Σύντομη Εισαγωγή στην Prolog

Φυλλάδιο σημειώσεων για το 3ο εργαστήριο του μαθήματος

1 Χρήσιμες πληροφορίες και συνηθισμένα λάθη

Ορθογραφικά λάθη: Είναι σύνηθες να πληκτρολογούμε λανθασμένα είτε το όνομα μίας μεταβλητής:

```
member(Element,[Elemnet|_]).
member(Element,[_|Tail]) :-
  member(Element,Tail).
```

είτε το όνομα ενός κατηγορήματος:

```
?- foo(A).
ERROR: toplevel: Undefined procedure: foo/1 (DWIM could not correct
     goal)
```

Στην πρώτη περίπτωση ο διερμηνέας μας προειδοποιεί ότι υπάρχει μία singleton μεταβλητή ενώ στην δεύτερη μας λέει ότι το predicate_name/arity δεν έχει οριστεί. Αυτά βέβαια συμβαίνουν στην περίπτωση που δεν είμαστε τόσο άτυχοι ώστε η ανορθόγραφη λέξη μας να ενοποιείται με κάτι διαφορετικό από το επιθυμητό.

Συνηθισμένο είναι επίσης να γράφουμε μία μεταβλητή με μικρό το πρώτο γράμμα και, σε αυτή την περίπτωση, συνήθως αποτυγχάνει το κατηγόρημά μας.

```
?- fib(7, x).
false.
```

ή (ακόμα χειρότερα) μέσα σε ένα πρόγραμμα:

```
ancestor(x, x).
ancestor(x, y) :- parent(x, z), ancestor(z, y).
```

Διαφορές στον αριθμό ορισμάτων: Αυτό το λάθος συνήθως συμβαίνει όταν αποφασίσουμε να προσθέσουμε μία ακόμη μεταβλητή στον ορισμό ενός κατηγορήματος και δεν θυμηθούμε να αλλάξουμε όλες τις κλήσεις στο κατηγόρημα αυτό. Και σε αυτή την περίπτωση ο διερμηνέας μας λέει ότι το predicate_name/arity δεν έχει οριστεί.

Τελείες και κόμματα: Ένα από τα πιο συνηθισμένα λάθη στην Prolog φαίνεται στο επόμενο πρόγραμμα, στόχος του οποίου είναι να ελέγξει την ισότητα λιστών:

```
eq([], []).
eq([X|L], M) :- del(X, M, N), eq(L, N),
del(X, [X|Y], Y).
del(X, [Y|L1], [Y|L2]) :- del(X, L1, L2).
```

Το δεύτερο clause του eq τελειώνει με κόμμα που πιθανώς είναι λάθος, αλλά το παραπάνω πρόγραμμα έχει ορθή σύνταξη καθώς ο όρος που ακολουθεί το κόμμα θεωρείται ως ο επόμενος στόχος. Έτσι το πρόγραμμα αυτό είναι ισοδύναμο με το:

```
eq([], []).
eq([X|L], M) :- del(X, M, N), eq(L, N), del(X, [X|Y], Y).
del(X, [Y|L1], [Y|L2]) :- del(X, L1, L2).
```

και σίγουρα δεν πρόκειται για ένα πρόγραμμα που ελέγχει την ισότητα λιστών.

Διάφορα απλά συντακτικά λάθη και τα αντίστοιχα μηνύματα σφάλματος:

1. Ξέχασα μία παρένθεση ή μία αγκύλη

```
?- fib(14, X, D is 2*X.
ERROR: Syntax error: Unexpected end of clause
ERROR: fib(14, X, D is 2*X
ERROR: ** here **
ERROR: .

?- member(a, [a,b,c).
ERROR: Syntax error: Illegal start of term
ERROR: member(a, [a,b,c
ERROR: ** here **
ERROR: ) .
```

2. Ξέχασα ένα κόμμα

```
?- fib(14, X) D is 2*X.
ERROR: Syntax error: Operator expected
ERROR: fib(14, X)
ERROR: ** here **
ERROR: D is 2*X .
```

3. Έβαλα κενό μετά το κατηγόρημα

```
?- parent (a,b).
ERROR: Syntax error: Operator expected
ERROR: parent (a,b
ERROR: ** here **
ERROR: ) .
```

