

1. ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ PROLOG

1.1. Εισαγωγή

Για να γράψουμε ένα πρόγραμμα στην PROLOG, πρέπει να εκφράσουμε το πρόβλημα που θέλουμε να επιλύσουμε με τη μορφή:

Αντικείμενα <---> Σχέσεις ανάμεσα στα αντικείμενα

Η περιγραφή των αντικειμένων και των σχέσεων που ισχύουν μεταξύ τους γίνεται με τη βοήθεια ειδικών δομικών μηχανισμών που η PROLOG δανείζεται από τη Μαθηματική Λογική. Οι μηχανισμοί αυτοί είναι: οι **φράσεις (clauses)** και η μοναδική δομή δεδομένων που ονομάζεται **λογικός όρος (logical term)**. Υπάρχουν τριών ειδών φράσεις: τα **γεγονότα (facts)**, οι **κανόνες (rules)** και οι **ερωτήσεις (queries)**. Έτσι ο προγραμματισμός στην PROLOG συνίσταται:

- (α) στον ορισμό των αντικειμένων και των σχέσεων που ισχύουν μεταξύ τους με τη βοήθεια γεγονότων και κανόνων και
- (β) υποβολή ερωτήσεων γύρω από τις σχέσεις αυτές.

Ένα πρώτο πρόγραμμα γραμμένο στην PROLOG δίνεται παρακάτω. Το πρόγραμμα αυτό περιγράφει τις σχέσεις **father**, **mother**, **parent** και **grandparent** που μπορούν να ισχύουν ανάμεσα στα αντικείμενα **uranus**, **kronos**, **demeter**, **zeus**, **athena**, **artemis**, **apollo**, **gaia**, **persephone**, και **rhea**.

Φ1:	father(uranus,kronos).
Φ2:	father(kronos,demeter).
Φ3:	father(kronos,zeus).
Φ4:	father(zeus,athena).
Φ5:	father(zeus,artemis).
Φ6:	father(zeus,apollo).
Φ7:	mother(gaia,kronos).
Φ8:	mother(demeter,persephone).

Φ9: **mother(rhea,zeus).**

Φ10: **parent(X,Y) :- father(X,Y).**
Φ11: **parent(X,Y) :- mother(X,Y).**
Φ12: **grandfather(X,Y) :- father(X,Z), parent(Z,Y).**

Φ13: **?- father(uranus, kronos).**
Φ14: **?- father(X, demeter).**
Φ15: **?- grandfather(X, Y).**

πρόγραμμα 1.1

1.2 Γεγονότα

Η πρώτη μορφή φράσης στην PROLOG είναι το **γεγονός**. Στην πιο απλή περίπτωση ένα πρόγραμμα αποτελείται από ένα πεπερασμένο αριθμό γεγονότων. Γενικά ένα γεγονός στην PROLOG δηλώνει ότι κάποια **σχέση** (**relation**) υφίσταται ανάμεσα σε κάποιες οντότητες-αντικείμενα (ή καλύτερα ότι η σχέση αυτή είναι αληθής). Η σχέση ονομάζεται **κατηγόρημα** (**predicate**) τα δε αντικείμενα ονομάζονται **ορίσματα** (**arguments**) του κατηγορήματος. Ο αριθμός των ορισμάτων αναφέρεται σαν **πληθικός αριθμός** (**arity**) του κατηγορήματος. Ισοδύναμα μπορούμε να πούμε, ότι αν **N** είναι ο πληθικός αριθμός ενός κατηγορήματος, ένα γεγονός καθορίζει τις εξαρτήσεις των **N**-άδων που ικανοποιούν τη συγκεκριμένη σχέση. Στην PROLOG είναι δυνατόν κάποια κατηγορήματα να έχουν το ίδιο όνομα, αλλά διαφορετικούς πληθικούς αριθμούς. Επομένως όταν θέλουμε να αναφερθούμε με ακρίβεια σε κάποιο κατηγορημα πρέπει να αναφέρουμε και το όνομά του και τον πληθικό του αριθμό. (Χρησιμοποιείται η εξής σημειογραφία: **p/N**, όπου **p** το όνομα του κατηγορήματος και **N** ο πληθικός του αριθμός.)

Στο πρόγραμμα 1.1, οι φράσεις (Φ1)-(Φ9) είναι γεγονότα. Στη φράση (Φ2), για παράδειγμα, οι σταθερές **kronos** και **demeter** παίζουν το ρόλο των αντικειμένων και **father** είναι η σχέση που υφίσταται μεταξύ τους. Η ερμηνεία που μπορούμε να δώσουμε στη φράση αυτή είναι ίσως προφανής:

"Ο Κρόνος είναι πατέρας της Δήμητρας".

Από συντακτική άποψη ισχύουν τα εξής:

- ♦ Τα ονόματα των σχέσεων και των αντικειμένων πρέπει να αρχίζουν από ένα μικρό γράμμα του αλφαβήτου.
- ♦ Η σχέση γράφεται πρώτη, και τα αντικείμενα γράφονται μέσα σε ένα ζευγάρι παρενθέσεων που ακολουθεί τη σχέση, χωρίζονται δε μεταξύ τους με τη βοήθεια κομμάτων ",".
- ♦ Ο χαρακτήρας "." τοποθετείται στο τέλος του γεγονότος.

