Εργαστήριο Λογικού Προγραμματισμού

Μανόλης Μαρακάκης, Καθηγητής
mmarak@cs.hmu.gr

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Σχολή Μηχανικών
Ελληνικό Μεσογειακό Πανεπιστήμιο

Ενότητα 2: Μάθημα 2.

- **2. Συστατικά ενός Prolog Προγράμματος**

Μάθημα 2

❑ 2. Συστατικά ενός Prolog Προγράμματος (συνέχεια).

- 2.4 . Προτάσεις Κανόνες
- 2.5. Σύνθετες Ερωτήσεις
- 2.6. Δεδομένα
  - ✓ 2.6.1. Σταθερές
  - ✓ 2.6.2 Μεταβλητές
  - ✓ 2.6.3. Σύνθετοι Όροι
- 2.7. Κατηγορήματα Εισόδου και Εξόδου
- 2.8. Άσκηση στη τάξη

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- **Η γενική μορφή ενός κανόνα** είναι η εξής:

  A :- B1, B2,…,Bn όπου n≥0

  όπου το , (κόμμα) σημαίνει λογικό *and*.
- **A είναι *ατομικός τύπος* και ονομάζεται η *κεφαλή του κανόνα*.**
- **Bi όπου i = 1,…,n είναι στοιχειώδεις τύποι (αρνητικοί ή θετικοί ατομικοί τύποι) και αποτελούν το σώμα του κανόνα.**
- Η αλήθεια του στόχου A προϋποθέτει την αλήθεια όλων των στόχων Bi (i = 1,…,n).
- Το σύμβολο **:-** της Prolog αντιστοιχεί στο σύμβολο ← ή **if** της λογικής.

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- **Οι μεταβλητές που υπάρχουν στη κεφαλή κανόνων** και πιθανόν επαναλαμβάνονται στο σώμα μιας πρότασης έχουν **καθολική δέσμευση. Ενώ οι μεταβλητές που εμφανίζονται *μόνο* στο σώμα μιας πρότασης** έχουν **υπαρξιακή δέσμευση.**
- Η **εμβέλειά** μιας μεταβλητής είναι **όλος ο κανόνας**. Αυτό σημαίνει ότι **ίδιο όνομα μεταβλητής σε διαφορετικούς κανόνες αντιστοιχεί σε διαφορετικές μεταβλητές**. Συνεπώς, η **μετονομασία μιας μεταβλητής** σ' ένα κανόνα **δεν αλλάζει την σημασιολογία του προγράμματος**.

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Για να ήταν πληρέστερη η βάση γνώσης του παραδείγματος 2.1 θα έπρεπε να υπάρχουν επιπλέον τα γεγονότα γ4 μέχρι γ9 για να εκφράσουν τη σχέση ότι *"ο Χ είναι παιδί του Υ"*.
  - γ4: paidi(kostas, yannis).
  - γ5: paidi(kostas, maria).
  - γ6: paidi(anna, kostas).
  - γ7: paidi(anna, soula).
  - γ8: paidi(aris, kostas).
  - γ9: paidi(aris, soula).
- Η σχέση **paidi(X, Y)** είναι αληθής **εάν** «***ο/η Χ είναι παιδί του/της Υ***».

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Η σχέση **paidi/2** εξαρτάται από τις σχέσεις **pateras/2** και **mitera/2**. Δηλαδή, «***ο/η Χ είναι παιδί του/της Υ εάν ο Υ είναι πατέρας του/της Χ ή η Υ είναι μητέρα του/της Χ***», Δεν χρειάζεται να προσθέσουμε τα γεγονότα **γ4, γ5, γ6, γ7, γ8** και **γ9** στο πρόγραμμα 2.1 αλλά να **προσθέσουμε κανόνες οι οποίοι να εκφράζουν την εξάρτηση της σχέσης** paidi/2 από τις σχέσεις **pateras/2** και **mitera/2**.
- Οι κανόνες **κ1** και **κ2** δημιουργούν τα γεγονότα **γ4** μέχρι **γ9** χρησιμοποιώντας τα ήδη υπάρχοντα γεγονότα στο πρόγραμμα 2.1.
  - **κ1: paidi(X, Y) :- pateras(Y, X).**
  - **κ2: paidi(X, Y) :- mitera(Y, X).**
- Οι κανόνες **κ1** και **κ2** μπορούν να αντικατασταθούν από τον λογικά ισοδύναμο κανόνα **κ**.
  - **κ: paidi(X, Y) :- pateras(Y, X) ; mitera(Y, X).**

Το σύμβολο **;** είναι το λογικό ***or***.

