

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

6^ο Εξάμηνο

- Αρχεία - Είσοδος/Εξοδος
- Προγράμματα Prolog που επεξεργάζονται προγράμματα Prolog

Δημοσθένης Σταμάτης

<http://www.iee.ihu.gr/~demos>

Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων

Είσοδος από αρχείο/ρεύμα εισόδου

see(File)

Συνδέει στο χώρο εργασίας της Prolog ως **ρεύμα εισόδου** το αρχείο **File**. Από το σημείο της σύνδεσης και μετά κάθε ενέργεια εισόδου απευθύνεται σε αυτό.

Αν **File='user'** επιλέγεται το πληκτρολόγιο ως ρεύμα εισόδου.

seeing(File)

Καταχωρεί στη μεταβλητή **File** το όνομα του αρχείου που κατά την τρέχουσα χρονική στιγμή είναι το ρεύμα εισόδου.

seen

Κλίνει το τρέχον ρεύμα εισόδου και προκαλεί επιστροφή στο πληκτρολόγιο (user_input)

Είσοδος από αρχείο/ρεύμα εισόδου

read(Term)

Διαβάζει τον επόμενο όρο από το τρέχον **ρεύμα εισόδου** και τον τοποθετεί στη μεταβλητή **Term**.

Το τέλος του διαβάσματος όρου πρέπει να σηματοδοτείται από "."

Ένα Παράδειγμα:

```

browse(File) :-
    seeing(Old),      % αποθήκευσε για αργότερα
    see(File),        % άνοιξε αυτό το αρχείο
    read(Data),       % διάβασε από το αρχείο
    process(Data),    % κάνε κάτι με αυτό που διάβασες
    seen,             % κλείσε το File
    see(Old).         % επέστρεψε στο Old
  
```

Έξοδος σε αρχείο/ρεύμα εισόδου

tell(File)

Συνδέει στο χώρο εργασίας της Prolog ως **ρεύμα εξόδου** το αρχείο **File**. Από το σημείο της σύνδεσης και μετά κάθε ενέργεια εξόδου απευθύνεται σε αυτό.

Αν **File='user'** επιλέγεται η οθόνη ως ρεύμα εξόδου.

telling(File)

Καταχωρεί στη μεταβλητή **File** το όνομα του αρχείου που κατά την τρέχουσα χρονική στιγμή είναι το ρεύμα εξόδου.

told

Κλίνει το τρέχον ρεύμα εξόδου και προκαλεί επιστροφή στο ρεύμα εξόδου της οθόνης (user_output)

Είσοδος από αρχείο/ρεύμα εισόδου

write(Term)

Γράφει (αποθηκεύει) τον όρο **Term** στο τρέχον **ρεύμα εξόδου**.

nl

Καταχωρεί αλλαγή γραμμής στο τρέχον ρεύμα εξόδου

Ένα Παράδειγμα:

```
Listing_to_file(File) :-
    telling(Old),      % αποθήκευσε για αργότερα
    tell(File),       % άνοιξε αυτό το αρχείο
    listing,          % γράψε τις φράσεις προγράμματος
    told,             % κλείσε το File
    tell(Old).        % επέστρεψε στο Old
```

Ενσωματωμένα κατηγορήματα πρόσβασης και μετατροπής του χώρου εργασίας της Prolog

Η PROLOG επιτρέπει την μετατροπή του κώδικά της κατά τη διάρκεια της εκτέλεσης του προγράμματος !!! με τη βοήθεια ενσωματωμένων κατηγορημάτων:



clause(Head, Body):

πετυχαίνει όταν υπάρχει φράση του προγράμματος στη βάση (χώρο εργασίας) της Prolog, της οποίας η κεφαλή (συμπέρασμα) να μπορεί να εννοποιηθεί με το **Head** και το σώμα της (προϋποθέσεις) να μπορεί να εννοποιηθεί με το **Body**. Η παράμετρος **Head** πρέπει να είναι οπωσδήποτε τοποθετημένη κατά τη διάρκεια εκτέλεσης.