Η συντακτική γραφή των προγραμμάτων της PROLOG διαφέρει από υλοποίηση σε υλοποίηση της γλώσσας. Αυτό συμβαίνει γιατί καθυστέρησε πολύ η διαδικασία τυποποίησης της γλώσσας. Μόλις το 1996 ολοκληρώθηκε ο κατά ISO καθορισμός των standards της PROLOG (ISO/IEC JTC1 SC22 WG17). Στα παραπάνω και σε ότι ακολουθεί χρησιμοποιούμε το standard αυτό συντακτικό που μοιάζει να έχει γίνει κοινά αποδεκτό [DERANSART & al, 1996].

Μία σχέση (ή ισοδύναμα το κατηγορημα που αντιστοιχεί σ' αυτή) μπορεί να οριστεί με τη βοήθεια περισσοτέρων του ενός γεγονότων. Έτσι στην περίπτωση του προγράμματος 1 τα γεγονότα (Φ1) έως και (Φ6) ορίζουν την σχέση **father** και τα γεγονότα (Φ7) έως και (Φ9) τη σχέση **mother**.

Μερικά επιπλέον παραδείγματα γεγονότων (καθώς και η ελεύθερη ερμηνεία τους) δίνονται παρακάτω:

γ1:	integer(5).	-->	Το 5 είναι ακέραιος
γ2:	likes(john,mary).	-->	Ο Γιάννης αρέσει τη Μαίρη
γ3:	language(greek).	-->	Τα Ελληνικά είναι μία γλώσσα
γ4:	plays(nick,basket).	-->	Ο Νίκος παίζει μπάσκετ
γ5:	different(5,6).	-->	Το 5 είναι διαφορετικό από το 6
γ6:	it_rains.	-->	Βρέχει
γ7:	vehicle(ship).	-->	Το πλοίο είναι μέσο μεταφοράς
γ8:	ship(knossos).	-->	Το Κνωσός είναι πλοίο

γ9: greek(knossos). --> Το Κνωσός είναι Ελληνικό
γ10: programming_language(prolog).
 --> Η PROLOG είναι γλώσσα προγραμματισμού
γ11: book(prolog,bratko,addison_wesley).
 --> Το βιβλίο με τίτλο PROLOG έχει
 συγγραφέα τον Bratko και εκδόθηκε από
 τον οίκο ADDISON WESLEY
γ12: lives(name(nick, antoniou), address(Kriezotou, 15)).
 --> Ο Νίκος Αντωνίου κατοικεί στην οδό
 Κριεζώτου 15

Υπάρχει περίπτωση ένα γεγονός να αναφέρεται σε ένα κατηγορήμα με πληθικό αριθμό 0, όπως αυτό συμβαίνει με το γεγονός (γ6).

Ενα γεγονός μπορεί να δηλώνει ότι κάποια οντότητα έχει μία συγκεκριμένη ιδιότητα ή τύπο, στην περίπτωση που το αντίστοιχο κατηγορήμα που την ορίζει έχει μόνο ένα όρισμα. Για παράδειγμα το γεγονός (γ1) δηλώνει ότι το 5 έχει την ιδιότητα *ακέραιος* ή ισοδύναμα ότι το 5 είναι τύπου *integer*. Ιδιότητες δηλώνουν επίσης τα γεγονότα (γ3) και (γ10). Σε ένα αντικείμενο μπορούμε να προσδώσουμε περισσότερες από μία ιδιότητες, όπως αυτό φαίνεται από τα γεγονότα (γ8) και (γ9). Επίσης ένας τύπος μπορεί να έχει υπο-τύπους όπως φαίνεται από τα γεγονότα (γ7) και (γ8).

Όταν χρησιμοποιούμε γεγονότα, για να ορίσουμε κάποια σχέση ανάμεσα σε δύο ή περισσότερα αντικείμενα χρειάζεται ιδιαίτερη προσοχή στη σειρά τοποθέτησης των ορισμάτων του αντίστοιχου κατηγορήματος. Στην ουσία η PROLOG δεν απαιτεί συγκεκριμένη σειρά, αλλά εμείς πρέπει να επιλέξουμε κάποια και να τη διατηρούμε, για να είμαστε συνεπείς με την ερμηνεία των γεγονότων. Για παράδειγμα στο γεγονός (γ11) επιλέξαμε το πρώτο όρισμα του κατηγορήματος *book* να παίζει το ρόλο του τίτλου του βιβλίου, το δεύτερο όρισμα να παίζει το ρόλο του συγγραφέα και το τρίτο όρισμα να παίζει το ρόλο του εκδοτικού οίκου. Επίσης στο γεγονός (γ2) επιλέξαμε τη σειρά των ορισμάτων με τέτοιο τρόπο, ώστε να εννοούμε ότι "ο Γιάννης αρέσει τη Μαίρη" και όχι ότι "η Μαίρη αρέσει το Γιάννη". Αυτό σημαίνει ότι το γεγονός *likes(john,mary)* είναι διαφορετικό από το γεγονός *likes(mary,john)*.

Ο τρόπος περιγραφής μιάς ή περισσότερων σχέσεων με τη βοήθεια κάποιων γεγονότων δεν είναι φυσικά μοναδικός. Σαν παράδειγμα ας θεωρήσουμε τα γεγονότα:

```
ship(knossos).  
gray(knossos).  
big(knossos). (I)
```

τα οποία χρησιμοποιούμε για να περιγράψουμε ότι:

"το Κνωσός είναι ένα μεγάλο πλοίο το χρώμα του οποίου είναι γκρι".

