ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

6ο εξάμηνο

Διδάσκων: Γιάννης Ρεφανίδης

1η Εργασία
26/3/2019

Ο κόσμος των κύβων (Blocks World)

Το πρόβλημα (ή ο κόσμος) των κύβων πάνω σε τραπέζι συνίσταται στην αναδιάταξη της λογικής θέσης των κύβων, μετακινώντας έναν κύβο κάθε φορά. Η «λογική» (σε αντιδιαστολή με την «φυσική») θέση ενός κύβου Α μπορεί να είναι:

- Ο κύβος Α ακουμπά πάνω στο τραπέζι (`(ontable A)`)
- Ο κύβος Α βρίσκεται πάνω σε έναν άλλο κύβο Β (`(on A B)`)

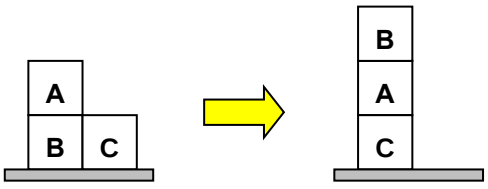
Οι κύβοι μπορούν δηλαδή να σχηματίζουν στοίβες, όπου κάθε στοίβα μπορεί να έχει από έναν έως Ν κύβους, όπου Ν το συνολικό πλήθος των κύβων στο πρόβλημα. Εάν ο κύβος Α βρίσκεται στην κορυφή μιας στοίβας, έχει καθαρή την επάνω έδρα του (`(clear A)`).

Για να αναδιατάξουμε τη διάταξη των κύβων πάνω στο τραπέζι, έχουμε τη δυνατότητα να τους μετακινούμε έναν-έναν (όχι πολλούς ταυτόχρονα δηλαδή). Για να μετακινηθεί ένας κύβος πρέπει να ισχύουν οι εξής προϋποθέσεις:

- Ο κύβος που μετακινείται έχει καθαρή την επάνω έδρα του
- Ο προορισμός της μετακίνησης είναι είτε το τραπέζι, ή ένας άλλος κύβος που έχει επίσης καθαρή την επάνω έδρα του (και που μετά τη μετακίνηση δεν θα είναι καθαρή)

Όλες οι μετακινήσεις θεωρείται ότι έχουν το ίδιο κόστος. Αναζητούμε εκείνη την σειρά μετακινήσεων που πετυχαίνει την ζητούμενη αναδιάταξη, ενώ ενδιαφερόμαστε να ελαχιστοποιήσουμε το συνολικό κόστος της αναδιάταξης.

Ως παράδειγμα, δίνεται το παρακάτω πρόβλημα με 3 κύβους:



Η λύση του προβλήματος είναι οι ακόλουθες δύο μετακινήσεις:

1. `Move(A,B,C)`
2. `Move(B, table, A)`

Για την περιγραφή των μετακινήσεων χρησιμοποιούμε την «ενέργεια» `Move` με τρία ορίσματα: Το πρώτο όρισμα είναι ο κύβος που μετακινείται, το δεύτερο όρισμα είναι η αρχική του θέση, και το τρίτο όρισμα είναι η τελική του θέση. Με τη λέξη `table` αναφερόμαστε στο τραπέζι, αν αυτό αποτελεί την αρχική ή την τελική θέση του μετακινούμενου κύβου.

Στην εργασία αυτή καλείστε να γράψετε ένα πρόγραμμα (σε όποια γλώσσα προγραμματισμού επιθυμείτε) που να λύνει προβλήματα τύπου blocks world, χρησιμοποιώντας τέσσερις διαφορετικούς αλγορίθμους αναζήτησης. Το πρόγραμμά σας θα διαβάζει το κάθε πρόβλημα από αρχείο και θα γράφει τη λύση σε αρχείο, όπως εξηγείται παρακάτω.

Μπορείτε να βρείτε αρχεία εισόδου στην παρακάτω ιστοσελίδα:

http://www.cs.colostate.edu/meps/repository/aips2000.html#blocks

Ως παράδειγμα, το αρχείο probBLOCKS-4-0.pddl περιγράφει ένα πρόβλημα με 4 κύβους:

```
(define (problem BLOCKS-4-0)
(:domain BLOCKS)
(:objects D B A C )
(:INIT (CLEAR C) (CLEAR A) (CLEAR B) (CLEAR D) (ONTABLE C) (ONTABLE A) (ONTABLE
B) (ONTABLE D) (HANDEMPTY))
(:goal (AND (ON D C) (ON C B) (ON B A))) )
```

Οι δύο πρώτες γραμμές είναι γενικές πληροφορίες που μπορείτε να αγνοήσετε. Η τρίτη γραμμή δίνει τα ονόματα των κύβων, `D`, `B`, `A` και `C`.

Η δήλωση που ξεκινά με `:INIT` περιγράφει την αρχική κατάσταση. Σε αυτήν φαίνεται ότι και οι τέσσερις κύβοι είναι καθαροί και όλοι βρίσκονται πάνω στο τραπέζι (ουσιαστικά στην αρχική κατάσταση έχουμε τέσσερις διαφορετικές στοίβες του ενός κύβου οι κάθε μια).