2. Components of a Prolog program

2.4. Clauses Rules

- Η **ΒΓ** των σχέσεων οικογενείας του **Σχήματος 2.1** που έχει δημιουργηθεί μέχρι τώρα φαίνεται στο **Πρόγραμμα 2.2**.

```
pateras(yannis,kostas)
pateras(kostas,anna).
pateras(kostas,aris).
mitera(maria,kostas).
mitera(soula,anna).
mitera(soula,aris).

paidi(X, Y) :-
        pateras(Y, X).
paidi(X, Y) :-
        mitera(Y, X).
```

Πρόγραμμα 2.2: *Prolog Πρόγραμμα για τις σχέσεις του σχήματος* ***Σχήμα 1.2***.

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Η σχέση «**adelfi(anna, aris)»** σημαίνει ότι η **anna** είναι αδελφή του **aris**. Για να εκφραστεί η σχέση **γ10** με κάποιο κανόνα θα πρέπει να υπάρχει μια επιπλέον μονομελής σχέση (ή ιδιότητα) του αντικειμένου) που θα προσδιορίζει το φύλο των προσώπων (αντικειμένων).

  γ10: adelfi(anna, aris).

- Οι νέες ιδιότητες **filo_arseniko** και **filo_thiliko** λύνουν αυτό το πρόβλημα. Τα γεγονότα από **γ11** μέχρι και **γ16** πρέπει να προστεθούν στο πρόγραμμα 2.1.

  γ11: filo_arseniko(yannis). **γ12: filo_arseniko(kostas).**

  γ13: filo_arseniko(aris). **γ14: filo_thiliko(maria).**

  γ15: filo_thiliko(soula). **γ16 filo_thiliko(anna).**

- Με ένα κανόνα μπορούμε να εκφράσουμε τη σχέση **adelfos,** χρησιμοποιώντας τις σχέσεις, **adelfi/2, : filo_arseniko/1** και **filo_thiliko/1**. Ο νέος κανόνας είναι ο εξής:

  1. Ο **X** είναι αδελφός της **Y** **εάν** η **Y** είναι αδελφή του **X** και ο **X** είναι φύλου αρσενικού και η **Y** είναι φύλου θηλυκού.

  κ4: adelfos(X, Y) :- adelfi(Y, X), filo_arseniko(X), filo_thiliko(Y).

2. Components of a Prolog program

2.4. Clauses Rules

- Η **ΒΓ** των σχέσεων οικογενείας του **Σχήματος 2.1** που έχει δημιουργηθεί μέχρι τώρα φαίνεται στο **Πρόγραμμα 2.3**.

```
pateras(yannis,kostas)
pateras(kostas,anna).
pateras(kostas,aris).
mitera(maria,kostas).
mitera(soula,anna).
mitera(soula,aris).
filo_arseniko(yannis).
filo_arseniko(kostas).
filo_arseniko(aris).
filo_thiliko(maria).
filo_thiliko(soula).
filo_thiliko(anna).
```

```
andras(yannis, maria).
andras(kostas, soula).
adelfi(anna, aris).
paidi(X, Y) :- pateras(Y, X).
paidi(X, Y) :- mitera(Y, X).
gynaika(X, Y) :- andras(Y, X).
adelfos(X, Y) :-  adelfi(Y, X),
        filo_arseniko(X),
        filo_thiliko(Y).
```

***Πρόγραμμα 2.3**: Prolog Πρόγραμμα για τις σχέσεις του σχήματος ,**Σχήμα 2.1**.*

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Προτάσεις με **ίδιο όνομα κατηγορήματος** και **ίδια πληθυκότητα** στην κεφαλή της πρότασης **πρέπει να τοποθετηθούν στη σειρά η μία μετά την άλλη**. Συνήθως, **πρώτα μπαίνουν οι προτάσεις γεγονότα** και **μετά ακολουθούν οι προτάσεις κανόνες**, αυτό εξαρτάται από το πρόβλημα.
- Μια πρόταση γεγονός θεωρείται ότι έχει ίδια σύνταξη με αυτή μιας προτάσης κανόνα όπου στην κεφαλή του κανόνα είναι το γεγονός και στο σώμα του κανόνα είναι το προδεδηλωμενο κατηγόρημα ***true*** το οποίο είναι πάντα αληθές. Για παράδειγμα, η επόμενη πρόταση γεγονός
  - **"pateras(yannis,kostas)."**

  μπορεί να γραφεί ισοδύναμα σαν πρόταση κανόνας ως εξής
  - **"pateras(yannis,kostas) :- true.".**

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Στα επόμενα θα χρησιμοποιήται ο εξής συμβολισμός για να δειχθεί η πληθυκότητα ενός κατηγορήματος.
  - **όνομα_κατηγορήματος/πληθυκότητα**
- Για παράδειγμα, **pateras/2** σημαίνει **ότι η σχέση pateras έχει δύο ορίσματα**.