Ισοδύναμα για την περιγραφή αυτή θα μπορούσαμε να χρησιμοποιήσουμε τα παρακάτω γεγονότα:

```
ship(knossos).  
color(knossos,gray).  
size(knossos,big). (II)
```

Επίσης η PROLOG δεν επιβάλλει κανένα περιορισμό στην επιλογή του ρόλου που μπορεί να παίξει κάποιο κατηγορήμα σε σχέση με τα ορίσματα του. Παρόλλα αυτά θα μπορούσαμε να πούμε ότι μπορούμε να χρησιμοποιούμε την άτυπη αρχή "στη θέση του κατηγορήματος πρέπει να αντιστοιχούμε την πιά γενική έννοια". Ετσι για να εκφράσουμε τη σχέση: "το πλοίο είναι μέσο μεταφοράς", η επιλογή του γεγονότος **vehicle(ship)** είναι καλύτερη από το γεγονός **ship(vehicle)**.

Η επιλογή του τρόπου αναπαράστασης της πληροφορίας αποτελεί στόχο-κλειδί της Τεχνητής Νοημοσύνης. Το πώς δηλαδή η πληροφορία που σχετίζεται με κάποια κοινά αποδεκτή γνώση (*common sense knowledge*) μπορεί να μετατραπεί σε μία ρητή περιγραφή με τη βοήθεια κάποιου προγραμματιστικού εργαλείου (της PROPOG στη συγκεκριμένη περίπτωση).

1.3 Κανόνες

Η δεύτερη μορφή φράσης για την PROLOG είναι ο **κανόνας (rule)**. Με τη βοήθεια κάποιων γνωστών σχέσεων και με τη χρήση κανόνων μπορούμε να ορίσουμε καινούργιες σχέσεις ανάμεσα στα αντικείμενα. Οι φράσεις (**Φ10**)-(**Φ12**) του προγράμματος 1, είναι κανόνες. Στη φράση (**φ12**) για παράδειγμα, ορίζουμε τη σχέση **grandparent** με τη βοήθεια των σχέσεων **father** και **parent**. Η φράση δηλώνει ότι:

- ο **X** είναι παππούς του **Y** **εαν**
- ο **X** είναι πατέρας του **Z** **και** ο **Z** είναι γονιός του **Y**.

Ενώ τα γεγονότα δηλώνουν σχέσεις που αληθεύουν αδιαφιλονίκητα, οι κανόνες δηλώνουν σχέσεις οι οποίες αληθεύουν μόνο με την προϋπόθεση ότι ισχύουν κάποιες συνθήκες. Οι συνθήκες αυτές είναι με τη σειρά τους σχέσεις. Γενικά ένας κανόνας αποτελείται:

- (**α**) από το τμήμα των **συνθηκών (conditions)** ή **υποθέσεων** (το δεξί τμήμα του κανόνα) και
- (**β**) από το **συμπέρασμα (conclusion)** (το αριστερό τμήμα του κανόνα).

Το συμπέρασμα ονομάζεται επίσης και **κεφαλή (head)** του κανόνα, ενώ το τμήμα συνθηκών ονομάζεται και **σώμα (body)** του κανόνα. Οι συνθήκες που αποτελούν το σώμα ενός κανόνα μπορούν να αναφέρονται και σαν **στόχοι (goals)**.

Από συντακτική άποψη ισχύουν τα εξής:

- ♦ Η κεφαλή χωρίζεται από το σώμα με το σύμβολο **":-** που διαβάζεται **εάν**.
- ♦ Οι συνθήκες του σώματος (εάν είναι περισσότερες από μία) χωρίζονται μεταξύ τους με κόμματα **","**
- ♦ Ο χαρακτήρας **."** (τελεία) τοποθετείται στο τέλος του κανόνα.

Το κόμμα **","** ανάμεσα σε δύο συνθήκες παίζει το ρόλο της **σύζευξης (conjunction)** και ερμηνεύεται σαν λογικό **ΚΑΙ (AND)**. Αυτό σημαίνει ότι όλες

οι συνθήκες του σώματος ενός κανόνα πρέπει να συναληθεύουν για να αληθεύει το συμπέρασμα του κανόνα.

Με τη χρήση μεταβλητών στη θέση των ορισμάτων των κατηγορημάτων που βρίσκονται στη κεφαλή ή/και στο σώμα ενός κανόνα μπορούμε να ορίσουμε σχέσεις που αναφέρονται όχι μόνο σε κάποιο συγκεκριμένο αντικείμενο αλλά σε μία ολόκληρη κατηγορία αντικειμένων. (Από συντακτική άποψη οι μεταβλητές αρχίζουν με ένα κεφαλαίο γράμμα του λατινικού αλφαβήτου.) Για παράδειγμα με τον κανόνα:

likes(john, X) :- studies(X,computer_science).

ορίζουμε ότι ο Γιάννης αρέσει όλους εκείνους (τους **X**) που σπουδάζουν Επιστήμη Υπολογιστών. Γενικά οι μεταβλητές που χρησιμοποιούνται σαν ορίσματα στο κατηγορήμα που αποτελεί την κεφαλή μίας φράσης, θεωρούνται **κοθολικά ποσοδεικτούμενες**, προσλαμβάνουν δηλαδή την έννοια του "για κάθε" (\forall). Έτσι ο παραπάνω κανόνας μπορεί να διαβαστεί:

Για κάθε **X**,
ο **john** αρέσει τον **X** εάν
ο **X** σπουδάζει **computer_science**.

