

Παράδειγμα count με χρήση findall

```
count(Country,T) :-
```

```
    findall(N,next(Country,N),Neighbours),
    length(Neighbours,T).
```

```
?- count('Greece',X).
```

```
X = 4
```

```
? - count(Country,X).
```

```
X = 30
```

Country = μεταβλητή χωρίς τιμή

56

Παράδειγμα count με χρήση bagof

```
count(Country,T) :-
```

```
    bagof(N,next(Country,N),Neighbours),
    length(Neighbours,T).
```

```
?- count('Greece',X).
```

```
X = 4
```

(ίδια απάντηση με findall)

```
?- count(Country,X).
```

```
Country = 'Greece'
```

```
X = 4 ;
```

```
Country = 'Turkey'
```

```
X = 5 ;
```

Εξαιτίας του **bagof** η μεταβλητή **Country** πρώτα παίρνει κάποια συγκεκριμένη τιμή και μετά αναζητούνται όλα τα **N** που συνορεύουν με αυτήν, ενώ στη **findall**, η **Country** είναι 'αδιάφορη' μεταβλητή, χωρίς τιμή.

57

Χειρισμός συνόλου λύσεων

- Να ορισθεί το κατηγορήμα **employees(Dept,Employees)** το οποίο να επιστρέφει όλους τους υπαλλήλους (λίστα **Employees**) που εργάζονται στο τμήμα **Dept**.

```
employees(Dept,Employees) :-
```

```
    findall(
        Emp,
        employee(_,data(Emp,_,_,department(Dept))),
        Employees
    ).
```

*Η λίστα **Employees** θα περιέχει τις τιμές που παίρνει η μεταβλητή **Emp***

*Η λίστα στην οποία "μαζεύονται" οι τιμές της μεταβλητής **Emp***

*Κλήση η οποία "παράγει" τις τιμές για τη μεταβλητή **Emp***

58

Χειρισμός συνόλου λύσεων

- Να ορισθεί κατηγορήμα **no_of_employees(Dept,N)** το οποίο να επιστρέφει το συνολικό αριθμό υπαλλήλων που εργάζονται στο τμήμα **Dept**.

```
no_of_employees(Dept,N) :-
```

```
    employees(Dept,Employees),
    length(Employees,N).
```

*Λίστα όλων των υπαλλήλων που δουλεύουν στο τμήμα **Dept***

Εύρεση μήκους της λίστας - δηλαδή του αριθμού των υπαλλήλων

59

Χειρισμός συνόλου λύσεων (setof)

- Να ορισθεί το κατηγορήμα `print_employees(Dept)`, το οποίο να τυπώνει τους υπαλλήλους του τμήματος `Dept` στην οθόνη, ταξινομημένους κατά επίθετο, όνομα, πατρώνυμο (έναν υπάλληλο ανά γραμμή).

```
print_employees(Dept) :-
```

```
setof(
```

```
    Last-First-Middle,
```

```
    Code^ID^Ad^HD^
```

```
    employee(Code,
```

```
        data(name(first(First), last(Last), middle(Middle)) ,
```

```
        ID, Ad, HD,
```

```
        department(Dept)) ,
```

```
    Employees
```

```
),
```

```
print_employees_aux(Employees).
```

Η λίστα *Employees* θα περιέχει τους **σύνθετους όρους** που δημιουργούνται από το συνδυασμό των τιμών που παίρνουν οι 3 μεταβλητές

Δεν μας ενδιαφέρουν οι τιμές των 4 αυτών μεταβλητών, συνεπώς **πρέπει** να δηλώσουμε ότι είναι "ελεύθερες"

Κλήση η οποία "παράγει" τις τιμές. Γίνεται ανάλυση του ονόματος στα στοιχεία του, γιατί πρέπει να γίνει **αλλαγή της σειράς** του **Last** με το **First**

Η λίστα στην οποία "μαζεύονται" οι τιμές

Βοηθητικό κατηγορήμα για την εκτύπωση των στοιχείων της λίστας

Χειρισμός συνόλου λύσεων (setof)

- Βοηθητικό κατηγορήμα για την εκτύπωση των στοιχείων της λίστας.

```
print_employees_aux([]).
```

```
print_employees_aux([L-F-M|Tail]) :-
```

```
    write(F), write(' '),
```

```
    write(M), write(' '),
```

```
    write(L), nl,
```

```
    print_employees_aux(Tail).
```

Κενό μεταξύ των λέξεων

Αλλαγή σειράς - Απαραίτητη!

Εκτύπωση των υπολοίπων στοιχείων της λίστας