2. Συστατικά ενός Prolog Προγράμματος

2.4. Προτάσεις Κανόνες

- Τα **γεγονότα** και οι **κανόνες** ενός προγράμματος Prolog λέγονται ***προτάσεις* (*clauses*).**
- Όλες οι προτάσεις ενός προγράμματος **με κεφαλή ίδιο κατηγόρημα και ίδια πληθυκότητα** αποτελούν μια ***διαδικασία* (*procedure***) σε Prolog με την έννοια του διαδικαστικού προγραμματισμού.
- **Ένα Prolog πρόγραμμα αποτελείται από μια συλλογή από Prolog διαδικασίες.**
- ***Παράδειγμα***

  Έστω το πρόγραμμα Πρόγραμμα 2.3 που ορίζεται από από τις προτάσεις **{C1, C2, C3, C4}.** Οι προτάσεις **{C1, C2}** ορίζουν την διαδικασία για το κατηγόρημα **p/1**. Οι προτάσεις **{C3, C4}** ορίζουν την διαδικασία για το κατηγόρημα **q/2**.

  - **C1: p(a).**
  - **C2: p(X) :- q(X,Y), p(Y).**
  - **C3: q(b,a).**
  - **C4: q(a,a).**

  Πρόγραμμα 2.3: *Prolog πρόγραμμα με δύο διαδικασίες*

2. Συστατικά ενός Prolog Προγράμματος

2.5. Σύνθετες Ερωτήσεις

- Πολύ συχνά ερωτήσεις απαιτούν **σύζευξη** ή/και **διάζευξη** ή/και **άρνηση** κάποιων κατηγορημάτων. Αυτές οι ερωτήσεις σχηματίζονται σε Prolog με τους τελεστές **σύζευξης *and*/(,)**, **διάζευξης *or*/(;)** και **άρνησης *not*/(\+)**.
- Η ερώτηση, ***"ποιοι είναι οι γονείς της Άννας;"*** μπορεί να γραφτεί σε Prolog σαν **σύζευξη** των κατηγορημάτων **pateras** και **mitera** ως εξής .
  - **?- pateras(X, anna), mitera(Y, anna).**

2. Συστατικά ενός Prolog Προγράμματος

2.5. Σύνθετες Ερωτήσεις

- Έστω η ερώτηση, ***"είναι η Άννα συγγενής του Κώστα;"*** Εάν η Άννα είναι συγγενής του Κώστα μπορεί να είναι είτε η μητέρα του ή αδελφή του ή παιδί του.
  - **?- mitera(anna, kostas); adelfi(anna, kostas); paidi(anna, kostas).**
- Για να ικανοποιηθεί αυτή η ερώτηση από το πρόγραμμα Πρόγραμμα 2.2 θα πρέπει να ικανοποιηθεί κάποιος από τους στόχους είτε ***mitera(*anna*, kostas)*** ή ***adelfi(*anna*, kostas)*** ή ***paidi(*anna*, kostas)*.**
- Η Prolog θα προσπαθήσει να ικανοποιήσει τον πρώτο στόχο ***mitera(*anna*, kostas)*** εάν δεν ικανοποιείται θα προσπαθήσει να ικανοποιήσει τον δεύτερο στόχο ***adelfi(*anna*, kostas)*** και ούτω καθεξής. Εάν κάποιος από τους στόχους αυτής της ερώτησης ικανοποιηθεί η απάντηση της Prolog είναι **yes** διαφορετικά είναι **no**. Για το παράδειγμά μας η απάντηση είναι **yes**.

2. Συστατικά ενός Prolog Προγράμματος

2.5. Σύνθετες Ερωτήσεις

- Εάν θέλουμε να γνωρίζουμε ποιος στόχος ικανοποιήθηκε. Δηλαδή εάν ***η Άννα είναι μητέρα, αδελφή ή παιδί του Κώστα*** τότε θα τρέξουμε τον παρακάτω στόχο.
  - **?- (mitera(anna, kostas), write('Η Άννα είναι μητέρα του Κώστα')); (adelfi(anna, kostas), write('Η Άννα είναι αδελφή του Κώστα')); (paidi(anna, kostas), write('Η Άννα είναι παιδί του Κώστα')).**

2. Συστατικά ενός Prolog Προγράμματος

2.5. Σύνθετες Ερωτήσεις

- Η ερώτηση, ΄΄**ποια άλλα παιδιά έχει η Σούλα εκτός της Άννας;**΄΄ αποκλείει την Άννα από το σύνολο των απαντήσεων της Prolog. Αυτό μπορεί να εκφραστεί σε Prolog μόνο με χρήση του τελεστή άρνησης ***not/*(\+).**
  - **?- paidi(X, soula), \+(X=anna).**
- Αυτός ο στόχος για να ικανοποιηθεί θα πρέπει να ικανοποιούνται και οι δύο στόχοι.
  - Ο πρώτος στόχος ***paidi(X, soula)*** ικανοποιείται από τη βάση γνώσης του παραδείγματος Πρόγραμμα 2.2, η μεταβλητή **X** δεσμεύεται με την τιμή **anna**, δηλαδή **X=anna**.
  - Ο δεύτερος στόχος γίνεται ψευδής λόγω αυτής της τιμής του X.
    - **\+ (X=anna)**
    - **\+ (anna=anna)**
    - **\+ (αληθής)**
    - **ψευδής**
- Κατά συνέπεια, **ο δεύτερος στόχος αποκλείει την Άννα**.