Οι μεταβλητές που χρησιμοποιούνται σαν ορίσματα σε κατηγορήματα που βρίσκονται στο σώμα ενός κανόνα, χωρίς όμως να χρησιμοποιούνται στην κεφαλή του, θεωρούνται **υπαρξιακά ποσοδεικτούμενες**, προσλαμβάνουν δηλαδή την έννοια του "υπάρχει" (\exists). Έτσι ο κανόνας (**Φ12**) του προγράμματος 1, μπορεί να διαβαστεί:

Για κάθε **X, Y**,
ο **X** είναι **παππούς** του **Y**, εάν
υπάρχει **Z** τέτοιο ώστε:
ο **X** να είναι **πατέρας** του **Z** και ο **Z** να είναι **γονιός** του **Y**.

Όπως και στην περίπτωση των γεγονότων, μπορούμε να χρησιμοποιήσουμε περισσότερους από έναν κανόνες, οι οποίοι έχουν στην κεφαλή τους το ίδιο κατηγορήμα, για να ορίσουμε μία σχέση. Έτσι για παράδειγμα οι σχέσεις (**Φ10**)

και (Φ11) του προγράμματος 1, χρησιμοποιούνται για τον ορισμό της σχέσης **parent**. Στη γενικότερη περίπτωση μία σχέση μπορεί να οριστεί από κοινού, με τη χρήση ενός αριθμού κανόνων και ενός αριθμού γεγονότων. Λέμε επίσης ότι το σύνολο των φράσεων αυτών (κανόνων και γεγονότων) ορίζουν μια **διδικασία (procedure)**.

Τα γεγονότα και οι κανόνες αποτελούν τη **βάση γνώσης (knowledge base)** της PROLOG που ονομάζεται επίσης και **χώρος εργασίας (workspace)**.

1.3.1 Αναδρομικοί Κανόνες

Οι κανόνες που αναφέραμε μέχρι τώρα ορίζουν νέες σχέσεις με τη βοήθεια κάποιων άλλων. Μία ενδιαφέρουσα επέκταση είναι η δυνατότητα δημιουργίας **αναδρομικών κανόνων (recursive rules)**, κανόνων δηλαδή που ορίζουν σχέσεις με τη βοήθεια του ευατού τους. Αυτό σημαίνει ότι ένα κατηγορημα μπορεί να εμφανίζεται και στην κεφαλή αλλά και στο σώμα ενός κανόνα. Ας θεωρήσουμε σαν παράδειγμα το παρακάτω πρόγραμμα:

```
L1:   located_in(soufli, n_alexandroupoli).
L2:   located_in(petrohori, n_xanthi).
L3:   located_in(toxotes,n_xanthi).
L4:   located_in(sindos,n_thessaloniki).

L5:   located_in(X,thraki) :- located_in(X,n_alexandroupoli).
L6:   located_in(X,thraki) :- located_in(X,n_xanthi).
L7:   located_in(X,makedonia) :- located_in(X,n_thessaloniki).

L8:   located_in(X,northGreece) :- located_in(X,thraki).
L9:   located_in(X,northGreece) :- located_in(X,makedonia).
```

πρόγραμμα 1.2

Το πρόγραμμα αυτό ορίζει ένα μόνο κατηγορημα (**located_in**) το οποίο περιγράφει τη σχέση μίας πόλης με κάποια γεωγραφικά διαμερίσματα. Με τη βοήθεια των γεγονότων (**L1**)-(**L4**) περιγράφεται η σχέση ότι κάποιες πόλεις ανήκουν σε κάποιο νομό. Στη συνέχεια με τη βοήθεια των αναδρομικών κανόνων (**L5**)-(**L9**) η σχέση αυτή επεκτείνεται για μεγαλύτερα γεωγραφικά διαμερίσματα, όπως "μία πόλη **X** ανήκει στη Μακεδονία, εάν ανήκει στο νομό Θεσσαλονίκης" (φράση **L7**) ή "μία πόλη **X** ανήκει στη Βόρεια Ελλάδα, εάν ανήκει στη Μακεδονία (φράση **L9**).

Είναι φυσικά δυνατό να χρησιμοποιήσουμε από κοινού αναδρομικούς και μή αναδρομικούς κανόνες, για να ορίσουμε μία σχέση, όπως για παράδειγμα το κατηγορημα **ancestor** (=απόγονος) που δίνεται παρακάτω και μπορεί να συμπληρώσει το πρόγραμμα 1:

```
ancestor(X,Y) :- parent(X,Y).  
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
```

1.4 Ερωτήσεις

Η ενεργοποίηση ενός προγράμματος της PROLOG γίνεται με τη βοήθεια μιας **ερώτησης**. Η ερώτηση είναι η τρίτη μορφή φράσης και μπορεί να αποτελείται από έναν ή περισσότερους **στόχους (goals)**. Οι φράσεις (**Φ13**)-(**Φ15**) του προγράμματος 1 είναι ερωτήσεις. Από συντακτική άποψη η ερώτηση γράφεται ακριβώς όπως το σώμα ενός κανόνα με τη διαφορά ότι προηγείται το σύμβολο "?-". (Το σύμβολο αυτό εκτυπώνεται συνήθως αυτόματα από το σύστημα της PROLOG όταν είμαστε σε κατάσταση ερώτησης). Η απάντηση στην ερώτηση ισοδυναμεί με την εξακρίβωση του εάν η ερώτηση είναι μία **λογική συνεπαγωγή** του προγράμματος. Για παράδειγμα με την ερώτηση (**Φ13**) του προγράμματος 1, ρωτάμε κατά πόσο ο στόχος **grandparent(uranus,kronos)** αληθεύει σε σχέση με τις φράσεις (**Φ1**) ως (**Φ12**). Η απάντηση είναι "ναί":

```
?- father(uranus,kronos).  
yes
```

