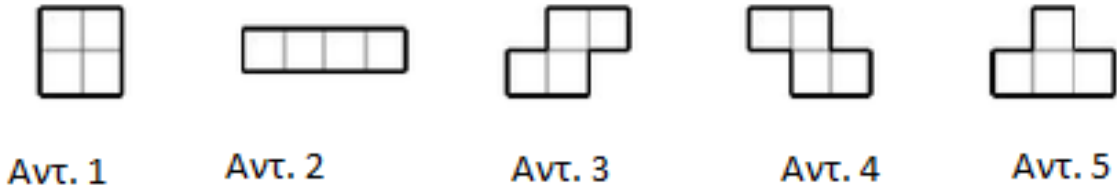


Ημερομηνία Ανάρτησης: 14/11/2017
Ημερομηνία Παράδοσης: 7/12/2017, 23:59μμ
Αρχές Γλωσσών Προγραμματισμού

1. (10%) Ο μικρός Μήτσος παίζει blackjack με τους φίλους του. Στο παιχνίδι αυτό ένας παίκτης τραβάει χαρτιά από το τραπέζι όσο το άθροισμα των χαρτιών που έχει στα χέρια του είναι μικρότερο ή ίσο από 21 ή μέχρι να πει **stop**. Στην αρχή του παιχνιδιού υπάρχουν 52 χαρτιά στο τραπέζι (13 είδη χαρτιών με 4 κάρτες από κάθε είδος). Τα 13 διαφορετικά είδη είναι τα 2,...,9, 10, Ρήγας, Βαλές, Ντάμα, και Άσσος. Οι αξίες των χαρτιών έχουν ως εξής: τα χαρτιά που έχουν αριθμό, αξίζουν όσο ο αριθμός αυτός. Ο Ρήγας, ο Βαλές και η Ντάμα αξίζουν 10, και ο Άσσος 11. Ο Μήτσος έχει ήδη τραβήξει N χαρτιά που το άθροισμά τους S είναι μικρότερο ή ίσο από 21, και έχει τώρα σοβαρούς προβληματισμούς για το αν πρέπει να τραβήξει ένα ακόμη. Έστω $X = 21 - S$. Ο φίλος μας σκέφτεται ότι δεν πρέπει να τραβήξει άλλο χαρτί αν ο αριθμός των χαρτιών στο τραπέζι με τιμή μεγαλύτερη του X είναι μεγαλύτερος ή ίσος από τον αριθμό των χαρτιών στο τραπέζι με τιμή μικρότερη ή ίση του X . Ορίστε σε Prolog τη σχέση **blackjack(L,Move)** η οποία παίρνει ως είσοδο μια λίστα L που περιέχει τις τιμές των N χαρτιών που έχει τραβήξει μέχρι τώρα ο φίλος μας και επιστρέφει τη σταθερά **Move** η οποία είναι **stop** αν πρέπει να σταματήσει ή **next** αν πρέπει να συνεχίσει. Για παράδειγμα, **blackjack([2,3,2,3,2,3],Move)** επιστρέφει **Move=stop** ενώ το **blackjack([10],Move)** επιστρέφει **Move=next**.
2. (15%) Ο μικρός Μήτσος ανακάλυψε ένα κενό ασφάλειας στους υπολογιστές του σχολείου του. Πιο συγκεκριμένα, πρόσεξε ότι αν δώσεις μια συμβολοσειρά που περιέχει ως υποσυμβολοσειρά το πραγματικό password ενός χρήστη, τότε μπορείς να μπεις χωρίς κανένα πρόβλημα στο λογαριασμό του (πχ. αν το password ενός χρήστη είναι **bob**, τότε και η συμβολοσειρά **acdbobae** θα σου επιτρέψει να μπεις στο λογαριασμό. Ο φίλος μας θέλει να βρει πόσα διατεταγμένα ζεύγη χρηστών υπάρχουν τέτοια ώστε ο πρώτος χρήστης χρησιμοποιώντας το δικό του password να μπορεί να μπει στο λογαριασμό του δεύτερου χρήστη. Ορίστε σε Prolog τη σχέση **passwords(L,N)** η οποία δεδομένης μιας λίστας L που περιέχει ως λίστες τα passwords των χρηστών, επιστρέφει τον αριθμό των διατεταγμένων ζευγών όπως περιγράφηκε παραπάνω. Το κάθε password αποτελείται από τουλάχιστον 1 και το πολύ 10 μικρά γράμματα του Αγγλικού αλφαβήτου. Για παράδειγμα, **passwords([[k,i,r],[k,i,r,t,a],[t,a],[i,r],[t]],N)** επιστρέφει $N=6$.
