

## 6. ΕΝΣΩΜΑΤΩΜΕΝΑ ΚΑΤΗΓΟΡΗΜΑΤΑ

Η χρήση της PROLOG σε πραγματικές εφαρμογές έκανε αναγκαία την ενσωμάτωση σε αυτή χαρακτηριστικών που ξεφεύγουν από το μαθηματικό μοντέλο πάνω στο οποίο στηρίζεται η γλώσσα. Το υποσύνολο της PROLOG χωρίς αυτά τα χαρακτηριστικά αναφέρεται συνήθως σαν **Καθαρή (Pure) PROLOG**. Τα παραπάνω χαρακτηριστικά παρέχονται από τις διάφορες υλοποιήσεις της PROLOG με τη μορφή ενσωματωμένων κατηγορημάτων και μπορούν να χωριστούν στις εξής κατηγορίες:

- ✦ αριθμητικά,
- ✦ σύγκρισης όρων,
- ✦ εισόδου/εξόδου και πρόσβασης στο λειτουργικό σύστημα,
- ✦ μετα-λογικά
- ✦ ελέγχου της εκτέλεσης του προγράμματος,
- ✦ πρόσβασης/μετατροπής του χώρου εργασίας.
- ✦ χειρισμού αλυσίδων χαρακτήρων,

Μερικά από τα ενσωματωμένα κατηγορήματα (όπως είναι τα αριθμητικά και του χειρισμού αλυσίδων χαρακτήρων) μπορούν να αντικατασταθούν από προγράμματα σε καθαρή PROLOG. Υλοποιούνται όμως σαν ενσωματωμένα, για να γίνει πιο αποτελεσματική (από πλευράς ταχύτητας) η εκμετάλλευση του υλικού των υπολογιστών.

### 6.1. Αριθμητικά

Η δυνατότητα της εκτέλεσης αριθμητικών πράξεων δίνεται στην PROLOG μόνο μέσω ειδικών ενσωματωμένων κατηγορημάτων (***evaluable predicates***), που δέχονται σαν ορίσματα αριθμητικές εκφράσεις και τις υπολογίζουν. (Υπενθυμίζουμε ότι τα κατηγορήματα που ορίζονται από το

χρήστη δεν υπολογίζουν ποτέ τα ορίσματά τους.) Το βασικό ενσωματωμένο κατηγορήμα για τον υπολογισμό αριθμητικών εκφράσεων είναι το **is** και έχει την εξής γενική μορφή (ορίζεται σαν ενθεματικό κατηγορήμα):

**X is AE**

Η αριθμητική έκφραση **AE** υπολογίζεται και το αποτέλεσμα της ενοποιείται με το **X**. Το κατηγορήμα αποτυχαίνει, εάν η **AE** δεν είναι αριθμητική έκφραση. Το **X** δεν πρέπει να εμφανίζεται στην έκφραση **AE**. Επίσης στο χρόνο εκτέλεσης όλες οι μεταβλητές που εμφανίζονται στην **AE** πρέπει να είναι τοποθετημένες είτε με έναν αριθμό είτε με μία αριθμητική έκφραση. Η **AE** μπορεί να πάρει μία από τις παρακάτω μορφές (οι **Y**, **Z** είναι με τη σειρά τους αριθμητικές εκφράσεις):

<b>Y + Z</b>	πρόσθεση
<b>Y - Z</b>	αφαίρεση
<b>Y * Z</b>	πολλαπλασιασμός
<b>Y / Z</b>	διαίρεση
<b>Y // Z</b>	πηλίκιο διαίρεσης ακεραίων
<b>Y mod Z</b>	υπόλοιπο διαίρεσης ακεραίων
<b>- Y</b>	μοναδιαίος τελεστής αφαίρεσης
<b>Y ^ Z</b>	ύψωση σε δύναμη
<b>exp(Y)</b>	εκθετική συνάρτηση ( $e^Y$ )
<b>log(Y)</b>	φυσικός λογάριθμος
<b>log10(Y)</b>	δεκαδικός λογάριθμος
<b>sqrt(Y)</b>	τετραγωνική ρίζα
<b>sin(Y)</b>	ημίτονο
<b>cos(Y)</b>	συνημίτονο
<b>tan(Y)</b>	εφαπτομένη
<b>asin(Y)</b>	τόξο ημιτόνου
<b>acos(Y)</b>	τόξο συνημιτόνου
<b>atan(Y)</b>	τόξο εφαπτομένης

## 6.2. Σύγκρισης Όρων

**T1 < T2 :**

Το **T1** είναι μικρότερο του **T2**.