Τη θέση ενός ορίσματος του στόχου μπορεί να την πάρει μία μεταβλητή, όπως για παράδειγμα:

```
?- father(X,demeter).
```

```
X = kronos ->
```

Η ερώτηση αυτή ερμηνεύεται ως εξής:

"Να βρεθεί ένα **X** τέτοιο ώστε

η σχέση **father(X,demeter)** να είναι αληθής"

Βασικό χαρακτηριστικό της PROLOG είναι ότι μία μεταβλητή μπορεί να πάρει τη θέση **οποιοδήποτε** ορίσματος ενός στόχου. Έτσι ενώ με την προηγούμενη ερώτηση αναζητούσαμε, στην ουσία, τον πατέρα της Δήμητρας, μπορούμε επίσης τοποθετώντας μεταβλητή στη θέση του δευτέρου ορίσματος, να αναζητήσουμε τα παιδιά του Κρόνου:

```
?- father(kronos,X).
```

```
X = demeter ->1
```

```
X = zeus ->
```

```
no
```

Στην παραπάνω ερώτηση παρατηρούμε ένα άλλο βασικό χαρακτηριστικό της PROLOG: ότι μία ερώτηση μπορεί να επιδέχεται περισσότερες από μία απαντήσεις τις οποίες ο μηχανισμός εκτέλεσης της PROLOG μπορεί να βρει.

Οι ερωτήσεις μπορούν να περιέχουν επίσης περισσότερες από μία μεταβλητές όπως για παράδειγμα:

```
?- parent(X,Y).
```

¹ Όταν η PROLOG δώσει την πρώτη απάντηση σε μία ερώτηση ο χρήστης έχει τις εξής δύο δυνατότητες:

(α) Εάν δεν ενδιαφέρεται για άλλες πιθανές απαντήσεις να πληκτρολογήσει ένα "**carriage return**"

(β) Να πληκτρολογήσει ένα ερωτηματικό (";") ακολουθούμενο από ένα "**carriage return**", στην περίπτωση που ενδιαφέρεται για εναλλακτικές απαντήσεις. Στην περίπτωση αυτή το σύστημα της PROLOG δίνει την επόμενη απάντηση ή απαντά αρνητικά ("no"), εάν αυτή δεν υπάρχει.

```

X = uranus      Y = kronos  ->;
X = kronos      Y = demeter ->;
X = kronos      Y = zeus    ->;
X = zeus        Y = athena  ->;
X = zeus        Y = artemis ->;
X = zeus        Y = apollo  ->;
X = gaia        Y = kronos  ->;
X = demeter     Y = persephone ->;
X = rhea        Y = zeus    ->;
no

```

Στην παραπάνω ερώτηση αναζητούμε τα ζευγάρια **X**, **Y** για τα οποία η σχέση **parent(X,Y)** είναι αληθής. Γενικά με μία ερώτηση που περιέχει μεταβλητές ρωτάμε εάν υπάρχουν κάποιες τιμές για αυτές τις μεταβλητές που κάνουν την ερώτηση λογική συνεπαγωγή του προγράμματος.

Μια ερώτηση μπορεί να περιέχει περισσότερους από ένα στόχους, όπως για παράδειγμα:

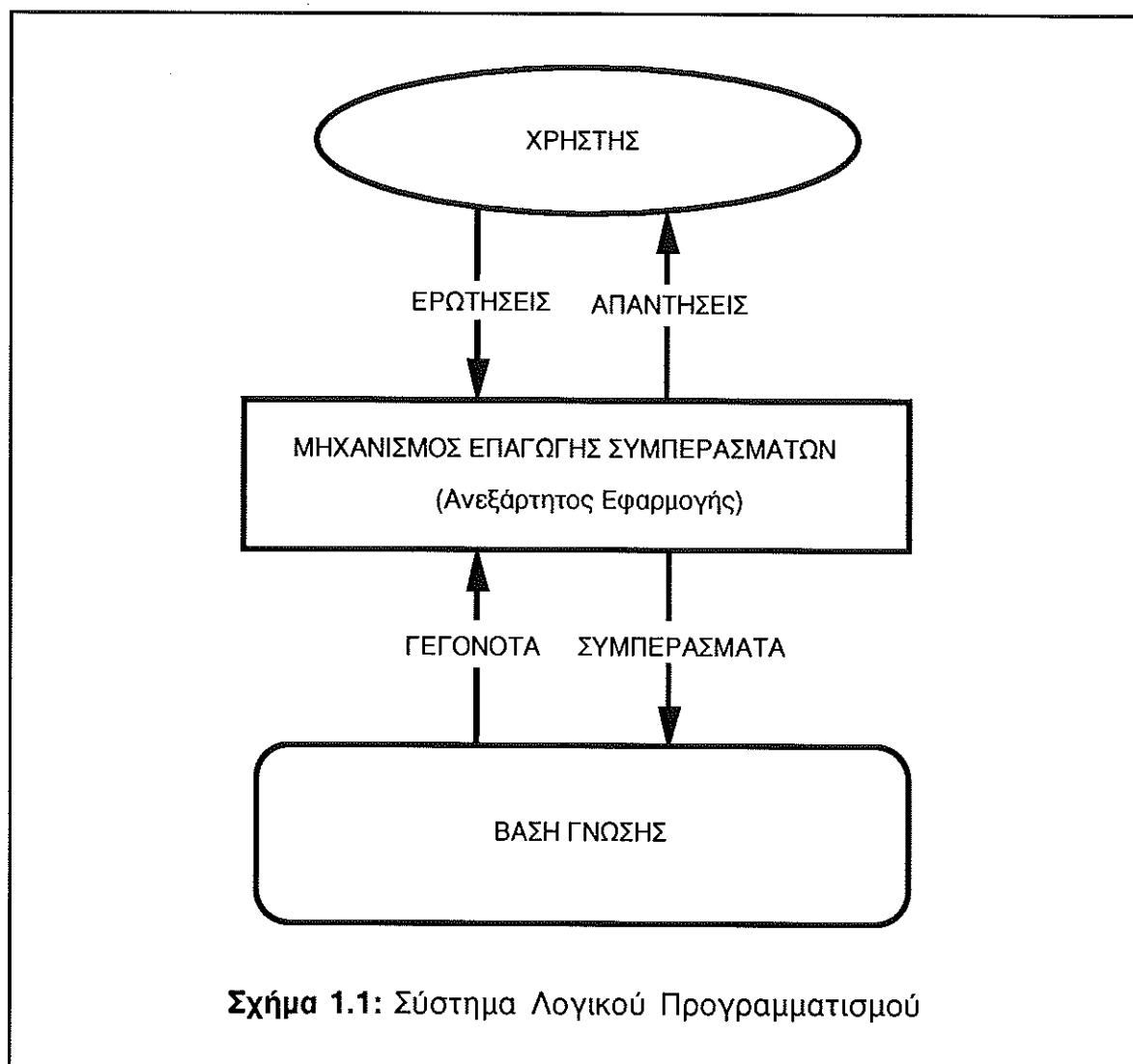
```

?- father(X,Y), mother(Z,Y).
X = uranus  Y = kronos  Z = gaia  ->

```

Στην περίπτωση αυτή ρωτάμε ποιός είναι ο πατέρας **X** και η μητέρα **Z** που έχουν το ίδιο παιδί **Y**. Γενικά σε μία ερώτηση που περιέχει περισσότερους από ένα στόχους, η PROLOG αναζητά εκείνες τις λύσεις που κάνουν τους στόχους αυτούς να συναληθεύουν.

Θα πρέπει να παρατηρήσουμε ότι στην PROLOG για να απαντηθεί μία ερώτηση, απαιτείται συχνά να ακολουθηθεί μία πολύπλοκη διεργασία, στην οποία εμπλέκεται μία διαδικασία επαγωγής συμπερασμάτων, αναζήτηση εναλλακτικών λύσεων και πιθανά διαδικασία οπισθοδρόμησης σε περίπτωση αποτυχίας. Όλη αυτή η διεργασία, γίνεται κατά κανόνα αυτόματα από το σύστημα της PROLOG χωρίς να εμπλέκεται ο χρήστης-προγραμματιστής (σχ. 1.1). Η διεργασία αυτή θα εξεταστεί με περισσότερη λεπτομέρεια στο επόμενο κεφάλαιο.



1.5 Ένας πιο Τυπικός Ορισμός των Φράσεων της PROLOG

Ένα πρόγραμμα στην PROLOG είναι ένα πεπερασμένο σύνολο από **φράσεις** (*clauses*). Οι φράσεις έχουν την παρακάτω γενική μορφή:

$$B \text{ :- } A_1, A_2, \dots, A_n. \quad \text{με } n \geq 0 \quad (1)$$

όπου τα B, A_1, A_2, \dots, A_n είναι **ατομικοί τύποι** (*atomic formulas*). Ένας ατομικός τύπος είναι μια έκφραση της μορφής:

$$p(t_1, t_2, \dots, t_m) \quad \text{με } m \geq 0 \quad (2)$$

Το p ονομάζεται m -θέσιο **κατηγόρημα** (*predicate*) (συμβολίζεται p/m) και τα t_1, t_2, \dots, t_m είναι οι **όροι** (*terms*) του κατηγορήματος. Ο όρος μπορεί να είναι μια σταθερά, μια μεταβλητή ή μια έκφραση της μορφής:

$$f(t_1, t_2, \dots, t_k) \quad \text{με } k > 0^2 \quad (3)$$

Το f ονομάζεται k -θέσιο **συναρτησιακό σύμβολο** (*functional symbol*) και τα t_1, t_2, \dots, t_k είναι με τη σειρά τους όροι. Οι όροι t_1, t_2 , κ.λ.π. των (2) και (3) ονομάζονται και **ορίσματα** (*arguments*) του κατηγορήματος ή του συναρτησιακού συμβόλου αντίστοιχα. Η έκφραση (3) ονομάζεται και **συναρτησιακός όρος** (*functional term*). Το κόμμα "," ανάμεσα σε δύο ατομικούς τύπους A_i, A_{i+1} της φράσης (1) παίζει το ρόλο της σύζευξης "Λ" (λογικό ΚΑΙ). Το σύμβολο ":-" παίζει το ρόλο της συνεπαγωγής " \leq " και διαβάζεται **εαν**. Το B ονομάζεται **κεφαλή** της φράσης και τα $\{A_1, A_2, \dots, A_n\}$ αποτελούν το **σώμα** της.