3. (25%) Έστω L μια λίστα που περιέχει μια μετάθεση (permutation) των αριθμών $1, \dots, N$. Επιλέγουμε μια υπολίστα της L , την αντιστρέφουμε και την ξαναβάζουμε στη λίστα. Ένα στοιχείο της νέας λίστας λέγεται **fixpoint** αν είναι ίσο με τη θέση του μέσα στη λίστα (για παράδειγμα, το 5ο στοιχείο της νέας λίστας είναι **fixpoint** αν είναι ίσο με 5). Θέλουμε να βρούμε ποιος είναι ο μέγιστος αριθμός από **fixpoints** που μπορούμε να εξασφαλίσουμε κάνοντας την παραπάνω διαδικασία. Ορίστε σε Prolog τη σχέση **fixpoints(L,F)** η οποία δεδομένης μιας λίστας L με μήκος N η οποία αποτελεί μετάθεση των αριθμών $1, \dots, N$, μας επιστρέφει το μέγιστο αριθμό F από **fixpoints** που μπορούμε να πετύχουμε. Για παράδειγμα, **fixpoints([3,2,1,4],F)** θα επιστρέψει $F=4$ (το οποίο μπορούμε να πετύχουμε αν αντιστρέψουμε την υπολίστα $[3,2,1]$).
4. (25%) Ο μικρός Μήτσος έχει ενθουσιαστεί με το νέο του ηλεκτρονικό παιχνίδι **skyscrapers**, στο οποίο υπάρχουν N ουρανοξύστες διατεταγμένοι από αριστερά προς τα δεξιά και ένας μικρός ήρωας, ο **Jumpy**. Ο i -οστός ουρανοξύστης έχει ύψος H_i και έχει στην ταράτσα του G_i χρυσά νομίσματα. Το παιχνίδι ξεκινάει με ένα αρχικό άλμα του **Jumpy** σε έναν οποιονδήποτε ουρανοξύστη και εξελίσσεται με μια σειρά από επόμενα βήματα. Σε κάθε βήμα, ο **Jumpy** μπορεί να μεταβεί σε έναν ουρανοξύστη που βρίσκεται στα δεξιά του (μπορεί να υπερπηδήσει κάποιους ουρανοξύστες) αρκεί να μην είναι πιο χαμηλός από αυτόν στον οποίο βρίσκεται. Από κάθε ουρανοξύστη στον οποίο προσγειώνεται, ο **Jumpy** μαζεύει όλα τα χρυσά νομίσματα. Ο **Jumpy** μπορεί να ολοκληρώσει το παιχνίδι μετά από οσαδήποτε βήματα (ακόμη και 0), αλλά πρέπει να έχει μαζέψει τουλάχιστον K χρυσά νομίσματα για να μπορέσει να προχωρήσει στο επόμενο επίπεδο του παιχνιδιού. Ο Μήτσος θέλει να ξέρει ποιος είναι ο αριθμός των τρόπων που μπορεί να παίξει το παιχνίδι ώστε να προχωρήσει στο επόμενο επίπεδο. Ορίστε σε Prolog τη σχέση **skyscrapers(K,L,N)** η οποία δεδομένου ενός αριθμού $K \geq 1$ και μιας λίστας L από ζεύγη (H_i, G_i) , επιστρέφει τον αριθμό των διαφορετικών τρόπων που μπορεί να παιχτεί το παιχνίδι. Για παράδειγμα, **skyscrapers(6,[(2,1),(6,3),(7,2),(5,6)],N)**, θα επιστρέψει $N=3$ (οι διαφορετικοί τρόποι είναι $\{1,2,3\}$, $\{1,4\}$ και $\{4\}$).