Λάθη σε υπολογισμούς: Πολλά λάθη στην Prolog, ιδιαίτερα για προγραμματιστές συνηθισμένους στο προστακτικό ή το συναρτησιακό στυλ προγραμματισμού, οφείλονται σε παράλειψη του "`is`" όταν θέλουμε να υπολογίσουμε το αποτέλεσμα μίας αριθμητικής παράστασης. Δείτε τον παρακάτω λάθος ορισμό του παραγοντικού:

```
fact(0, 1).
fact(N, F) :- N > 0, fact(N-1, F1), F is F1*N.
```

Μπορείτε να δείτε γιατί το παρακάτω goal αποτυγχάνει; Αν όχι, χρησιμοποιήστε τον debugger (`trace.`).

```
?- fact(2, F).
false.
```

2 Ασκήσεις εξάσκησης

Παραγοντικό: Γράψτε έναν tail-recursive ορισμό του παραγοντικού.

Οι πύργοι του Hanoi: Γράψτε ένα πρόγραμμα σε Prolog που να λύνει το πρόβλημα των πύργων του Hanoi.

Τρίλιζα: Γράψτε ένα πρόγραμμα σε Prolog που να βρίσκει πότε ο παίκτης "x" είτε έχει ήδη κερδίσει την τρίλιζα, ή μπορεί να κερδίσει σε μία μόνο κίνηση. Ορίστε δηλαδή τη σχέση `winner(L)`, όπου `L` μια λίστα από λίστες με την αναπαράσταση της τρίλιζας (όπου "x" και "o" τα σύμβολα των δύο παικτών και το "b" παριστάνει το κενό τετραγωνάκι). Π.χ.

```
?- winner([[x,b,o],[o,x,x],[o,b,x]]).
true .

?- winner([[x,b,o],[x,o,x],[b,b,o]]).
true .

?- winner([[x,b,o],[b,o,b],[b,b,x]]).
false.
```

Τα τρία goals αντιστοιχούν στις τρεις παρακάτω θέσεις, από αριστερά προς τα δεξιά. Στην πρώτη, ο παίκτης "x" έχει ήδη κερδίσει με τρίλιζα στη διαγώνιο. Στη δεύτερη, ο παίκτης "x" μπορεί να κερδίσει σε μία κίνηση με τρίλιζα στην πρώτη στήλη. Στην τρίτη, ο παίκτης "x" ούτε έχει ήδη κερδίσει, ούτε μπορεί να κερδίσει σε μία μόνο κίνηση.

x		o
o	x	x
o		x

x		o
x	o	x
		o

x		o
	o	
		x

Ζέβρα: Έστω το παρακάτω πρόβλημα. Υπάρχουν πέντε σπίτια.

1. Ο Εγγλέζος μένει στο κόκκινο σπίτι.
2. Ο Ισπανός έχει σκύλο.
3. Στο πράσινο σπίτι πίνουν καφέ.
4. Ο Ουκρανός πίνει τσάι.
5. Το πράσινο σπίτι είναι ακριβώς δεξιά του άσπρου σπιτιού.
6. Αυτός που καπνίζει Old Gold έχει σαλιγγάρια.
7. Στο κίτρινο σπίτι καπνίζουν Kools.
8. Στο μεσαίο σπίτι πίνουν γάλα.
9. Ο Νορβηγός μένει στο πρώτο σπίτι.
10. Αυτός που καπνίζει Chesterfields μένει δίπλα σε εκείνον που έχει την αλεπού.
11. Αυτός που καπνίζει Kools μένει στο σπίτι δίπλα σε εκείνο όπου βρίσκεται το άλογο.
12. Αυτός που καπνίζει Lucky Strike πίνει πορτοκαλάδα.
13. Ο Γιαπωνέζος καπνίζει Parliaments.
14. Ο Νορβηγός μένει δίπλα στο μπλε σπίτι.

Γράψτε ένα πρόγραμμα σε Prolog που να απαντάει στις παρακάτω δύο ερωτήσεις:

- Ποιος πίνει νερό;
- Ποιος έχει τη ζέβρα;