

7/2/2014

ΥΣ02 Τεχνητή Νοημοσύνη

ΠΡΩΤΗ ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ (μέρος 2^ο)

ΟΝΟΜΑ: Κωνσταντίνα Γαλούνη

A.M. : 1115201000034

AlienTiles:

Το πρόβλημα έχει λυθεί με τον αλγόριθμο A* χρησιμοποιώντας τις ευρετικές συναρτήσεις που πρότεινα, με βασική την h3. Η υλοποίηση έχει βασιστεί πάνω στον κώδικα Python που διατίθεται στη ιστοσελίδα <http://aima.cs.berkeley.edu/>. Σημειώνεται πως έχει δημιουργηθεί και γραφικό user interface για το πρόβλημα με τη χρήση της βιβλιοθήκης Tkinter. Οι πειραματικές αποτιμήσεις σχετικά με τον αριθμό των explored nodes και το χρόνο έχουν γίνει μέσω Unix μέσω putty.

Στην υλοποίηση κάθε χρώμα έχει αντιστοιχιστεί με έναν αριθμό ως εξής: red-0, green-1, blue-2, purple-3. Κάθε state είναι tuple, το οποίο περιλαμβάνει tuple, και 4 integers. Συγκεκριμένα το εσωτερικό tuple περιλαμβάνει n*n τιμές, οι οποίες αναπαριστούν το χρώμα κάθε tile στον πίνακα, διαβάζοντας τον πίνακα γραμμή-γραμμή. Οι integers αναπαριστούν τον αριθμό των tiles που στη συγκεκριμένη κατάσταση έχουν χρώμα red, green, blue και purple αντίστοιχα.

Η συνάρτηση actions επιστρέφει για κάθε κατάσταση όλες τις δυνατές ενέργειες που μπορούν να γίνουν, δηλαδή κάθε φορά επιστρέφει τη γραμμή και τη στήλη του tile που θα επιλεγεί.

Η συνάρτηση result επιστρέφει την κατάσταση που θα μεταβούμε από αυτή που βρισκόμαστε τώρα μέσω μιας ενέργειας a.

Τέλος, η συνάρτηση goal_test εξετάζει αν η κατάσταση στην οποία βρισκόμαστε είναι κατάσταση στόχου.

Οι ευρετικές που έχουν χρησιμοποιηθεί είναι 3, με βασική την h3. Η λειτουργία κάθε μίας έχει αναλυθεί στο γραπτό μέρος της άσκησης. Η βασική ιδέα της h3 βασίζεται στο ότι κάθε tile αλλάζει χρώμα κυκλικά από red σε green σε blue σε purple και πάλι σε red και κάθε επιλεγμένο tile επηρεάζει το χρώμα των tiles που βρίσκονται στη γραμμή και τη στήλη του. Στη γενική περίπτωση η h3 κάνει την παραδοχή ότι τα tiles που φτάνουν σε χρώμα-στόχο παύουν να εξαρτώνται από τα υπόλοιπα, εκτός από συγκεκριμένες περιπτώσεις όπου κάποια χρώματος-στόχου θα αλλάξουν και θα πάρουν το χρώμα με τη μέγιστη απόσταση από το στόχο. Όσον αφορά στο user interface, δημιουργείται ένα παράθυρο, στο οποίο εμφανίζονται:

4 κουτάκια διαφορετικού χρώματος, όπου ανάλογα με το χρώμα αυτού που θα επιλέξεις, το χρώμα αυτό θεωρείται ως στόχος.

n^*n κουτάκια, των οποίων το χρώμα αλλάζει ανάλογα με την initial state και solution state. Αρχικά όλα τα κουτάκια εμφανίζονται με κόκκινο χρώμα

1 κουτάκι help, το οποίο αν πατηθεί εμφανίζει παράθυρο με περιγραφή της λειτουργίας κάθε κουμπιού στο βασικό παράθυρο

1 κουτάκι initial state, το οποίο αλλάζει το χρώμα των n^*n παραπάνω σύμφωνα με την αρχική κατάσταση που δόθηκε στο πρόβλημα

1 κουτάκι solution, το οποίο αλλάζει το χρώμα των n^*n παραπάνω σύμφωνα με τη λύση που επιστρέφει ο A^* αλγόριθμος.

Η κατάσταση εισόδου δίνεται μέσω του αρχείου myinput.txt, το χρώμα στόχος μέσω του user interface (προεπιλογή: κόκκινο), και η διάσταση του πίνακα στατικά μέσα στον κώδικα του προβλήματος (προεπιλογή: 3). Το αρχείο με τον κώδικα ονομάζεται alientilesgui.py.

Το πρόβλημα των κύβων:

Το πρόβλημα έχει λυθεί με τον αλγόριθμο A^* χρησιμοποιώντας τις ευρετικές συναρτήσεις που πρότεινα, με βασική την $h2$. Η υλοποίηση έχει βασιστεί πάνω στον κώδικα Python που διατίθεται στη ιστοσελίδα <http://aima.cs.berkeley.edu/>. Οι πειραματικές αποτιμήσεις σχετικά με τον αριθμό των explored nodes και το χρόνο έχουν γίνει μέσω Unix μέσω putty.

Κάθε state είναι tuple, το οποίο περιλαμβάνει n διαφορετικά tuples για διάσταση n του προβλήματος. Κάθε εσωτερικό tuple συμβολίζει τη θέση κάθε κύβου ως προς το ποιος κύβος/τραπέζι βρίσκεται από κάτω και από πάνω του. Αν ένας κύβος είναι ελεύθερος, η δεύτερη τιμή του tuple που αντιστοιχεί στον κύβο αυτό είναι -1. Αν ένας κύβος βρίσκεται πάνω στο τραπέζι, η πρώτη τιμή του tuple που αντιστοιχεί στον κύβο είναι -1.

Συμβολισμός: State: ((«ο κύβος 0 έχει από κάτω τον κύβο x /τραπέζι», «ο κύβος 0 έχει από πάνω τον κύβο y /τραπέζι»), ...(),... («ο κύβος $n-1$ έχει από κάτω τον κύβο k /τραπέζι», «ο κύβος $n-1$ έχει από πάνω τον κύβο l /τραπέζι»)).

Η συνάρτηση actions επιστρέφει για κάθε κατάσταση όλες τις δυνατές ενέργειες που μπορούν να γίνουν, δηλαδή κάθε φορά επιστρέφει (i, j) : (αριθμός του κύβου που θα μεταβεί στον κύβο/τραπέζι j , αριθμός κύβου/-1 πάνω στον οποίο θα μεταβεί ο κύβος i).

Η συνάρτηση result επιστρέφει την κατάσταση που θα μεταβούμε από αυτή που βρισκόμαστε τώρα μέσω μιας ενέργειας a .

Τέλος, η συνάρτηση goal_test εξετάζει αν η κατάσταση στην οποία βρισκόμαστε είναι κατάσταση στόχου.

Οι ευρετικές που έχουν χρησιμοποιηθεί είναι 3, με βασική την $h2$, η οποία εξετάζει για τη κατάσταση που βρισκόμαστε πόσοι κύβοι βρίσκονται σε λάθος θέση σε σχέση με το ποιος κύβος/τραπέζι βρίσκεται από κάτω.

Η κατάσταση εισόδου δίνεται μέσω του αρχείου myblocks.txt, η κατάσταση στόχος μέσω του αρχείου mygoalblocks.txt και η διάσταση του πίνακα στατικά ως όρισμα από τη γραμμή εντολών. Το αρχείο με τον κώδικα ονομάζεται blocks.py.