Ενσωματωμένα κατηγορήματα πρόσβασης και μετατροπής του χώρου εργασίας της Prolog



assert(Clause):

Το στιγμιότυπο της **Clause**, τη στιγμή της εκτέλεσης, ερμηνεύεται σαν μία φράση η οποία προστίθεται στη βάση (χώρο εργασίας) της Prolog. Η παράμετρος **Clause** πρέπει να είναι οποσδήποτε τοποθετημένη κατά τη διάρκεια εκτέλεσης.



retract(Clause):

Η πρώτη φράση που μπορεί να ενοποιηθεί με την **Clause** διαγράφεται από το χώρο εργασίας. Σε περίπτωση οπισθοδρόμησης διαγράφεται η επόμενη φράση που ενοποιείται με την **Clause**.

Ενσωματωμένο κατηγορήμα “=..” (univ)

Term =..List

Επιστρέφει μία λίστα **List** η οποία έχει ως κεφαλή το συναρτησιακό σύμβολο του σύνθετου όρου **Term** και τα υπόλοιπα στοιχεία της είναι τα ορίσματα του σύνθετου όρου.

Λειτουργεί και αντίστροφα μετατρέποντας μία λίστα σε σύνθετο όρο.

Παραδείγματα εκτέλεσης:



? - parent(nikos, maria) =.. L.
L = [parent, nikos, maria]



?- T =.. [parent, nikos, maria].
T = parent(nikos,maria)

Ενσωματωμένο κατηγορημα functor

functor(Term, Name, Arity)

Επιστρέφει true όταν **Term** είναι όρος με συναρτησιακό όνομα **Name** και αριθμό ορισμάτων **Arity**. Αν **Term** είναι μεταβλητή αυτή ενοποιείται με ένα νέο όρο που τα ορίσματά του είναι όλα διαφορετικές μεταβλητές (skeleton). Εάν ο όρος **Term** είναι σταθερά το **Arity** τοποθετείται με **0**. Σε οποιαδήποτε άλλη περίπτωση προκαλείται σφάλμα (error).

Παραδείγματα εκτέλεσης:



```
?- functor(onomateponimo(kostas,nikolaou), N, A).
```

```
N = onomateponimo, A = 2.
```



```
?- functor(T, onomateponimo, 2).
```

```
T = onomateponimo(_G3107, _G3108).
```

Φόρτωση Προγράμματος Prolog από αρχείο



```
my_consult(File):-
    see(File),
    read(Clause),
    put_in_db(Clause).

put_in_db(end_of_file):-!,           % ! Αποκοπή (cut)
    seen.

put_in_db(Clause):-
    assert(Clause),
    read(New_Clause),
    put_in_db(New_Clause).
```

Δημιουργία λίστας με τα κατηγορήματα που βρίσκονται στο χώρο εργασίας της Prolog



```

make_list(PredicatesList):-
    read(Clause),
    process(Clause,[ ],PredicatesList).

process(end_of_file,PredicatesList,PredicatesList):- !,
    seen.

process(Clause,Sofar,PredicatesList):-
    getHead(Clause,Head),
    functor(Head,Predicate,N),
    read(New_Clause),
    process(New_Clause,[Predicate/N|Sofar],PredicatesList).

getHead((Head:-Body),Head):- !.
getHead(Head,Head).

```

Διαγραφή κατηγορημάτων που βρίσκονται στο χώρο εργασίας της Prolog



```

free_db([ ]).
free_db([Predicate/N|Rest]):-
    functor(Head,Predicate,N),
    retract_all(Head),
    free_db(Rest).

retract_all(Head):-
    clause(Head,Body),
    retract((Head:-Body)),
    fail.           % Οπισθοδρόμηση με το «ζόρι»
retract_all(Head).

```

Επιανα-φόρτωση Προγράμματος Prolog από αρχείο



```
my_reconsult(File):-  
    see(File),  
    make_list(PredicatesList),  
    free_db(PredicatesList),  
    seen,  
    my_consult(File).
```