**T1 > T2 :**

Το **T1** είναι μεγαλύτερο του **T2**.

**T1 <= T2 :**

Το **T1** είναι μικρότερο ή ίσο του **T2**.

**T1 >= T2 :**

Το **T1** είναι μεγαλύτερο ή ίσο του **T2**.

**T1 = T2 :**

Πετυχαίνει, εάν τα **T1** και **T2** μπορούν να ενοποιηθούν.

**T1 \= T2 :**

Πετυχαίνει, εάν τα **T1** και **T2** δεν μπορούν να ενοποιηθούν.

**T1 == T2 :**

Πετυχαίνει, όταν στο χρόνο εκτέλεσης οι δύο όροι **T1** και **T2** είναι πανομοιότυποι. Ειδικά οι τυχόν μεταβλητές που εμφανίζονται στις αντίστοιχες θέσεις των **T1** και **T2** πρέπει, είτε να έχουν το ίδιο ακριβώς όνομα, είτε να είναι τοποθετημένες με την ίδιο όρο.

**T1 \== T2 :**

Πετυχαίνει, όταν στο χρόνο εκτέλεσης οι δύο όροι **T1** και **T2** δεν είναι πανομοιότυποι.

**T1 := T2 :**

Πετυχαίνει, όταν τα **T1** και **T2** έχουν την ίδια τιμή. Τα **T1** και **T2** μπορεί να είναι αριθμητικές εκφράσεις των οποίων οι τιμές υπολογίζονται πριν από την εκτέλεση του ενσωματωμένου κατηγορήματος  $\text{==}$ .

**T1**  $\neq$  **T2** :

Πετυχαίνει, όταν τα **T1** και **T2** δεν έχουν την ίδια τιμή. Τα **T1** και **T2** μπορεί να είναι αριθμητικές εκφράσεις των οποίων οι τιμές υπολογίζονται πριν από την εκτέλεση του ενσωματωμένου κατηγορήματος  $\neq$ .

### Παραδείγματα :

?-  $f(a,b) = f(a,X)$ .

yes

?-  $f(a,b) == f(a,X)$ .

no

?-  $f(a,X) == f(a,Y)$ .

no

?-  $X = Y, f(a,X) == f(a,Y)$ .

yes

?-  $1 + 3 = 2 + 2$ .

no

?-  $1 + 3 == 2 + 2$ .

no

?-  $1 + 3 ::= 2 + 2$ .

yes

## 6.3 Εισόδου/Εξόδου - Πρόσβασης στο Λειτουργικό Σύστημα

### 6.3.1 Εισόδου/Εξόδου και Χειρισμού Αρχείων

#### **consult(F) :**

Ζητά από το σύστημα της PROLOG να φορτώσει στο χώρο εργασίας της το πρόγραμμα (σύνολο φράσεων) που βρίσκεται στο αρχείο **F**. Όταν διαβάζεται κάποια φράση, αυτή τοποθετείται αμέσως μετά από τις φράσεις που έχουν ήδη φορτωθεί για το ίδιο κατηγορημα. Όταν στο αρχείο υπάρχει κάποια ερώτηση, αυτή εκτελείται τη στιγμή που διαβάζεται.

#### **reconsult(F) :**

Όπως και το ενσωματωμένο κατηγορημα **consult** με τη διαφορά ότι για κάθε κατηγορημα του αρχείου **F** διαγράφονται οι τυχόν φράσεις, που υπήρχαν στο χώρο εργασίας για το κατηγορημα αυτό, πριν από την εκτέλεση του **reconsult**.

#### **see(F) :**

Το αρχείο **F** γίνεται το τρέχον αρχείο εισόδου (*current input stream*). Αρχικά το τρέχον αρχείο εισόδου είναι αυτό που αντιστοιχεί στο πληκτρολόγιο.

#### **seeing(F) :**

Η μεταβλητή **F** ενοποιείται με το όνομα που αντιστοιχεί στο τρέχον αρχείο εισόδου.

#### **seen :**

Κλείνει το τρέχον αρχείο εισόδου. Τρέχων αρχείο εισόδου γίνεται αυτό που ήταν και πριν από την εκτέλεση του κατηγορήματος **see** που προηγήθηκε της εκτέλεσης του **seen**.

#### **tell(F) :**

Το αρχείο **F** γίνεται το τρέχον αρχείο εξόδου (*current output stream*). Αρχικά το τρέχον αρχείο εξόδου είναι αυτό που αντιστοιχεί στην οθόνη.

