

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: ??/7/2008

Γελάδα (0.25 βαθμοί)

Το πρόβλημα του τηλεπαιχνιδιού “Η Γελάδα ποτέ δεν πεθαίνει” είναι γνωστό από την πρώτη σειρά ασκήσεων. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Κάποια παραδείγματα από τη χρήση του κυρίου κατηγορήματος `gelada/2` που πρέπει να περιέχει φαίνονται παρακάτω. Παρατηρήστε ότι το κατηγορήμα αποτυγχάνει αν δεν υπάρχει λύση.

```
?- gelada([t(0,0,2), t(2,1,2), t(1,1,2), t(0,3,5)], N).
```

```
N = 5 ;
```

```
No
```

```
?- gelada([t(1,1,1), t(3,1,4)], N).
```

```
N = 0 ;
```

```
No
```

```
?- gelada([t(0,0,1)], N).
```

```
No
```

TV zapping (0.25 βαθμοί)

Το πρόβλημα του TV zapping είναι γνωστό από τη δεύτερη σειρά ασκήσεων. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Κάποια παραδείγματα από τη χρήση του κυρίου κατηγορήματος `zapping/3` που πρέπει να περιέχει φαίνονται παρακάτω:

```
?- zapping(5, [3,1,3,3], N).
```

```
N = 6 ;
```

```
No
```

```
?- zapping(3, [3,1,3,3], N).
```

```
N = 3 ;
```

```
No
```

Euro 2048 (0.25 + 0.25 = 0.5 βαθμοί)

Στα περισσότερα τουρνουά ποδοσφαίρου, η διαδικασία των ομίλων κάποια στιγμή σταματάει και τα knock out παιχνίδια μεταξύ των 2^N ομάδων που έχουν περάσει από τους ομίλους αρχίζουν. Στο τέλος των knock out παιχνιδιών, μπορούμε να φτιάξουμε έναν πίνακα με τον αριθμό των knock out παιχνιδιών που κάθε ομάδα έδωσε και τα goals που έβαλε και δέχθηκε. Για παράδειγμα, στο Euro 2048 όπου από τους ομίλους μπορεί να περνάνε λιγότερες ομάδες από όσες σήμερα, η Ελλάδα για ακόμα μια φορά θα έχει αποκλειστεί σύντομα από τις τότε ομάδες-μεγαθήρια, και ο αντίστοιχος πίνακας με τους τότε τέσσερις finalists μπορεί να δείχνει ως εξής:

| | | | |
|---------------|---|----|---|
| monaco | 2 | 10 | 2 |
| andorra | 2 | 6 | 4 |
| sanmarino | 1 | 1 | 4 |
| liechtenstein | 1 | 0 | 7 |

Ο παραπάνω πίνακας μπορεί να αντιστοιχεί στα παρακάτω παιχνίδια:

| | |
|----------------------|-----|
| monaco-andorra | 3-2 |
| monaco-liechtenstein | 7-0 |
| andorra-sanmarino | 4-1 |

Το πρόγραμμά σας πρέπει να κάνει ακριβώς το παραπάνω: με βάση έναν πίνακα από τους αγώνες και τα συνολικά τέρματα υπέρ και κατά κάποιων 2^N ομάδων, βρείτε ένα σύνολο από αποτελέσματα αγώνων που αντιστοιχούν στον πίνακα. Είναι προφανές ότι η λύση μπορεί να μην είναι μοναδική, αλλά δεν είμαστε επιλεκτικοί στο ποια λύση θέλουμε.

Το πρόγραμμά σας σε Prolog πρέπει να δουλεύει ως εξής:

```
?- euro([country(monaco,2,10,2),
        country(andorra,2,6,4),
        country(sanmarino,1,1,4),
        country(liechtenstein,1,0,7)], Matches).
```

```
Matches = [match(monaco,andorra,3,2),
           match(monaco,liechtenstein,7,0),
           match(andorra,sanmarino,4,1)]
```

Yes

Προσέξτε ότι δεν είναι απαραίτητο για το πρόγραμμά σας σε Prolog να βρίσκει μόνο μία λύση – αλλά εμείς θα ελέγξουμε μόνο την πρώτη από αυτές. Τόσο η σειρά της λίστας εισόδου όσο και η σειρά της λίστας εξόδου δεν παίζουν κάποιο ρόλο. Επίσης, δεν παίζει ρόλο η σειρά με την οποία αναφέρονται οι ομάδες στους αγώνες, π.χ. ο πρώτος αγώνας θα μπορούσε να αναφέρεται και ως `match(andorra,monaco,2,3)`. Επειδή όλοι οι αγώνες είναι knock out δεν υπάρχει ισοπαλία – μέχρι το 2048 οι κανόνες του ποδοσφαίρου θα έχουν αλλάξει και οι ομάδες θα παίζουν συνεχείς παρατάσεις μέχρι κάποια από τις δύο να κερδίσει!

Τα παραπάνω ισχύουν για τη λύση σας σε Prolog. Όμως πρέπει να γράψετε το αντίστοιχο πρόγραμμα και σε κάποια άλλη γλώσσα του μαθήματος και οι επιλογές σας είναι C, ML ή Java. Η είσοδος σε ML είναι παρόμοια με αυτή σε Prolog με τις μόνες διαφορές ότι δε χρησιμοποιούνται constructors και τα ονόματα των χωρών αναπαρίστανται ως strings (π.χ. ("monaco",2,10,2)). Η έξοδος είναι μια λίστα με tuples όπως π.χ. ("monaco", "andorra",3,2).

Τα προγράμματα σε C ή Java θα πρέπει να διαβάζουν την είσοδο από το stdin ως εξής:

```
> java Euro                               ή απλώς Euro στη C
4
monaco 2 10 2
andorra 2 6 4
sanmarino 1 1 4
liechtenstein 1 0 7
```

Ο πρώτος αριθμός δίνει τον αριθμό των γραμμών του πίνακα (τις γραμμές που ακολουθούν).
Μια πιθανή έξοδος του προγράμματος σε C ή Java θα ήταν:

```
monaco - andorra 3 - 2
monaco - liechtenstein 7 - 0
andorra - sanmarino 4 - 1
```

Με άλλα λόγια αφήστε ένα κενό μεταξύ του χαρακτήρα - και της χώρας ή των goals και τουλάχιστον ένα κενό μεταξύ των χωρών και του αποτελέσματος.

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ 2 ατόμων. Μπορείτε να αλλάξετε ομάδα σε σχέση με την προηγούμενη άσκηση.
- Δεν επιτρέπεται να μοιράζεστε ασκήσεις με άλλους συμφοιτητές σας ή να βάλετε τις ασκήσεις σας σε μέρος που άλλοι μπορούν να τις βρουν εύκολα (π.χ. στο διαδίκτυο, σε ιστοτόπους συζητήσεων, ...)
- Τα προγράμματα σε Prolog πρέπει να δουλεύουν σε κάποιο από τα παρακάτω συστήματα: SWI Prolog, GNU Prolog ή YAP
- Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά (και αυτή τη φορά θα τηρηθεί αυστηρά χωρίς να υπάρχει η δυνατότητα να τα στείλετε μέσω mail στο διδάσκοντα). Θα υπάρξει σχετική ανακοίνωση στο moodle για την ακριβή διαδικασία υποβολής αλλά λογικά θα είναι σαν την προηγούμενη άσκηση.