Γενικά οι φράσεις (1) στην PROLOG μπορούν να πάρουν μία από τις τρεις παρακάτω μορφές:

- ♦ $B.$ (1α)
- ♦ $B :- A_1, A_2, \dots, A_n.$ (1β)
- ♦ $:- A_1, A_2, \dots, A_n.$ (1γ)

Οι φράσεις του τύπου (1α) (φράσεις με κενό σώμα) ονομάζονται **γεγονότα**.

Οι φράσεις του τύπου (1β) ονομάζονται **κανόνες**. Το B ερμηνεύεται σαν το **συμπέρασμα** του κανόνα και οι A_1, A_2, \dots, A_n ερμηνεύονται σαν οι **υποθέσεις** του κανόνα.

Οι φράσεις του τύπου (1γ) (φράσεις χωρίς κεφαλή) ονομάζονται **ερωτήσεις** (*queries*) ή **στόχοι** (*goals*). Οι ατομικοί τύποι που αποτελούν ένα στόχο

² Εάν θέσουμε $K=0$ μπορούμε να θεωρήσουμε ότι έχουμε μία σταθερά (συναρτησιακό όρο με πληθικό αριθμό 0).

ονομάζονται και **υποστόχοι**. Μια ερώτηση :- Α. μπορεί εναλλακτικά να γράφεται και ?- Α.³

Το σύνολο των φράσεων του τύπου (1α) και (1β) που έχουν σαν κεφαλή τους το ίδιο κατηγορήμα ορίζουν μια **διαδικασία (procedure)**.

Παρατήρηση :

Στην PROLOG ο τρόπος κατασκευής των ατομικών τύπων (κατηγορημάτων), από συντακτική άποψη, είναι ακριβώς ίδιος με αυτόν των όρων. Για το λόγο αυτό πολλές φορές ένας ατομικός τύπος μπορεί να αναφέρεται και απλά σαν όρος (*term*). 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 σαν παράμετροι στις κλήσεις των κατηγορημάτων.

1.6 Ασκήσεις

1] Μεταφράστε τις παρακάτω προτάσεις στην PROLOG:

(α) Όλοι (ο καθένας) όσοι έχουν παιδιά είναι χαρούμενοι.

(β) Για όλους τους X, εάν ο X έχει ένα παιδί, το οποίο έχει μια αδελφή, τότε ο X έχει δύο παιδιά.

2] "Παντρεύτηκα μιά χήρα (ας την ονομάσουμε w). Ο πατέρας μου (f), που μας επισκέπτονταν συχνά, ερωτεύτηκε τη θετή μου κόρη (k) και την παντρεύτηκε. Έτσι ο πατέρας μου έγινε θετός γιός μου και η θετή μου κόρη έγινε μητέρα μου. Μερικούς μήνες αργότερα, η γυναίκα μου απέκτησε ένα γιό (s1), ο οποίος έγινε θετός αδελφός του πατέρα μου, καθώς επίσης και θείος μου. Η γυναίκα του πατέρα μου (δηλαδή η θετή μου κόρη) είχε επίσης ένα γιό (s2). "

[Algorithms+Data Structures = Programs, N. Wirth, 1976]

³ Η ακριβής σημασία των δύο παραλλαγών είναι πιθανόν να διαφέρει από σύστημα σε σύστημα της PROLOG

Να γραφεί πρόγραμμα με τη βοήθεια της PROLOG που να αναπαριστά τις σχέσεις που περιγράφονται στο παραπάνω πρόβλημα. Στη συνέχεια με βάση το πρόγραμμα αυτό να ελεγχθεί ο ισχυρισμός:

"Είμαι ο παππούς μου;"

3].

(α) Μεταφράστε τις παρακάτω προτάσεις σε φράσεις της PROLOG:

- ♦ Η Εύη είναι όμορφη.
- ♦ Ο Αγγελος είναι πλούσιος και ευπαρουσίαστος.
- ♦ Η Αλεξάνδρα είναι πλούσια και δυναμική.
- ♦ Ο Στέργιος είναι δυναμικός και ευπαρουσίαστος.
- ♦ Ο Μιχάλης είναι ευγενικός και δυναμικός.
- ♦ Όλοι οι άνδρες αρέσουν τις όμορφες γυναίκες.
- ♦ Όλοι οι πλούσιοι άνδρες είναι ευτυχισμένοι.
- ♦ Εάν ένας άνδρας αρέσει μία γυναίκα, που τρέφει τα ίδια αισθήματα γι' αυτόν, τότε είναι ευτυχισμένος.
- ♦ Εάν μία γυναίκα αρέσει έναν άνδρα, που τρέφει τα ίδια αισθήματα γι' αυτήν, τότε είναι ευτυχισμένη.
- ♦ Η Αλεξάνδρα αρέσει όλους τους άνδρες, που τρέφουν τα ίδια αισθήματα γι' αυτήν.
- ♦ Η Εύη αρέσει οποιονδήποτε άνδρα τρέφει τα ίδια αισθήματα γι' αυτήν, με την προϋπόθεση, ότι είναι είτε πλούσιος και ευγενικός, είτε ευπαρουσίαστος και δυναμικός.

(β) Χρησιμοποιείτε την PROLOG για να βρείτε ποιός είναι χαρούμενος.

(γ) Προσθέστε έναν κανόνα στις φράσεις της ερώτησης (α) ώστε να γίνουν όλοι χαρούμενοι.