**telling(F) :**

Η μεταβλητή **F** ενοποιείται με το όνομα που αντιστοιχεί στο τρέχον αρχείο εξόδου.

**told :**

Κλείνει το τρέχον αρχείο εξόδου. Τρέχων αρχείο εξόδου γίνεται αυτό που ήταν και πριν από την εκτέλεση του κατηγορήματος **tell** που προηγήθηκε της εκτέλεσης του **told**.

**close(F) :**

Κλείνει το αρχείο **F** που ήταν ανοιχτό είτε για είσοδο είτε για έξοδο.

### **6.3.2 Εισόδου/Εξόδου Ορων**

**read(X) :**

Ο επόμενος όρος διαβάζεται από το τρέχον αρχείο εισόδου και ενοποιείται με το **X**. Ο όρος πρέπει οπωσδήποτε να ακολουθείται από μία τελεία, ακολουθούμενη από ένα κενό ή ένα "*carriage-return*". Εάν η κλήση του **read(X)** συμπέσει με το τέλος του αρχείου, το **X** ενοποιείται με την ειδική σταθερά **end\_of\_file**.

**write(X) :**

Ο όρος **X** γράφεται στο τρέχον αρχείο εξόδου.

### **6.3.3 Εισόδου/Εξόδου Χαρακτήρων**

**get0(X) :**

Διαβάζεται ο επόμενος χαρακτήρας από το τρέχον αρχείο εισόδου και ο ASCII κωδικός του ενοποιείται με το **X**. Εάν η κλήση του **get0(X)** συμπέσει με το τέλος του αρχείου, το **X** ενοποιείται με τον ASCII κωδικό του χαρακτήρα **control-Z** (ο χαρακτήρας αυτός πιθανά να διαφέρει από υλοποίηση σε υλοποίηση) και το τρέχον αρχείο εισόδου κλείνει.

**get(X) :**

Ιδιο όπως και το **get0(X)** με τη διαφορά ότι ο χαρακτήρας που διαβάζεται πρέπει να είναι εκτυπώσιμος και να μην είναι κενό.

**put(X) :**

Ο χαρακτήρας που αντιστοιχεί στον ASCII κωδικό **X**, τυπώνεται στο τρέχον αρχείο εξόδου. Το **X** μπορεί να είναι και μία ακέραια έκφραση.

**tab(X) :**

**X** κενά τυπώνονται στο τρέχον αρχείο εξόδου. Το **X** μπορεί να είναι και μία ακέραια έκφραση.

**nl :**

Μετά την εκτέλεση του η πρώτη θέση της επόμενης γραμμής στο τρέχον αρχείου εξόδου, γίνεται θέση εκτύπωσης.

Τα κατηγορήματα για είσοδο/έξοδο (όπως τα **read(X)**, **write(X)**) θεωρούνται εξω-λογικά γιατί παραβιάζουν τη βασική διαδικασία περάσματος τιμών της PROLOG που είναι η ενοποίηση.

## **6.4. Μετα-Λογικά**

**var(X) :**

Ελέγχει εάν τη στιγμή της εκτέλεσης η **X** είναι τοποθετημένη με μία μεταβλητή.

**nonvar(X) :**

Ελέγχει εάν τη στιγμή της εκτέλεσης το **X** είναι τοποθετημένο με έναν όρο που δεν είναι μεταβλητή.

**atom(X) :**

Ελέγχει εάν τη στιγμή της εκτέλεσης το **X** είναι τοποθετημένο με ένα **άτομο** (μιά σταθερά δηλαδή που δεν είναι αριθμός).

**number(X) :**

Ελέγχει εάν τη στιγμή της εκτέλεσης το **X** είναι τοποθετημένο με έναν αριθμό.

**atomic(X) :**

Ελέγχει εάν τη στιγμή της εκτέλεσης το **X** είναι τοποθετημένο με μία σταθερά ή έναν αριθμό.

**functor(Term,F,N) :**

Το συναρτησιακό σύμβολο του όρου **Term** έχει όνομα **F** και πληθικό αριθμό **N**. Στο χρόνο εκτέλεσης πρέπει τουλάχιστον το **Term** ή τα **F** και **N** να είναι τοποθετημένα με μη-μεταβλητές.

**arg(I,Term,X) :**

Το υπ' αριθμόν **I** όρισμα του όρου **Term** ενοποιείται με το **X**. Πρέπει οπωσδήποτε στο χρόνο εκτέλεσης το **I** να είναι τοποθετημένο με ένα ακέραιο θετικό αριθμό.

