



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΩΤΗ ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ

ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΠΕΙΡΑΙΑΣ 2020

Η Βάση Γνώσης της Εργασίας

```
1 % The family that we are asked to implement from the tree diagram.
2 family(
3     person(tom, fox, date(7, may, 1950), works(bbc, 15200)),
4     person(ann, fox, date(9, may, 1951), unemployed),
5     [person(pat, fox, date(5, may, 1973), unemployed),
6     person(jim, fox, date(5, may, 1973), unemployed)]
7 ).
8
9 % 2 more examples of families.
10 family(
11     person(tommy, duke, date(13, july, 1962), works(bmw, 17900)),
12     person(jane, duke, date(22, november, 1961), works(kotsovolos, 9750)),
13     [person(jacob, duke, date(18, november, 1990), student),
14     person(jonathan, duke, date(11, december, 1993), student)]
15 ).
16
17 family(
18     person(oliver, miller, date(23, july, 1961), unemployed),
19     person(karen, miller, date(2, november, 1959), works(logitech, 17500)),
20     [person(juliana, miller, date(18, octomber, 1999), student),
21     person(helen, miller, date(7, august, 1990), works(ote, 19800)),
22     person(johan, miller, date(10, july, 1999), unemployed),
23     person(david, miller, date(6, june, 1999), unemployed)]
24 ).
25
26 % wife/1: Retrieve all wives that are working/ employed.
27 wife(X) :-
28     family(_, person(X, _, _, works(_, _)), _).
29
30 % exists/1: Retrieve all unemployed parents who were born before 1963.
31 % Each for every mother/father
32 exists(X) :-
33     family(_, person(X, _, date(_, _, Y), unemployed), _),
34     Y < 1963.
35
36 exists(X) :-
37     family(person(X, _, date(_, _, Y), unemployed), _, _),
38     Y < 1963.
```

Γενικά

Όπως ορίζει και η εκφώνηση της εργασίας, δηλώνουμε το κατηγορημα family/3 με αντικείμενα τύπου person. Κάθε άτομο (atom) μέσα στη δήλωση κατέχει διαφορετικό ρόλο. Πιο συγκεκριμένα το πρώτο «person» αναφέρεται στο **πατέρα**, το δεύτερο στη **μητέρα** και το τρίτο, το οποίο είναι λίστα, στα **παιδιά**. Στο παράδειγμα που μας δίδεται η οικογένεια αποτελείται από 2 μόνο παιδιά ενώ στη συνέχεια προσθέτουμε δύο ακόμη οικογένειες για να εμπλουτίσουμε τη βάση γνώσης μας.

Για μεγαλύτερη διαφοροποίηση των νέων οικογενειών που προσθέτουμε στο παράδειγμα μας σε σχέση με το παράδειγμα του διαγράμματος προσθέτουμε:

- Διαφορετικές ημερομηνίες γέννησης στο κατηγορημα date/3.
- Αντικείμενο student σε μερικά από τα παιδιά, σε κάποια unemployed και σε κάποια άλλα το κατηγορημα works/2.
- Μεγαλύτερο αριθμό παιδιών, πιο συγκεκριμένα 2+ (βλ. γραμμές 20-23).

Τα ζητούμενα της εργασίας είναι δύο:

1. Να ορισθεί κατηγορημα wife/1 και να χρησιμοποιηθεί για να ανακτήσουμε **όλες τις συζύγους που εργάζονται**.
2. Να ορισθεί κατηγορημα exists/1 και να χρησιμοποιηθεί για να ανακτήσουμε **όλους τους άνεργους γονείς που γεννήθηκαν πριν το 1963**.