2. Συστατικά ενός Prolog Προγράμματος

2.5. Σύνθετες Ερωτήσεις

- Η Prolog προσπαθεί να ικανοποιήσει ξανά τον πρώτο στόχο **"?- paidi(X, soula)."**.
  - Σ' αυτή την περίπτωση το **X** δεσμεύεται με την τιμή **X= aris**.
- Ο δεύτερος στόχος, **\+(X=anna),** γίνεται αληθής, συνεπώς η ερώτηση είναι αληθής.
  - **\+ (X=anna)**
  - **\+ (aris=anna)**
  - **\+ (ψευδής)**
  - **αληθής**
- Τελικά η Prolog δίνει σαν απάντηση **X=aris**.

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σταθερές

- Τα **δεδομένα** τα οποία υποστηρίζει η Prolog ονομάζονται ***όροι (terms)***. Ένας **όρος** μπορεί να αποτελείται από ***σταθερές***, ***μεταβλητές*** **και** ***σύνθετους όρους***.
- Οι **σταθερές** που υποστηρίζονται από την Prolog είναι **οι** ***ατομικοί όροι***, **οι** ***αριθμοί*** ***και τα*** ***ειδικά σύμβολα***.
- Οι **ατομικοί όροι** χωρίζονται σε δύο κατηγορίες:
  - **Ατομικοί όροι** που αποτελούνται μόνο από ακολουθίες γραμμάτων του λατινικού αλφάβητου, των ψηφίων 0-9 και από την υπογράμμιση. Οι ατομικοί όροι αυτής της κατηγορίας πρέπει πάντα **να αρχίζουν με πεζό γράμμα του λατινικού αλφάβητου**.
    - **maria, pateras, filo_thiliko, x__y, x1, xy12z_a, aXY**
  - **Ατομικοί όροι** που αποτελούνται από χαρακτήρες μέσα σε μονά ’ εισαγωγικά. Αυτοί οι όροι λέγονται **συμβολοσειρές**.
    - **’* X=’, ’Yannis’, ’?*!fk’, ‘elene’**

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σταθερές

- Δύο ειδών αριθμοί υποστηρίζονται από την Prolog, οι ***ακέραιοι*** και οι ***πραγματικοί***.
  - Το **διάστημα των επιτρεπτών ακεραίων** εξαρτάται από την υλοποίηση της Prolog και τον υπολογιστή στον οποίο τρέχει.
- Οι **ατομικοί όροι** και οι **αριθμοί** ονομάζονται ***απλοί όροι***.
- Τα **ειδικά σύμβολα** κατασκευάζονται μόνο από ειδικούς χαρακτήρες όπως **; : , = -** και σχηματίζουν σύμβολα όπως **:-, =:=, ;** κτλ. Η έννοια αυτών των συμβόλων είναι **προδεδηλωμένη** στην Prolog. Για παράδειγμα,
  - **":-"** είναι το σύμβολο ← (**if**),
  - **"=:="** είναι το σύμβολο της αριθμητικής ισότητας,
  - **";"** είναι το λογικό ***or*** κτλ.

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Μεταβλητές

- Οι ***μεταβλητές*** σε Prolog **αντιπροσωπεύουν αντικείμενα** τα οποία δεν μπορούμε να ονομάσουμε επειδή δεν ξέρουμε ποια αντικείμενα είναι.
- Μια μεταβλητή εφόσον **δεσμευτεί με κάποια τιμή δεν μπορεί να αλλάξει τιμή**, δηλαδή δεν μπορεί να εκπροσωπήσει δύο διαφορετικά αντικείμενα.
- Τα ονόματα των μεταβλητών είναι ακολουθίες κεφαλαίων και πεζών του λατινικού αλφάβητου, των ψηφίων 0-9 και του συμβόλου της υπογράμμισης. Το όνομα μιας μεταβλητής **πρέπει να αρχίζει** είτε με κεφαλαίο γράμμα του λατινικού αλφάβητου ή με το σύμβολο της υπογράμμισης _.
  - ***Παραδείγματα:*** **X, _adelfia, Paidia, X32, _3x, _**

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι

- Ένας ***σύνθετος ή δομημένος όρος*** είναι ένα αντικείμενο το οποίο αποτελείται από άλλα αντικείμενα τα οποία ονομάζονται **συστατικά μέρη**. Κάποιο συστατικό μέρος ενός σύνθετου όρου μπορεί με την σειρά του να αποτελείται από άλλα συστατικά μέρη.
- Ένας **σύνθετος όρος** αποτελείται από το **όνομα της συνάρτησης *(functor)*** και **τα ορίσματά της**. Ένας σύνθετος όρος σε Prolog έχει την εξής μορφή:
  - **όνομα_συνάρτησης(όρισμα$_1$, όρισμα$_2$,…., όρισμα$_κ$)**
- όπου **όρισμα$_i$ (1 ≤ i ≤ κ)** μπορεί να είναι ένας **απλός** ή **σύνθετος όρος**.