(δ) Βρείτε τις περιπτώσεις αμοιβαίας συμπάθειας (δύο άτομα που αρέσουν ο ένας τον άλλον).

(ε) Βρείτε τις περιπτώσεις που κάποιος αρέσει δύο ή και περισσότερα άτομα.

(στ) Βρείτε τις περιπτώσεις που δύο ή περισσότερα άτομα αρέσουν το ίδιο άτομο.

4]. Εστω το πρόγραμμα:

```
prerequisite(programming1, programming2).  
prerequisite(algorithms, data_structures).  
prerequisite(data_structures, data_bases1).  
prerequisite(data_bases1, databases2).  
prerequisite(programming2, systems_analysis).  
prerequisite(data_structures, operating_systems).
```

Ποιές είναι οι απαντήσεις που θα δοθούν στις παρακάτω ερωτήσεις:
(Ζητούνται όλες οι λύσεις)

?- prerequisite(X,Y), prerequisite(Z,Y).

?- prerequisite(X,Y), prerequisite(Y,Z).

5]. Να συγκριθεί η γλώσσα προγραμματισμού PROLOG με τις καθιερωμένες αλγοριθμικές γλώσσες προγραμματισμού και παραθέστε τις διαφορές τους. Η σύγκριση αυτή μπορεί να γίνει είτε σε γενικό επίπεδο, είτε με επιλογή κάποιας συγκεκριμένης γλώσσας (π.χ. PASCAL ή C).

6]. Σε τι διαφέρουν οι μεταβλητές της PROLOG από τις μεταβλητές μιας διατακτικής γλώσσας προγραμματισμού;

7]. Πώς ορίζεται η έννοια του **υποπρογράμματος (procedure)** στην PROLOG; Αναφέρετε βασικά σημεία στα οποία ένα υποπρόγραμμα στην PROLOG διαφέρει από ένα υποπρόγραμμα μιας καθιερωμένης γλώσσας προγραμματισμού. Συγκρίνετε ιδιαίτερα το μηχανισμό περάσματος παραμέτρων (parameter passing) της γλώσσας προγραμματισμού PROLOG με τους μηχανισμούς που χρησιμοποιούν οι καθιερωμένες αλγοριθμικές γλώσσες προγραμματισμού. Η σύγκριση μπορεί να γίνει είτε σε γενικό επίπεδο, είτε με επιλογή κάποιας συγκεκριμένης γλώσσας (π.χ. PASCAL ή C).

8]. Τα παρακάτω γεγονότα αναφέρονται στη χρονολογία γεννήσης και θανάτου μερικών φημισμένων συνθετών:

`composer(bach, 1685,1750).`
`composer(mozart, 1756,1791).`
`composer(beethoven, 1770,1827).`
`composer(verdi, 1813,1901).`

Να κατασκευαστεί το κατηγορήμα `contemporary(X,Y)`, το οποίο να αληθεύει όταν οι συνθέτες **X** και **Y** έχουν ζήσει κάποια κοινή χρονική περίοδο.

Μπορούν να χρησιμοποιηθούν τα ενσωματωμένα κατηγορήματα:

$X \neq Y$ (Το **X** δεν ενοποιείται με το **Y**)

$X \geq Y$ (Το **X** είναι μεγαλύτερο ή ίσο του **Y**).

9]. Δίνονται οι εξής πληροφορίες που σχετίζονται με τον τύπο, τον κυβισμό και την τελική ταχύτητα διαφόρων αυτοκινήτων:

"Αυτοκίνητο FIAT με 1000 cc και τελική ταχύτητα 145 Km/ h"

"Αυτοκίνητο VW με 1300 cc και τελική ταχύτητα 150 Km/ h"

"Αυτοκίνητο TOYOTA με 1300 cc και τελική ταχύτητα 165 Km/ h"

"Αυτοκίνητο LADA με 1500 cc και τελική ταχύτητα 160 Km/ h"

"Αυτοκίνητο BMW με 2000 cc και τελική ταχύτητα 195 Km/ h"

κ.λπ.

(α) Να γραφεί πρόγραμμα της Prolog για την αναπαράσταση της παραπάνω πληροφορίας.

(β) Να κατασκευαστούν οι παρακάτω ερωτήσεις:

- Ποιά είναι τα γνωστά αυτοκίνητα; (στο πρόγραμμα)

- Ποιά είναι τα αυτοκίνητα των 1300 cc;

- Ποιά αυτοκίνητα έχουν τελική ταχύτητα μεγαλύτερη των 160 Km/ h;

(γ) Να κατασκευαστεί το κατηγορήμα `category(X,Y)` το οποίο να αληθεύει όταν τα αυτοκίνητα με τύπο **X** και **Y** δεν διαφέρουν μεταξύ τους παραπάνω από 200 cc και οι τελικές τους ταχύτητες δεν έχουν διαφορά μεγαλύτερη των 10K m/ h.

10]. Δικαίωμα ψήφου σε μία εκλογική περιφέρεια έχει ένας πολίτης, εάν είναι μεγαλύτερος των 18 ετών, είναι κάτοικος στην περιφέρεια αυτή για τουλάχιστον δύο χρόνια και εάν είναι εγγεγραμμένος στους εκλογικούς καταλόγους. Να γραφεί πρόγραμμα που να περιγράφει τις παραπάνω πληροφορίες. Στη συνέχεια να βρεθούν οι εκλογείς της Β' περιφέρειας Θεσσαλονίκης.

4] Περιγράψτε με φράσεις της PROLOG την παρακάτω εικόνα:

