

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 18/7/2011, 14:30¹

Σκέτη μαγεία (0.25 βαθμοί)

Το πρόβλημα με τους μαγικούς αριθμούς είναι γνωστό από την πρώτη σειρά ασκήσεων. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Κάποια παραδείγματα, αντίστοιχα αυτών της πρώτης άσκησης, από τη χρήση του κυρίου κατηγορήματος `magic/3` που το πρόγραμμά σας πρέπει να περιέχει, φαίνονται παρακάτω. Προσέξτε ότι το κατηγορήμα πρέπει να αποτυγχάνει αν δεν υπάρχει κάποια λύση (αντί να επιστρέφει 0 όπως αναφερόταν στην πρώτη σειρά ασκήσεων) ή να επιτυγχάνει το πολύ μία φορά αν υπάρχει λύση (στα παρακάτω παραδείγματα, ανάλογα με το σύστημα Prolog που θα χρησιμοποιήσετε, η γραμμή με το `fail.` μπορεί και να μην τυπώνεται).

```
?- magic(10, 3, Magic).  
Magic = 495 ;  
fail.  
?- magic(4, 6, Magic).  
Magic = 3369 ;  
fail.  
?- magic(3, 4, Magic).  
fail.
```

Βάζε γραμμές (0.25 βαθμοί)

Το πρόβλημα της τοποθέτησης ευθυγράμμων τμημάτων σε ένα ορθογώνιο πλέγμα είναι γνωστό από τη δεύτερη σειρά ασκήσεων του μαθήματος. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Κάποια παραδείγματα, αντίστοιχα αυτών της δεύτερης άσκησης, από τη χρήση του κυρίου κατηγορήματος `lines/5` που το πρόγραμμά σας πρέπει να περιέχει φαίνονται παρακάτω.

```
?- lines(2, 1, 2, 3, Lines).  
Lines = 2 ;  
fail.  
?- lines(5, 3, 3, 4, Lines).  
Lines = 48 ;  
fail.
```

Τόσο σε αυτήν την άσκηση όσο και στην προηγούμενη μπορείτε να υποθέσετε ότι οι είσοδοι/έξοδοι θα είναι τέτοιες ώστε οι ακέραιοι να χωράνε στους ακεραίους που μπορούν να αναπαρασταθούν από όλα τα συστήματα Prolog (αν και η SWI Prolog υποστηρίζει αυτόματα bignums).

¹ Η ημερομηνία υποβολής ισχύει μόνο για αυτούς που επιθυμούν βαθμό μέσα στον Ιούλιο.

MA-m-a μία (0.25+0.25 βαθμοί)

Οι **MA-m-a** μία επεξεργαστές είναι μια πολύ απλή οικογένεια επεξεργαστών που μπορούν να πάρουν μόνο **μία** ακέραια τιμή εισόδου και να εκτελέσουν μόνο δύο πράξεις σε αυτήν την τιμή: πολλαπλασιασμό με τον ακέραιο **m** και πρόσθεση με τον ακέραιο **a**. Για παράδειγμα, ο MA-2-3 επεξεργαστής όταν εκτελέσει το πρόγραμμα AAMAM με τιμή εισόδου το 3 δίνει ως αποτέλεσμα 42 (ο υπολογισμός: $3 \Rightarrow 6 \Rightarrow 9 \Rightarrow 18 \Rightarrow 21 \Rightarrow 42$) ενώ ο MA-6-4 επεξεργαστής δίνει ως αποτέλεσμα 420 όταν εκτελέσει το ίδιο πρόγραμμα με την ίδια τιμή εισόδου ($3 \Rightarrow 7 \Rightarrow 11 \Rightarrow 66 \Rightarrow 70 \Rightarrow 420$).

Έστω ότι μας έχουν δώσει τα **a** και **m** και ακέραιες τιμές **li**, **hi**, **lo** και **ho** (με αυτή τη σειρά). Ο σκοπός μας είναι να κατασκευάσουμε το μικρότερο δυνατό πρόγραμμα τέτοιο ώστε, αν η είσοδος του προγράμματος είναι σίγουρα ένας ακέραιος **x** μεταξύ του **li** και **hi** ($li \leq x \leq hi$) τότε η έξοδός του είναι ένας ακέραιος **y** μεταξύ του **lo** και **ho** ($lo \leq y \leq ho$). Σε περίπτωση που υπάρχουν περισσότερα από ένα τέτοια προγράμματα, πρέπει να επιλέξουμε εκείνο που είναι λεξικογραφικά μικρότερο αν δούμε τα προγράμματα ως συμβολοσειρές από A και M.