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι

- ***Παράδειγμα 1*:**

  Η διεύθυνση μιας κατοικίας μπορεί να παρασταθεί σε Prolog σαν ένας σύνθετος όρος με **4 συστατικά μέρη**, την **οδό**, τον **αριθμό**, τον **ταχυδρομικό κώδικα** και την **πόλη**. Η διεύθυνση ’’**Κύπρου 76 71201, Ηράκλειο**’’ παριστάνεται σε Prolog ως εξής:

  - **dieuthinsi(kuprou, 76, 71201, heraklion)**

- Εάν θέλαμε μια δομή για να παραστήσουμε **όλες τις διευθύνσεις του Ηρακλείου** θα είχε τη μορφή:

  - **dieuthinsi(Odos, Arithmos, Tax_kodikas, heraklion)**

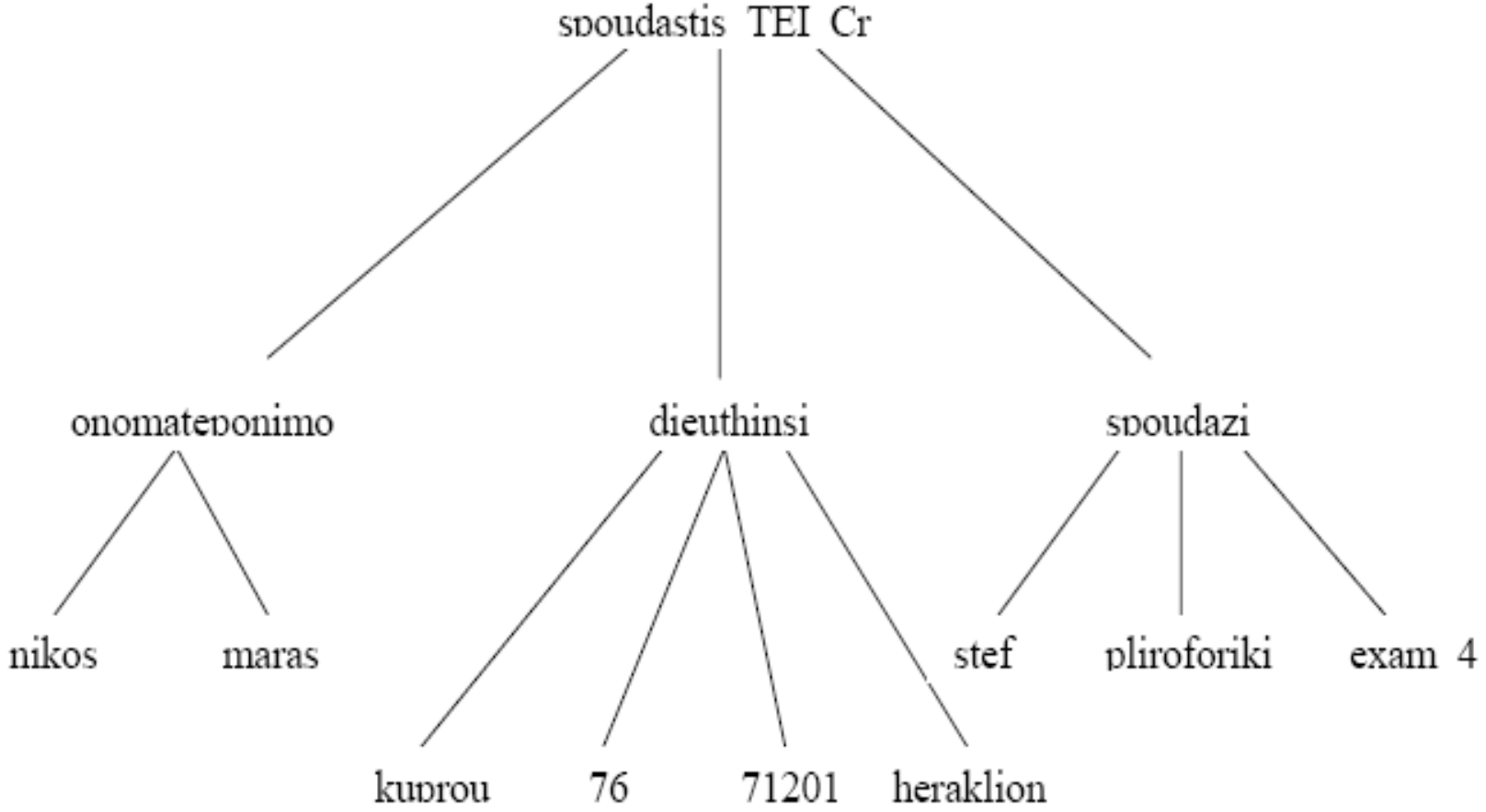
2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι

- ***Παράδειγμα 2*: Θα θέλαμε να φτιάξουμε ένα σύνθετο όρο για τους σπουδαστές του ΤΕΙ Κρήτης ο οποίος να περιέχει**
  - **1) το ονοματεπώνυμο κάθε σπουδαστή, 2)την διεύθυνσή του,**
  - **3) την σχολή, 4) το τμήμα και 5) το εξάμηνο στο οποίο ανήκει.**
- Ένας **τέτοιος όρος θα πρέπει σαν ορίσματα άλλους σύνθετους όρους**. Το κάθε όρισμα του θα είναι ένας μικρότερος όρος ο οποίος λαμβάνεται από την Prolog σαν ένα αντικείμενο. Ο σύνθετος όρος θα είναι ο εξής:
  - **spoudastis_TEI_Cr(onomateponimo(Onoma, Epitheto),**
  - **dieuthinsi(Odos, Arithmos, Tax_kodikas, heraklion),**
  - **spoudazi(Sholi, Tmima, Examino)).**
- **Για κάποιο συγκεκριμένο σπουδαστή ο σύνθετος όρος θα έχει την μορφή**,
  - **spoudastis_TEI_Cr(onomateponimo(nikos, maras),**
  - **dieuthinsi(kuprou, 76, 71201, heraklion),**
  - **spoudazi(stef, pliroforiki, exam_4o)).**

2. Συστατικά ενός Prolog Προγράμματος

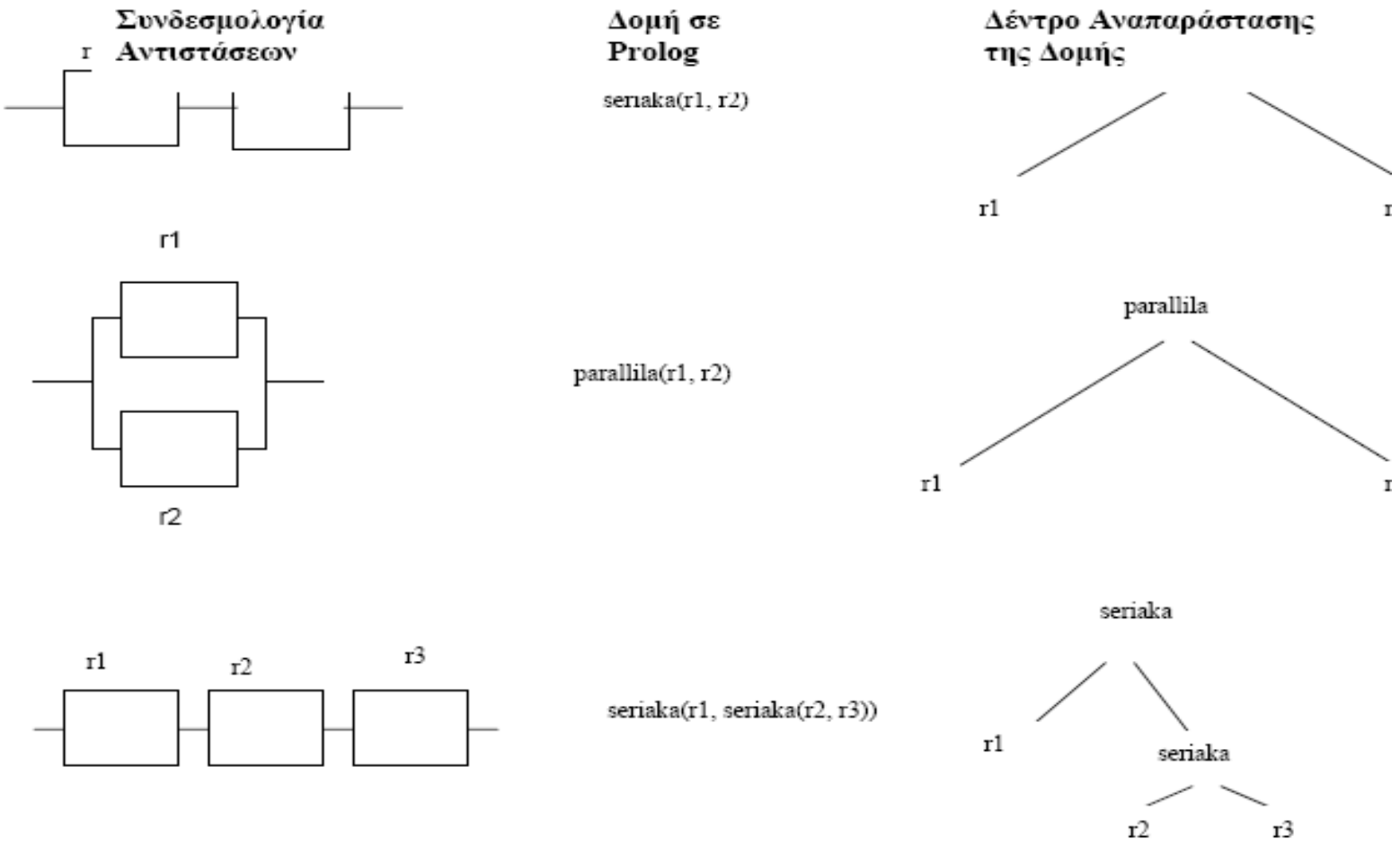
2.6. Δεδομένα: Σύνθετοι Όροι



Σχήμα 2.3:Σύνθετος όρος σε Prolog.