Υλοποίηση & Παραδείγματα Ορθής Εκτέλεσης

1. Για την υλοποίηση του πρώτου ζητούμενου (βλ. γραμμές 27-28) παρατηρούμε ότι στη δομή του δημοτολογίου, η κάθε μητέρα βρίσκεται στην δεύτερη θέση του κατηγορήματος family/3. Με τη βοήθεια **μεταβλητής X** στο κατηγορημα wife/1 δημιουργούμε **κανόνα** με τα εξής χαρακτηριστικά:

- Για κάθε οικογένεια, θέλουμε τη τιμή του δεύτερου ατόμου (atom), του κατηγορήματος family/3 (μητέρα).
- Για κάθε μητέρα, θέλουμε τη τιμή του πρώτου ατόμου (atom), του κατηγορήματος person/4 (όνομα μητέρας)
- Για κάθε μητέρα, εξασφαλίζουμε ότι εργάζεται με τη χρήση του κατηγορήματος works/2 και δύο ανώνυμες μεταβλητές για τα άτομα (atoms), τα οποία δε μας αφορούν.

Παρακάτω θα δούμε μερικά παραδείγματα της χρήσης του ανωτέρω κανόνα:

```
% c:/Users/bmano/Desktop/Prolog files/City-Hall-Planner/city
_hall.pl compiled 0.02 sec, 0 clauses
?- wife(X).
X = jane ;
X = karen.
```

Πράγματι, η **jane** και η **karen** είναι οι μοναδικές μητέρες που εργάζονται στη βάση γνώσης μας. Μπορούμε επίσης, να εργαστούμε αντίστροφα: κάνοντας την ερώτηση συγκεκριμένα για την ann:

```
?- wife(ann).
false.
```

Προφανώς η απάντηση είναι **False** καθώς η ann είναι άνεργη (unemployed). Τέλος, μπορεί να γίνει και με τη μορφή διαδοχικών ερωτήσεων προς τα διάφορα ονόματα:

```
?- wife(karen), wife(jane).
true.
```

2. Για την υλοποίηση του δεύτερου ζητούμενου (βλ. γραμμές 32-38) παρατηρούμε ότι σύμφωνα με τη δομή του δημοτολογίου θα πρέπει να έχουμε πρόσβαση σε 3 άτομα (atoms) του κατηγορήματος family/3: **μητέρα**, **πατέρα**, **έτος γέννησης** και **κατάσταση εργασίας**. Ορίζουμε λοιπόν, το κατηγορήμα exists/1 με **μεταβλητές X (όνομα), Y (έτος γέννησης)** έτσι ώστε να αντλήσουμε τα εξής χαρακτηριστικά:

- Για κάθε οικογένεια, θέλουμε τη τιμή του πρώτου και δεύτερου ατόμου (atom) του κατηγορήματος family/3 (όνομα μητέρα ή πατέρα - X).
- Για κάθε γονέα, θέλουμε το τρίτο άτομο (atom) του κατηγορήματος date/3 (έτος γέννησης - Y). Αυτό θα μεταφέρεται στην μεταβλητή Y, για να γίνει

στη συνέχεια η αριθμητική σύγκριση με το έτος 1963 (βλ. γραμμές 34, 38).

- Τέλος, για κάθε γονέα, θέλουμε το τρίτο άτομο (atom) του κατηγορήματος person/3 να έχει τιμή **unemployed**.

Παρακάτω θα δούμε μερικά παραδείγματα της χρήσης του κανόνα exists/1:

```
?- exists(X).  
X = ann ;  
X = oliver.
```

Πράγματι, αν δούμε τη βάση γνώσης μας, οι μοναδικοί γονείς που γεννήθηκαν πριν το 1963 και είναι άνεργοι είναι η **ann** και ο **oliver**. Παρόμοια, μπορούμε να εκτελέσουμε την ερώτηση και για μεμονωμένα πρόσωπα της βάσης γνώσεών μας:

```
?- exists(tom).  
false.
```

```
?- exists(jane).  
false.
```

```
?- exists(ann).  
true
```

```
?- exists(tommy).  
false.
```

```
?- exists(jane).  
false.
```

```
?- exists(oliver).  
true.
```

Τόσο ο κώδικας Prolog της εργασίας όσο και στιγμιότυπα επιτυχής εκτέλεσης του προγράμματος θα είναι διαθέσιμα στο GitHub μέσω του παρακάτω συνδέσμου:

<https://github.com/vagman/City-Hall-Planner>