Τέλος η δήλωση που ξεκινά με `:goal` περιγράφει τον στόχο. Ο στόχος είναι σύζευξη (`(AND )` διάφορων επιμέρους δηλώσεων `ON`. Δηλαδή, στον στόχο πρέπει ο κύβος `D` να είναι πάνω στον `C` (`(ON D C)`), ο κύβος `C` να είναι πάνω στον `B` (`(ON C B)`) και ο κύβος `B` να είναι πάνω στον `A` (`(ON B A)`). Η προφανής βέλτιστη λύση αυτού του προβλήματος έχει τρία βήματα και είναι η:

1. `Move(B, table, A)`
2. `Move(C, table, B)`
3. `Move(D, table, C)`

Επιτρέπεται να τροποποιήσετε τη δομή των αρχείων των προβλημάτων, ώστε να είναι ευκολότερη η ανάγνωσή τους από το πρόγραμμα που θα φτιάξετε. Εάν κάνετε κάτι τέτοιο, θα πρέπει μαζί με την εργασία σας να υποβάλλετε και τα αρχεία προβλημάτων στα οποία δοκιμάσατε.

Σε κάθε περίπτωση, η έξοδος του προγράμματός σας θα πρέπει να είναι στην παραπάνω μορφή (μία εντολή ανά σειρά, αριθμημένες).

Θέμα 1ο : Αλγόριθμοι τυφλής αναζήτησης

Κατασκευάστε ένα πρόγραμμα που να λύνει προβλήματα blocks world με τους αλγορίθμους πρώτα σε βάθος (depth-first search) και πρώτα σε πλάτος (breadth-first search). Το πρόγραμμα θα δέχεται ως παραμέτρους τη μέθοδο επίλυσης (`depth` ή `breadth`), το όνομα του αρχείου περιγραφής του προβλήματος και το όνομα του αρχείου στο οποίο θα γραφεί η λύση. Για παράδειγμα, εάν το όνομα του προγράμματός σας είναι `bw.exe` (μπορείτε φυσικά να το ονομάσετε όπως αλλιώς θέλετε), θέλετε να χρησιμοποιήσετε αναζήτηση πρώτα σε βάθος, το αρχείο εισόδου είναι το probBLOCKS-4-0.pddl (ή ό,τι άλλο όνομα εσείς επιθυμείτε) και θέλετε η λύση να γραφεί στο αρχείο `solution.txt` (ή ό,τι άλλο όνομα εσείς επιθυμείτε), θα πρέπει να καλέσετε το πρόγραμμά σας με την εντολή:

```
bw.exe depth probBLOCKS-4-0.pddl  solution.txt
```

Το πρόγραμμά σας μπορεί να τυπώνει περιορισμένης έκτασης μηνύματα στην οθόνη, όπως π.χ. το χρόνο που χρειάστηκε για να λύσει το πρόβλημα, ενδεχόμενα μηνύματα λάθους (π.χ. αδυναμία επίλυσης του προβλήματος μέσα σε συγκεκριμένα χρονικά όρια, π.χ. 60 secs κλπ).

Δώστε ιδιαίτερη προσοχή στον τρόπο κλήσης του προγράμματός σας, καθώς και στη μορφή των αρχείων εξόδου, σύμφωνα με όσα περιγράφηκαν παραπάνω, ώστε το πρόγραμμά (συμπεριλαμβανομένων των λύσεων που αυτό παράγει) να μπορεί να **ελεγχθεί αυτόματα**.

Θέμα 2ο : Αλγόριθμοι πληροφορημένης (ευρετικής) αναζήτησης

Επεκτείνετε το πρόγραμμά σας ώστε να υποστηρίζει και τους αλγορίθμους ευρετικής αναζήτησης πρώτα στο καλύτερο (best-first search) και A*. Σχεδιάστε μια ευρετική συνάρτηση για το συγκεκριμένο πρόβλημα. Σκεφθείτε ότι κάθε κύβος που δεν βρίσκεται στην τελική του θέση, θα χρειαστεί τουλάχιστον μία μετακίνηση και το πολύ δύο μετακινήσεις για να μεταβεί σε αυτήν.

Καλείτε το πρόγραμμά σας επιλέγοντας αλγόριθμο `best` ή `astar` αντίστοιχα:

Θέμα 3ο : Πειραματική αξιολόγηση των αλγορίθμων

Δοκιμάστε τους αλγορίθμους αναζήτησης που υλοποιήσατε σε διάφορα προβλήματα που θα βρείτε στην διεύθυνση που σας δόθηκε στην πρώτη σελίδα. Συγκρίνετε τους χρόνους επίλυσης των αλγορίθμων και τα κόστη των λύσεων που βρήκαν και παρουσιάσετε τα αποτελέσματά σας σε μορφή εγγράφου εργασία.

Κριτήρια αξιολόγησης:

Αναζήτηση πρώτα σε βάθος	15
Αναζήτηση πρώτα σε πλάτος	15
Αναζήτηση πρώτα στο καλύτερο	15
Αναζήτηση A*	15
Υπολογιστική μελέτη	20
Γενική εικόνα (ευανάγνωστος κώδικας, σχολιασμός, ανάπτυξη προγράμματος ελέγχου λύσεων, κλπ.)	20
ΣΥΝΟΛΟ	**100**

Ο συνολικός βαθμός θα διαιρεθεί δια 100, ώστε να προκύψει ο τελικός βαθμός της εργασίας (με μέγιστη τιμή το 1).

Οδηγίες υποβολής: Η εργασία θα πρέπει να υποβληθεί μέσω του Compus. Θα υποβάλλετε ένα έγγραφο κειμένου με περιγραφή των πειραμάτων σας και των αποτελεσμάτων σας, καθώς και τα αρχεία κώδικα και προβλημάτων που υλοποιήσατε/χρησιμοποιήσατε. Ο κώδικας θα πρέπει να είναι ευανάγνωστος και σχολιασμένος.

ΠΑΡΑΡΤΗΜΑ

Ως βοήθεια σας δίνεται ενδεικτικά λυμένη παρεμφερής εργασία που αφορά το πρόβλημα N-puzzle.