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι



Σχήμα 2.4:Σύνθετοι όροι για αναπαράσταση συνδεσμολογιών αντιστάσεων

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι

- ***Παράδειγμα 4***
- Έστω οι παρακάτω if-then κανόνες σε ψευδοκώδικα από κάποιο έμπειρο σύστημα διαχείρισης ποτάμιων οικοσυστημάτων.

```
if  Lab_values  = yes and Which_ones = bod then
            begin
               if BOD =< 5 then  Answer = “No Problem O.M.P. ”;
               if BOD > 5 and BOD =< 7 then  Answer = “Moderate O.M.P. ”;
               if BOD > 7 and BOD =< 15 then Answer = “Severe O.M.P. ”;
               if BOD > 15 then  Answer = “Very Severe O.M.P. ”;
            end
if  Lab_values = no and Cloudy_water = yes then
          begin
             if Colour_of_water = grey then  Answer = “Very Severe O.M.P. ”;
             if  Colour_of_water = brown then  Answer = “No Problem O.M.P. ”;
          end
```

2. Συστατικά ενός Prolog Προγράμματος

2.6. Δεδομένα: Σύνθετοι Όροι

- Το κατηγόρημα **omp(Data, Ans)** δέχεται στη μεταβλητή **Data** ένα **σύνθετο όρο με τα δεδομένα του προβλήματος** και επιστρέφει στην μεταβλητή **Ans** τη απάντηση που αφορά την μόλυνση από οργανικό υλικό στο ποτάμι**.**

```
omp(data(yes, bod, BOD_lab_value), Ans) :-
        bod(BOD_lab_value, Ans).
omp(data(no, yes, Water_colour), Ans) :-
        (Water_colour = grey,  Ans = 'Cloudy water: Very Severe OMP');
        (Water_colour = brown, Ans = 'Cloudy water: No Problem OMP').
bod(BOD_lab_value, Ans_bod) :-
        (BOD_lab_value =< 5,  Ans_bod = 'BOD: No Problem OMP');
        (BOD_lab_value > 5,  BOD_lab_value =< 7,
        Ans_bod = 'BOD: Moderate OMP');
        (BOD_lab_value > 7,  BOD_lab_value =< 15,
        Ans_bod = 'BOD: Severe OMP');
        (BOD_lab_value > 15, Ans_bod = 'BOD: Very Severe OMP').
```

Πρόγραμμα 2.4: *Χρήση σύνθετων όρων σε υλοποίηση κανόνων Έμπειρου Συστήματος.*

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- Η ***στάνταρντ είσοδος*** είναι από το **πληκτρολόγιο** και η ***στάνταρντ έξοδος*** είναι στην **οθόνη του υπολογιστή**. Τα εξής κατηγορήματα χρησιμοποιούνται για είσοδο και έξοδο σε Prolog.
- **read(X).** Αυτό το κατηγόρημα **ενοποιεί** το **X** (ή διαβάζει στο **X**) με **τον όρο από την τρέχουσα είσοδο**. Ο όρος πρέπει να τελειώνει με τελεία (.).
  - **Εάν** το όρισμα του **read/1** δεν **ενοποιείται** με την τιμή που δόθηκε **τότε** αποτυγχάνει.
  - Το κατηγόρημα **read/1** είναι **deterministic**, δηλαδή **σε περίπτωση αποτυχίας δεν οπισθοδρομεί για να διαβάσει άλλο όρο αλλά αποτυγχάνει.**

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

Παράδειγμα 1

```
?- read(X).
   |: 3.
  X = 3
?- read(X).
   |: a.
   X = a
?- read(X).
   |: ‘yannis’.
   X = yannis
```

```
?- read(X).
    |: “yannis”.
     X = [121,97,110,
     110,105,115]
?- read(ab).
     |: ab.
     yes
?- X=ab, read(X).
     |: cd.
     no
```

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- **write(X).** Αυτό το κατηγόρημα χρησιμοποιείται για **να γράψει ένα όρο X στην τρέχουσα έξοδο**.
- ***Παράδειγμα 2***

```
?- X = 3, write(X).
     X = 3
?- write([a,b,c]).
     [a,b,c]
?- write("yannis").
     [121,97,110,110,105,115]
```

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- **put_code(N).** Αυτό το κατηγόρημα εκτυπώνει στην τρέχουσα έξοδο τον **χαρακτήρα που αντιστοιχεί στον ASCII αριθμό N**.
- ***Παράδειγμα 3***

```
?- put_code(66).
    B
    yes
?- put_code(55).
    7
    yes
```

