

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: ??/7/2015, 23:59:59

Εκρηκτικοί συνδυασμοί νέων μέτρων (0.25 βαθμοί)

Το πρόβλημα με τους εκρηκτικούς συνδυασμούς νέων μέτρων είναι γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Επειδή τα συστήματα Prolog δεν τρέχουν native code, ο χρονικός περιορισμός για την άσκηση αυξάνεται σε ένα λεπτό αντί για 10 secs. Το πρόγραμμά σας θα πρέπει να περιέχει ένα κατηγορημα `danger/2` το οποίο θα πρέπει να επιστρέφει στο δεύτερο του όρισμα το μέγιστο δυνατό πλήθος δημοσιονομικών μέτρων που μπορούν να επιλεγούν χωρίς να συμπεριλαμβάνεται σε αυτά κανένας επικίνδυνος συνδυασμός. Για το παράδειγμα της εκφώνησης της πρώτης άσκησης το κατηγορημα συμπεριφέρεται όπως φαίνεται παρακάτω¹.

```
?- danger('forbidden1.txt', Measures).  
Measures = [1,3,5,6,8] ;  
false.
```

Στην ιστοσελίδα του μαθήματος μπορείτε να βρείτε ένα πρόγραμμα Prolog που διαβάζει αρχεία με δεδομένα εισόδου αυτής της άσκησης και σας τα επιστρέφει σε μια δομή δεδομένων την οποία μπορείτε είτε να τη χρησιμοποιήσετε ως έχει ή να την τροποποιήσετε κατάλληλα για τις ανάγκες της λύσης σας.

Πράσινα, κόκκινα, κίτρινα, μπλε και γκρίζα σημεία διαπραγματεύσεως (0.25 βαθ.)

Το πρόβλημα με τις μετακινήσεις των σημείων στις διαπραγματεύσεις είναι γνωστό από τη δεύτερη σειρά ασκήσεων. Το ζητούμενο είναι να γραφεί η λύση του προβλήματος σε Prolog. Το κατηγορημα `diapragmateysi/2`, που το πρόγραμμά σας θα πρέπει να περιέχει, δέχεται μια συμβολοσειρά που αναπαριστά τον αρχικό σχηματισμό στο πρώτο του όρισμα και ενοποιεί το δεύτερο του όρισμα με μια συμβολοσειρά που αναπαριστά τον ελάχιστο αριθμό μετακινήσεων για να βρεθούμε στον τελικό σχηματισμό. Ένα παράδειγμα φαίνεται παρακάτω.

```
?- diapragmateysi("GgbrgbyGGyGr", Moves).  
Moves = "1433".
```

Όπως και στην προηγούμενη άσκηση, το όριο του χρόνου εκτέλεσης αυξάνεται σε ένα λεπτό.

Αριθμητικά τυχεροί αριθμοί (0.25+0.25 = 0.5 βαθμοί)

Στην πρώτη σειρά ασκήσεων είδαμε τους χαρούμενους αριθμούς. Σε αυτήν τη σειρά θα δούμε τους αριθμητικά τυχερούς. Σας δίνεται ένας μη αρνητικός ακέραιος και μπορείτε να τον μετατρέψετε σε μια αριθμητική έκφραση ακολουθώντας τους παρακάτω κανόνες:

¹ Σε όλα τα παραδείγματα αυτής της σειράς ασκήσεων, ανάλογα με το σύστημα Prolog που θα χρησιμοποιήσετε, στις περιπτώσεις που υπάρχει κάποια λύση, η γραμμή με το `false.` μπορεί να λείπει `fail.` ή `no` ή μπορεί να μην τυπώνεται (που μάλλον είναι το καλύτερο διότι δείχνει ότι η εκτέλεση του κατηγορημάτός σας είναι ντετερμινιστική).

1. Μπορείτε να ομαδοποιήσετε τα ψηφία του αριθμού με όποιον τρόπο θέλετε, αλλά χωρίς να αλλάξετε τη σειρά εμφάνισής τους.
2. Κάθε ομάδα ψηφίων (από την παραπάνω ομαδοποίηση) θεωρείται ένας αριθμός.
3. Μπορείτε να βάλετε οποιονδήποτε αριθμητικό τελεστή (+, -, * και /) μεταξύ των ομάδων ψηφίων.
4. Μπορείτε να βάλετε όποιες παρενθέσεις θέλετε μεταξύ των αριθμητικών υποεκφράσεων για να κάνετε τις πράξεις με την προτεραιότητα και την προσεταιριστικότητα που θέλετε.

Ένα παράδειγμα των παραπάνω είναι το $618856 \rightarrow 6 + (18 + ((8 - 5) * 6))$ που αποτιμάται σε 42.