5. (25%) Ο μικρός Μήτσος σταμάτησε το skyscrapers και άρχισε τα πιο παραδοσιακά παιχνίδια. Άρχισε λοιπόν να ασχολείται με μια version του Tetris που υποστηρίζει τα παρακάτω αντικείμενα που είναι φτιαγμένα από τουβλάκια:



Σχήμα 1: Τα πέντε αντικείμενα του Tetris

Κάθε ένα από τα παραπάνω αντικείμενα μπορεί να περιστραφεί οσοδήποτε φορές κατά 90 μοίρες. Έστω ότι μας δίνεται μια οθόνη που περιέχει αντικείμενα της παραπάνω μορφής. Δύο αντικείμενα θα λέγονται ‘γειτονικά’ αν το ένα ακουμπάει σε κάποια πλευρά του άλλου. Θεωρούμε ότι αν δύο αντικείμενα είναι γειτονικά, τότε έχουν διαφορετικά χρώματα. Θέλουμε να γράψουμε ένα πρόγραμμα που να μας λέει πόσα αντικείμενα από το κάθε είδος υπάρχουν στην οθόνη μας. Υποθέτουμε ότι η οθόνη μας αναπαρίστανται από μια λίστα L που αποτελείται από N λίστες μήκους M η κάθε μία. Κάθε μία από τις N λίστες αναπαριστά μια γραμμή της οθόνης η οποία υποθέτουμε ότι έχει M τετραγωνάκια. Κάθε τετραγωνάκι αναπαρίστανται από ένα χαρακτήρα. Κάθε χαρακτήρας μπορεί να είναι είτε ‘o’ το οποίο αναπαριστά ένα κενό τετράγωνο στην οθόνη μας, είτε ένας οποιοσδήποτε άλλος μικρός (lowercase) χαρακτήρας του Αγγλικού αλφαβήτου ο οποίος αναπαριστά μέρος ενός αντικειμένου. Διαφορετικά γράμματα αναπαριστούν διαφορετικά χρώματα, και όλα τα τμήματα ενός δεδομένου αντικειμένου έχουν το ίδιο χρώμα. Διαφορετικές εμφανίσεις ενός αντικειμένου μπορεί να έχουν διαφορετικό χρώμα. Ορίστε σε Prolog τη σχέση `tetrisfigures(L,T)` η οποία δεδομένης μιας λίστας από λίστες L που περιγράφει την οθόνη μας, επιστρέφει μια λίστα T με 5 στοιχεία όπου το κάθε στοιχείο αντιστοιχεί στον αριθμό των εμφανίσεων του i -οστού αντικειμένου μέσα στην οθόνη που αναπαριστά η L . Για παράδειγμα:

```
tetrisfigures([ [o,c,o,o,o,o,o],
                [c,c,d,d,d,d,o],
                [c,a,a,b,b,c,c],
                [a,a,b,b,a,c,c],
                [o,o,o,a,a,a,o] ],T)
```

θα πρέπει να επιστρέφει $T = [1,1,2,1,1]$.

Παράδοση Ασκήσεων: Η παράδοση πρέπει να γίνει μέχρι τις 23:59μμ, την 7/12/2017. Θα δημιουργήσετε ένα αρχείο το οποίο θα περιέχει τις λύσεις όλων των ασκήσεων και θα το στείλετε με *email* και στις δύο παρακάτω διευθύνσεις: `antru@di.uoa.gr` και `prondo@di.uoa.gr`. Ερωτήσεις σχετικά με τις ασκήσεις θα πρέπει να απευθύνονται στο πρώτο mail (Αντώνης Τρουμπούκης). **Δεν θα υπάρξει παράταση στην παράδοση των ασκήσεων.** Τα ονόματα των κατηγορημάτων που θα χρησιμοποιήσετε στα προγράμματά σας πρέπει να είναι **ακριβώς** τα ίδια με αυτά που καθορίζονται από την παραπάνω εκφώνηση. Καθυστερημένες ασκήσεις δεν θα βαθμολογηθούν.

Σημείωση: Για να μπορέσει κάποιος να λάβει μέρος στην τελική εξέταση του μαθήματος, θα πρέπει να έχει παραδώσει τις δύο πρώτες εργασίες (Prolog και Haskell) με προβιβάσιμο βαθμό.