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- **get_code(X).** Αυτό το κατηγόρημα **ενοποιεί** το **X** με (ή διαβάζει στην μεταβλητή **X**) **τον ASCII αριθμό του πρώτου χαρακτήρα εισόδου**.
- ***Παράδειγμα 4***

```
?- get_code(X).
     |: 3.
     X = 51
?- get_code(X).
     |: a.
     X = 97
?- get_code(X), get_code(Y).
     |: a3.
     X=97, Y=51.
```

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- ***Παράδειγμα 5***
- Έστω οι παρακάτω **if-then** κανόνες σε ψευδοκώδικα από κάποιο έμπειρο σύστημα διαχείρισης ποτάμιων οικοσυστημάτων. ***Σημείωση***: **O.M.P.** σημαίνει **organic matter pollution** (**μόλυνση με οργανικό υλικό**). **BOD** σημαίνει **Biochemical Oxygen Demand** (**Βιοχημικές ανάγκες οξυγόνου**).

```
if Lab_values = yes then
        begin
            if BOD =< 5 then  Answer = “No Problem O.M.P. ”;
            if BOD > 5 and BOD =< 7 then
                                Answer = “Moderate O.M.P. ”;
            if BOD > 7 and BOD =< 15 then
                                Answer = “Severe O.M.P. ”;
            if BOD > 15 then  Answer = “Very Severe O.M.P. ”;
        end
if Lab_values = no then  O.M.P = unknown;
```

2. Συστατικά ενός Prolog Προγράμματος

2.7. Κατηγορήματα Εισόδου και Εξόδου

- Το κατηγόρημα **find_polution(OMP)** επιστρέφει στην μεταβλητή **OMP** τη απάντηση που αφορά την μόλυνση από οργανικό υλικό στο ποτάμι. Το πρόγραμμα διαβάζει διαλογικά τιμές για τις μεταβλητές **Lab_values** (εργαστηριακές τιμές) και **BOD**.

```
find_polution(OMP)  :-
        write('Are there lab values?  -Give yes or no- '), nl,
        read(Lab_val),       find_omp(Lab_val, OMP).
 find_omp(yes, OMP)  :-
       write('Give the value for BOD'), nl,    read(BOD),       bod(BOD, OMP).
find_omp(no, 'unknown').

bod(BOD_lab_value, Ans_bod) :-
       (BOD_lab_value =< 5,  Ans_bod = 'BOD: No Problem OMP');
       (BOD_lab_value > 5, BOD_lab_value =< 7,   Ans_bod = 'BOD: Moderate OMP');
       (BOD_lab_value > 7, BOD_lab_value =< 15,  Ans_bod = 'BOD: Severe OMP');
       (BOD_lab_value > 15, Ans_bod = 'BOD: Very Severe OMP').
```

Πρόγραμμα 2.5: *Εφαρμογή χρήσης των κατηγορημάτων read/1 write/1.*

2. Συστατικά ενός Prolog Προγράμματος

2.8. Άσκηση στη τάξη

- Να επεκτείνετε το πρόγραμμα **find_polution(OMP)** της ενότητας 2.7 το οποίο επιστρέφει στην μεταβλητή **OMP** την μόλυνση από οργανικό υλικό σε ποτάμιο οικοσύστημα. Το πρόγραμμα διαβάζει διαλογικά τιμές για τις μεταβλητές **Lab_values** (εργαστηριακές τιμές) και **BOD**. Εάν δεν υπάρχουν εργαστηριακές τιμές, τότε αποφαίνεται για τη μόλυνση στο ποτάμι από τα οπτικά χαρακτηριστικά του νερού, όπως θολότητα και χρώμα.
- Οι παρακάτω **if-then** κανόνες σε ψευδοκώδικα καθορίζουν την μόλυνση με βάση τα οπτικά χαρακτηριστικά του νερού σ' ένα ποτάμιο οικοσύστημα.
- ***Σημείωση***: **O.M.P.** σημαίνει **organic matter pollution** (**μόλυνση με οργανικό υλικό**). **BOD** σημαίνει **Biochemical Oxygen Demand** (**Βιοχημικές ανάγκες οξυγόνου**).

```
if  Lab_values = no and Cloudy_water = yes then
        begin
           if Colour_of_water = grey then  Answer = "Very Severe O.M.P. ";
           if  Colour_of_water = brown then  Answer = "No Problem O.M.P. ";
        end
if Lab_values = no and Cloudy_water = no and Colour_of_water = yes then
      if Which_colour = brown or Which_colour = tea then  Answer = "No Problem O.M.P. ";
if Lab_values = no and Cloudy_water = no and Colour_of_water = no then
        Answer = "No Problem O.M.P. ";
```

Τέλος Διάλεξης

Ευχαριστώ!

Ερωτήσεις;