**Term =.. List<sup>1</sup>(\*) :**

Ενοποιεί το όρισμα **Term** με τον όρο του οποίου η αναπαράσταση με μορφή λίστας είναι **List**. (Εάν  $f(t_1, t_2, \dots, t_n)$  είναι ένας όρος, με πληθικό αριθμό **n**, η αναπαράσταση του με μορφή λίστας είναι η  $[f, t_1, t_2, \dots, t_n]$  με **n+1** στοιχεία.) Το κατηγορήμα "**=..**" δουλεύει και προς τις δύο κατευθύνσεις. Εάν το **Term** είναι τοποθετημένο με κάποιον όρο, τότε το **List** ενοποιείται με την αντίστοιχη αναπαράσταση της λίστας **List**, ενώ στην αντίθετη περίπτωση το **Term** ενοποιείται με τον όρο που έχει σαν συναρτησιακό σύμβολο το πρώτο στοιχείο της λίστας **List** και ορίσματα τα υπόλοιπα στοιχεία της.

**call(X) ή X:**

---

<sup>1</sup> Προφέρεται γιουνίβ (univ)



Εάν τη στιγμή της εκτέλεσης το **X** είναι τοποθετημένο με μία σύζευξη στόχων, που μπορεί να είναι αποδεκτή σαν το σώμα μιας φράσης, τότε ο στόχος **call(X)** εκτελείται, όπως ακριβώς θα εκτελούνταν η σύζευξη αυτή, εάν υπήρχε στη θέση του **call(X)**. Το ενσωματωμένο αυτό κατηγορημα ονομάζεται **μηχανισμός της μετα-μεταβλητής (meta-variable facility)**.

**name(A,L) :**

Εάν το **X** είναι τοποθετημένο με ένα άτομο ή αριθμό τότε το **L** είναι μία λίστα που έχει σαν στοιχεία τους ASCII κωδικούς των χαρακτήρων που αποτελούν το **X**. Εάν το **X** είναι τοποθετημένο με μία μεταβλητή τότε το **L** πρέπει να είναι υποχρεωτικά τοποθετημένο με μία λίστα ASCII κωδικών.

**Παραδείγματα :**

?- **arg(2,g(a,b,c),X).**

**X = b**

?- **functor(g(a,b,c),F,N).**

**F = g**

**N = 3**

?- **functor(T,g,3).**

**T = g(X,Y,Z)**

?- **f(a,b) =.. L.**

**L= [f,a,b]**

?- **T =.. [point,1,2,3].**

**T = point(1,2,3)**

?- **T =.. [f,1,g(1,2),3].**

**T = f(1,g(1,2),3)**

**?- name(prolog,L).**

**L = [112, 114, 111, 108, 111, 103]**

**?- name(X, [112, 114, 111, 108, 111, 103]).**

**X = prolog**

ή ισοδύναμα (βλέπε §3):

**?- name(X, "prolog").**

**X = prolog**

## 6.5. Σύνολο Λύσεων ενός Στόχου

Πολλές φορές είναι αναγκαία, σε ένα προγράμμα, η αναφορά στο σύνολο των λύσεων που μπορεί να έχει ένας στόχος. Οι περισσότερες υλοποιήσεις της PROLOG διαθέτουν ενσωματωμένα κατηγορήματα όπως είναι το **bagof** και το **setof**, που επιτρέπουν τον υπολογισμό αυτού του συνόλου.

**bagof(X,P,S) :**

Ενοποιεί το **S** με τη λίστα όλων των τιμών του **X**, για τις οποίες το κατηγορήμα **P** αληθεύει.

**setof(X,P,S) :**

Όπως το **bagof(X,P,S)**, με τη διαφορά ότι η λίστα **S** που δημιουργείται είναι διατεταγμένη και χωρίς τυχόν επαναλαμβανόμενες τιμές του **X**.

## 6.6. Ελέγχου της Εκτέλεσης του Προγράμματος

**true :**

Πετυχαίνει πάντα.

**false :**

Αποτυχαίνει, πάντα.

**!** :

Το κατηγορημα της αποκοπής (*cut*). Ακυρώνει όλα τα σημεία οπισθοδρόμησης που δημιουργήθηκαν από τη στιγμή της εκτέλεσης του στόχου που ενεργοποίησε το "!" (βλέπε §7).

**not(P)** :

Αποτυχαίνει, όταν υπάρχει λύση για το κατηγορημα **P**. Στην αντίθετη περίπτωση πετυχαίνει.