Η άσκηση σας ζητάει να γράψετε προγράμματα σε Prolog και στη γλώσσα της αρεσκείας σας (μεταξύ των C, ML και Java) που να λύνουν το παραπάνω πρόβλημα.

Το πρόγραμμά σας σε Prolog πρέπει να ορίζει ένα κατηγορημα που να δουλεύει ως εξής:

```
?- mama_mia(1, 2, 2, 3, 10, 20, Prog).
Prog = 'AMM' ;
fail.
?- mama_mia(1, 3, 2, 3, 22, 33, Prog).
Prog = 'MAAM' ;
fail.
?- mama_mia(3, 2, 2, 3, 4, 5, Prog).
fail.
?- mama_mia(5, 3, 2, 3, 2, 3, Prog).
Prog = '' ;
fail.
```

Στην Prolog, τα προγράμματα χρησιμοποιούν άτομα για την αναπαράστασή τους. Προσέξτε ότι το κατηγορημα `mama_mia/7` μπορεί να επιστρέφει το κενό άτομο (πρόγραμμα) ως έξοδο. Επίσης, στην Prolog το κατηγορημα πρέπει να αποτυγχάνει όταν δεν υπάρχει κάποιο πιθανό πρόγραμμα που να ικανοποιεί τους περιορισμούς εισόδου εξόδου. Στις άλλες γλώσσες το πρόγραμμά σας πρέπει να επιστρέφει το string "impossible".

Το πρόγραμμά σας σε ML πρέπει να έχει τη συμπεριφορά που φαίνεται παρακάτω:

```
- mama_mia 1 2 2 3 10 20;
val it = "AMM" : string
- mama_mia 3 2 2 3 4 5;
val it = "impossible" : string
```

Το πρόγραμμά σας σε Java (ή σε C) δέχεται τα ορίσματα ως command line arguments και πρέπει να έχει τη συμπεριφορά που φαίνεται παρακάτω – όταν το πρόγραμμα είναι κενό το πρόγραμμά σας πρέπει να τυπώνει το string empty:

```
> java MamaMia.class 1 2 2 3 10 20
AMM
```

```
> java MamaMia.class 3 2 2 3 4 5
impossible
> java MamaMia.class 5 3 2 3 2 3
empty
```

Στην Prolog πιθανώς να σας βοηθήσει το ενσωματωμένο κατηγορημα `atom_chars/2` το οποίο μετατρέπει ένα άτομο στη λίστα των χαρακτήρων από τους οποίους αποτελείται και αντίστροφα όπως φαίνεται και στις παρακάτω κλήσεις

```
?- atom_chars('MAMA', L).
L = ['M', 'A', 'M', 'A'].
?- atom_chars(A, ['M', 'A', 'M', 'A']).
A = 'MAMA'.
?- atom_chars(A, "MAMA").
A = 'MAMA'.
```

Το τελευταίο παράδειγμα δείχνει ότι οι συμβολοσειρές έχουν την ίδια αναπαράσταση με τις λίστες χαρακτήρων στην Prolog. Ανάλογο του `atom_chars/2` είναι και το ενσωματωμένο κατηγορημα `atom_codes/2`. Φυσικά, δεν υπάρχει καμία απολύτως υποχρέωση το πρόγραμμά σας να χρησιμοποιεί αυτά τα κατηγορήματα.

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ 2 ατόμων (αλλά μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με τις προηγούμενες ασκήσεις).
- 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. σε κάποια σελίδα στο διαδίκτυο, σε ιστοτόπους συζητήσεων, ...).
- Τα προγράμματα σε Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog, GNU Prolog ή YAP.
- Το πρόγραμμα στην άλλη γλώσσα μπορεί να είναι σε C, ML (SML/NJ ή MLton) ή Java.
Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στις προηγούμενες ασκήσεις. Θα υπάρξει σχετική ανακοίνωση μόλις το submission site καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλου είδους έξοδο διότι δε θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.