Ένας αριθμός θεωρείται ότι είναι *αριθμητικά τυχερός* αν κανένας από τους παραπάνω συνδυασμούς αριθμητικών εκφράσεων του δεν αποτιμάται σε 100. Για παράδειγμα:

- Ο αριθμός 0 είναι προφανώς τυχερός διότι δίνει αποτέλεσμα μόνο το 0.
- Ο αριθμός 42 είναι επίσης τυχερός διότι τα αποτελέσματα των συνδυασμών των ψηφίων του είναι τα 2, 6, 8 και 42.
- Οι αριθμοί 707409 και 561709 είναι επίσης τυχεροί.
- Ο αριθμός 593347 δεν είναι τυχερός: $593347 \rightarrow 5 + ((9 / (3 / 34)) - 7) = 100$.
- Ο αριθμός 271353 δεν είναι τυχερός: $271353 \rightarrow 2 - (7 * (((1 / 3) * 5) * 3)) = 100$.

Ένας συνδυασμός πρέπει να αποτιμάται με κάποιον τρόπο σε ακριβώς 100 για να μην θεωρείται τυχερός, αλλά παρατηρήστε ότι ενδιάμεσες υποεκφράσεις μπορεί να περιέχουν κλασματικούς αριθμούς (όπως οι 593347 και 271353 παραπάνω).

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε Prolog και το άλλο σε όποια γλώσσα επιθυμείτε από τις C/C++, ML ή Java) τα οποία να παίρνουν ως είσοδο τους μια λίστα/ακολουθία από *τέσσερις το πολύ* αριθμούς και να επιστρέφουν (Prolog και ML) ή να τυπώνουν (C/C++/Java) ως έξοδο τους μια λίστα από Boolean τιμές που κάθε μία δείχνει κατά πόσο ο αριθμός στην αντίστοιχη θέση είναι αριθμητικά τυχερός ή όχι.

Περιορισμοί: $1 \leq |\text{ψηφία αριθμού}| \leq 10$. Όριο μνήμης: 2GB, χρόνου εκτέλεσης: 3 λεπτά (για Prolog), 1 λεπτό (για τις άλλες γλώσσες).

Το πρόγραμμά σας σε Prolog πρέπει να περιέχει ένα κατηγορημα `lucky_numbers/2` που να συμπεριφέρεται ως εξής:

```
?- lucky_numbers([0, 42], L).
L = [true, true].
?- lucky_numbers([707409], L).
L = [true].
?- lucky_numbers([561709, 593347, 271353], L).
L = [true, false, false].
```

Το πρόγραμμά σας σε SML/NJ πρέπει να έχει τη συμπεριφορά που φαίνεται παρακάτω:

```
- lucky_numbers [42, 707409, 271353];
val it = [true,true,false] : bool list
```

Το πρόγραμμά σας σε Java, C, C++, MLton ή OCaml πρέπει να δουλεύει όπως φαίνεται παρακάτω:

```
$ java LuckyNumbers 42 707409 271353
true true false
$ ./a.out 42 707409 271353
true true false
```

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με τις προηγούμενες σειρές ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά.
- Δεν επιτρέπεται να μοιράζεστε τα προγράμματά σας με συμφοιτητές εκτός της ομάδας σας ή να τα βάλετε σε μέρος που άλλοι μπορούν να τα βρουν (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοσελίδες συζητήσεων, ...). Σε περίπτωση που παρατηρηθούν «περίεργες» ομοιότητες σε προγράμματα, ο βαθμός των εμπλεκόμενων φοιτητών στις ασκήσεις γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.
- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. κάποιο κώδικα που διαχειρίζεται κάποια δομή δεδομένων) που βρήκατε στο διαδίκτυο στα προγράμματά σας, με την προϋπόθεση ότι το πρόγραμμά σας περιέχει σε σχόλια την παραδοχή για την προέλευση αυτού του κώδικα και ένα σύνδεσμο σε αυτόν.
- Τα προγράμματα σε Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (6.6.6), GNU Prolog ή YAP.
- Τα προγράμματα στην άλλη γλώσσα μπορεί να είναι σε C/C++, ML (SML/NJ v110.76 ή MLton 20100608 ή Objective Caml version 4.01.0) ή σε Java. Το σύστημα ηλεκτρονικής υποβολής επιτρέπει να επιλέξετε μεταξύ αυτών των γλωσσών και των υλοποιήσεων της ML.
- Ο κώδικας των προγραμμάτων σε C και ML πρέπει να είναι σε ένα αρχείο. Ο κώδικας των προγραμμάτων σε Java μπορεί να βρίσκεται σε περισσότερα του ενός αρχείου αλλά θα πρέπει να μπορεί να μεταγλωττιστεί χωρίς προβλήματα με τον Java compiler με την εντολή `javac LuckyNumbers.java`
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά όπως και στην προηγούμενη άσκηση. Θα υπάρξει σχετική ανακοίνωση μόλις το σύστημα υποβολής καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.