**P ; Q** :

Διάζευξη των στόχων **P** και **Q**. Εάν ο στόχος **P** πετύχει, ο **Q** αγνοείται. Αλλιώς εκτελείται ο **Q**.

**P -> Q ; R** :

"if **P** then **Q** else **R**". Εκτελείται ο στόχος **P**. Εάν πετύχει, εκτελείται ο **Q**. Αλλιώς εκτελείται ο **R**. Συμπεριφέρεται σαν να έχει οριστεί από τις φράσεις:

**P -> Q ; R :- P, !, Q.**

**P -> Q ; R :- R.**

**P -> Q** :

"if **P** then **Q**". Εκτελείται ο στόχος **P**. Εάν πετύχει, εκτελείται ο **Q**. Είναι ισοδύναμο με το:

**P -> Q ; fail.**

**repeat** :

Πετυχαίνει πάντοτε και δημιουργεί ένα μή πεπερασμένο αριθμό από σημεία οπισθοδρόμησης. Συμπεριφέρεται σαν να έχει οριστεί από τις φράσεις:

**repeat.**

**repeat :- repeat.**

## 6.7. Πρόσβασης - Μετατροπής του Προγράμματος

Πολλές εφαρμογές απαιτούν τη δυνατότητα πρόσβασης στις φράσεις του προγράμματος κατά τη διάρκεια της εκτέλεσης του. Το ενσωματωμένο κατηγορήμα **clause(Head,Body)** δίνει αυτήν τη δυνατότητα, ενώ τα κατηγορήματα **assert(Clause)** και **retract(Clause)** δίνουν τη δυνατότητα της πρόσθεσης και αφαίρεσης φράσεων από το χώρο εργασίας στη διάρκεια της εκτέλεσης ενός προγράμματος.

**assert(Clause) :**

Το στιγμιότυπο του **Clause**, τη στιγμή της εκτέλεσης, ερμηνεύεται σαν μία φράση η οποία προστίθεται στο χώρο εργασίας. Η θέση της νέας φράσης εξαρτάται από τη συγκεκριμένη υλοποίηση της PROLOG (συνήθως προστίθεται στο τέλος). Εάν το **Clause** είναι τοποθετημένο με μία μεταβλητή έχουμε κατάσταση λάθους.

**asserta(Clause) :**

Όπως το **assert(Clause)**, με τη διαφορά ότι η νέα φράση **Clause**, προστίθεται πρώτη σε σχέση με τις φράσεις που ορίζουν το ίδιο κατηγορήμα.

**assertz(Clause) :**

Όπως το **assert(Clause)**, με τη διαφορά ότι η νέα φράση **Clause**, προστίθεται τελευταία σε σχέση με τις φράσεις που ορίζουν το ίδιο κατηγορήμα.

**retract(Clause) :**

Η πρώτη φράση που μπορεί να ενοποιηθεί με το **Clause** διαγράφεται από το χώρο εργασίας. Σε περίπτωση οπισθοδρόμησης ο στόχος **retract(Clause)** διαγράφει την επόμενη φράση που η κεφαλή της μπορεί επίσης να ενοποιηθεί με το **Clause**.

**clause(Head,Body) :**

Αναζητείται η πρώτη φράση του χώρου εργασίας η κεφαλή της οποίας μπορεί να ενοποιηθεί με το **Head**. Σαν αποτέλεσμα το σώμα της φράσης αυτής ενοποιείται με το **Body**. Σε περίπτωση οπισθοδρόμησης ο στόχος

**clause(Head,Body)** πετυχαίνει για την επόμενη φράση, της οποίας η κεφαλή μπορεί να ενοποιηθεί με το **Head**. Εάν στο χρόνο εκτέλεσης το **Head** είναι τοποθετημένο με μία μεταβλητή έχουμε κατάσταση λάθους.

**listing :**

Μία λίστα, όλων των φράσεων που υπάρχουν στο χώρο εργασίας, εκτυπώνεται στο τρέχον αρχείο εξόδου.

**listing(P) :**

Μία λίστα, των φράσεων που υπάρχουν στο χώρο εργασίας και ορίζουν το κατηγορημα **P**, εκτυπώνεται στο τρέχον αρχείο εξόδου. Το ορισμα του listing μπορεί επίσης να είναι της μορφής **P/N**, οπότε εκτυπώνονται μόνον οι φράσεις που αντιστοιχούν στο κατηγορημα **P** με πληθικό αριθμό **N**.

**abolish(P/N) :**

Διαγράφει όλες τις φράσεις από το χώρο εργασίας που αντιστοιχούν στο κατηγορημα **P** με πληθικό αριθμό **N**.