” to do **it**”. The opposite is PROCEDURAL (or NON-DECLARATIVE)

# Έννοιες του Λογικού Προγραμματισμού

## ■ Τελεστές

- σύζευξη, διάξευξη, άρνηση, συνεπαγωγή

## ■ Καθολικοί και υπαρξιακοί ποσοδείκτες

## ■ Προτάσεις

- κάποιες φορές είναι αληθείς, κάποιες φορές ψευδείς, συχνά δε γνωρίζουμε
- αξιώματα – θεωρούνται αληθή
- θεωρήματα – μπορούν να αποδειχθούν αληθή
- υποθέσεις (στόχοι) – αυτά που θέλουμε να αποδείξουμε ότι είναι αληθή

# Prolog

## ■ Η σημασία της πρότασης

```
mother(mary, fred) .
```

```
% μπορείτε να το σκεφτείτε σαν ένα
```

```
% κατηγορημα που βεβαιώνει ότι η mary
```

```
% είναι η μητέρα του fred -
```

```
% ή σαν μια δομή δεδομένων (δένδρο)
```

```
% στην οποία το συναρτησιακό σύμβολο
```

```
% (άτομο) mother είναι η ρίζα,
```

```
% η mary είναι το αριστερό παιδί, και
```

```
% ο fred είναι το δεξιό παιδί
```

```
fat(albert) .
```

```
rainy(rochester) .
```

# Prolog

- Οι κανόνες είναι θεωρήματα που επιτρέπουν στο διερμηνέα να συναγάγει την αλήθεια προτάσεων
- Για να έχουν ενδιαφέρον, οι κανόνες συνήθως περιέχουν μεταβλητές `employed(X) :- employs(Y, X).`

μπορεί να διαβαστεί ως:

για όλα τα  $X$ , το  $X$  εργάζεται αν υπάρχει ένα  $Y$  τέτοιο  
ώστε το  $Y$  να απασχολεί το  $X$

- Κατεύθυνση της συνεπαγωγής:

# Prolog

- Η εμβέλεια μιας μεταβλητής είναι η πρόταση στην οποία εμφανίζεται
  - Οι μεταβλητές που εμφανίζονται πρώτη φορά στο αριστερό μέλος έχουν έμμεσα καθολικούς ποσοδείκτες
  - Οι μεταβλητές που εμφανίζονται πρώτη φορά στο σώμα της πρότασης έχουν έμμεσα υπαρξιακούς ποσοδείκτες

# Prolog

## ■ Παρομοίως:

```
grandmother(A, C) :- mother(A, B),  
                    mother(B, C).
```

**μπορεί να διαβαστεί ως:**

για όλα τα  $A, C$  [το  $A$  είναι η grandmother του  $C$  αν υπάρχει κάποιο  $B$  ώστε το  $A$  να είναι η μητέρα του  $B$  και το  $B$  να είναι η μητέρα του  $C$ ].

**Θα θέλουμε και έναν άλλο κανόνα που να λέει**

```
grandmother(A, C) :- mother(A, B),  
                    father(B, C).
```

# Prolog

- Για να εκτελέσουμε ένα πρόγραμμα σε Prolog, κάνουμε μια ερώτηση στο διερμηνέα
  - Αυτό γίνεται διατυπώνοντας ένα θεώρημα – βεβαιώνοντας μια πρόταση – το οποίο ο διερμηνέας προσπαθεί να αποδείξει
    - Αν μπορεί, λέει *ναι*
    - Αν δε μπορεί, λέει *όχι*
    - Αν η πρότασή σας περιείχε μεταβλητές, ο διερμηνέας τυπώνει τις τιμές που πρέπει να τους δώσει για να κάνει την πρόταση αληθή.



# PROLOG EXAMPLES WITH EXPLANATIONS

Prolog always performs depth-first-search, Matches facts & rules (i.e. knowledge base) in top-down manner.

Resolves the goals or subgoals in left-to-right manner.

Most important thing to keep in mind while writing prolog program - "order of writing facts & rules always matters".

Example 1 : Below food table shows the facts, rules, goals and their english meanings.

### Facts

English meanings

**food(burger).** // burger is a food  
**food(sandwich).** // sandwich is a food  
**food(pizza).** // pizza is a food  
**lunch(sandwich).** // sandwich is a lunch  
**dinner(pizza).** // pizza is a dinner

### Rules

**meal(X) :- food(X).**

// Every food is a meal OR  
Anything is a meal if it is a food

### Queries / Goals

**?- food(pizza).**  
// Is pizza a food?  
**?- meal(X), lunch(X).**  
// Which food is meal and lunch?  
**?- dinner(sandwich).**  
// Is sandwich a dinner?

## PROLOG EXAMPLES WITH EXPLANATIONS

Example 2 : Below student-professor relation table shows the facts, rules, goals and their english meanings.

### Facts

**studies(charlie, csc135).**

// charlie studies csc135

**studies(olivia, csc135).**

// olivia studies csc135

**studies(jack, csc131).**

// jack studies csc131

**studies(arthur, csc134).**

// arthur studies csc134

**teaches(kirke, csc135).**

// kirke teaches csc135

**teaches(collins, csc131).**

// collins teaches csc131

**teaches(collins, csc171).**

// collins teaches csc171

**teaches(juniper, csc134).**

// juniper teaches csc134

### Rules

**professor(X, Y) :-**

**teaches(X, C), studies(Y, C).**

// X is a professor of Y if X teaches C and Y studies C.

### Queries / Goals

**?- studies(charlie, What).**

// charlie studies what? OR  
What does charlie study?

**?- professor(kirke, Students).**

// Who are the students of professor kirke.