

9. Μετα-Λογικός προγραμματισμός

Τα "**μετα-προγράμματα**" χειρίζονται άλλα προγράμματα σα δεδομένα και έχουν τη δυνατότητα να τα αναλύσουν, να τα μετατρέψουν και να τα προσομοιώσουν. Μια γλώσσα που μπορεί να χειριστεί τα προγράμματα κάποιας άλλης ονομάζεται **μεταγλώσσα**. Είναι πολύ σημαντικό μια γλώσσα προγραμματισμού να έχει την ιδιότητα της μεταγλώσσας, γιατί έτσι μπορεί να επαυξήσει εύκολα τις δυνατότητες της με καινούργιους γλωσσικούς μηχανισμούς ή και με χαρακτηριστικά άλλων γλωσσών.

Ο μετα-προγραμματισμός είναι ιδιαίτερα εύκολος στη PROLOG διότι ο τρόπος αναπαράστασης δεδομένων και προγράμματος είναι ομοιόμορφος και διότι η PROLOG είναι ήδη επαυξημένη με μεταλογικά χαρακτηριστικά (κυρίως με τη μορφή ενσωματωμένων κατηγορημάτων). Ένα κλασσικό παράδειγμα μετα-προγραμματισμού είναι το πρόγραμμα που υλοποιεί ένα διερμηνέα της καθαρής PROLOG με τρεις μόνο φράσεις.

```
E1:  execute(true).  
E2:  execute((Goal,Rest_Goals)) :-  execute(Goal),  
                                     execute(Rest_Goals).  
E3:  execute(Goal) :-  clause(Goal,Body),  
                       execute(Body).
```

Ο μετα-διερμηνέας λειτουργεί ως εξής: Ο στόχος **true** (κενή φράση) πετυχαίνει με τη χρήση της φράσης (E1). Μία σύζευξη στόχων της μορφής (**Goal,Rest_Goals**) επιλύεται με τη χρήση της (E2), όπου ο διερμηνέας καλεί αναδρομικά τον εαυτό του πάνω στον στόχο **Goal** και στη σύζευξη των στόχων **Rest_Goals**. Για την επίλυση ενός απλού στόχου ο διερμηνέας κάνει κλήση της φράσης "**Goal :- Body**" και επιλύει αναδρομικά το σώμα της. Οι διαδικασίες της κλήσης και της ενοποίησης επιτυγχάνονται με τη βοήθεια του ενσωματωμένου κατηγορήματος "**clause**". Η αποτυχία του στόχου προκαλεί οπισθοδρόμηση και αναγκάζει την **clause(Goal,Body)** να αναζητήσει άλλη πιθανή λύση.