



Τεχνητή Νοημοσύνη

Χειμερινό Εξάμηνο 2013-2014

Προδιαγραφές 1^{ης} Άσκησης

17 Ιανουαρίου, 2014

v1.0

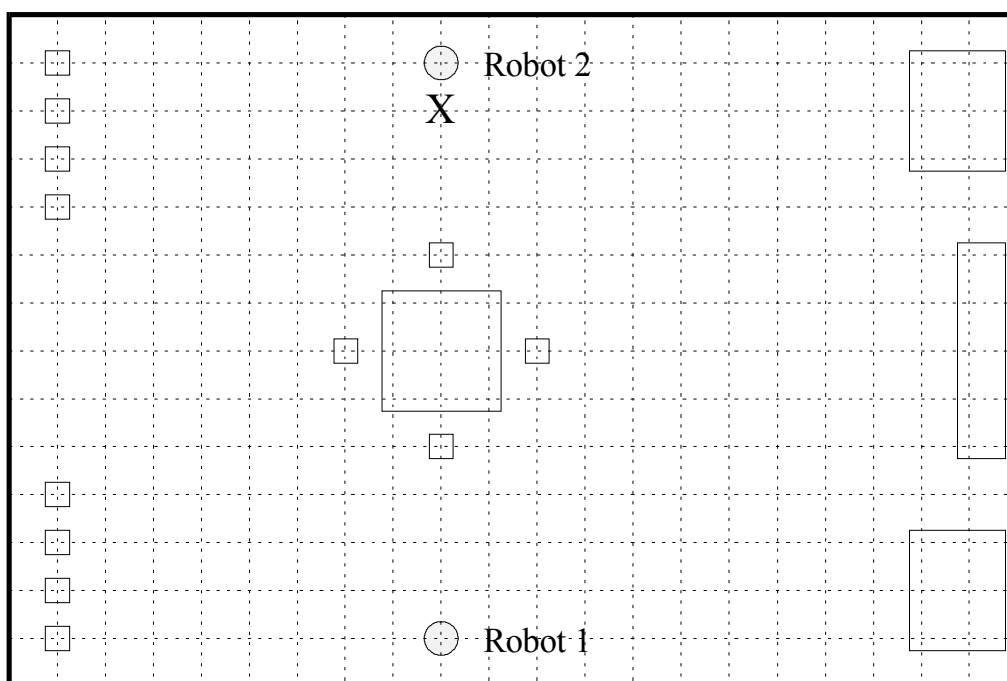
Διδάσκοντες:

Κώστας Κοντογιάννης, Αναπλ. Καθηγητής

Γιώργος Στάμου, Επικ. Καθηγητής

1. Γενική Περιγραφή Έργου

Σε αυτή την άσκηση θα σχεδιάσουμε και θα υλοποιήσουμε μια εφαρμογή που βασίζεται σε αλγόριθμους εύρεσης λύσης σε χώρο καταστάσεων. Το σενάριο είναι ότι έχουμε μια αίθουσα όπου ένα robot (1) προσπαθεί να συναντηθεί με ένα άλλο robot (2). Τα Robot κινούνται στο χώρο της αίθουσας του μουσείου που περιέχει εμπόδια (π.χ. καθίσματα για τους επισκέπτες, εκθέματα κλπ.). Τα Robot έχουν αποθηκευμένη στη μνήμη τους τη κάτοψη της αίθουσας του μουσείου. Το Robot (1) κινείται με ταχύτητα διπλάσια του Robot (2). Το Robot (1) γνωρίζει κάθε φορά που βρίσκεται και που έχει πάει το Robot (2). Το Robot (2) κινείται πρώτο (προσπαθώντας να αποφύγει το Robot (1)) και το Robot (1) οργανώνει την αναζήτησή της πορείας του κάθε φορά με βάση την παρούσα θέση του Robot (2). Επειδή το Robot (1) κινείται με ταχύτητα διπλάσια από αυτή του Robot (2) σύντομα θα το συναντήσει. Ουσιαστικά είναι ένα πρόβλημα όπου σε κάθε κίνηση έχουμε και ένα νέο πρόβλημα αναζήτησης (στόχος αλλάζει – θέση Το Robot (2)). Τα Robot μπορούν να κινηθούν μπροστά, πίσω, αριστερά και δεξιά (όχι διαγώνια). Θεωρούμε ότι το Robot (2) κινείται με ταχύτητα 1 μιας θέσης ανά κίνηση ενώ το Robot (1) με ταχύτητα δύο θέσεων ανά κίνηση. Ένα παράδειγμα της κάτοψης μίας αίθουσας δίνεται στο Σχήμα 1.



Σχήμα 1. Κάτοψη χώρου

Για αυτή την εργασία θα υλοποιήσουμε το αλγόριθμο A^* για να υπολογίσουμε την πορεία που πρέπει να επιλέξει ένα Robot (π.χ. το Robot 1) από τη τρέχουσα θέση του, σε κάποια νέα θέση X του Robot (2). Για αυτή την εργασία θα υλοποιήσουμε τον A^* με τη χρήση υπο-εκτιμητή (admissible heuristic) και με τη χρήση υπερ-εκτιμητή (non-admissible heuristic). Επίσης θα θεωρήσουμε ότι η κάτοψη του χώρου διαβάζεται από το πρόγραμμά μας από κάποιο αρχείο. Η μορφή του αρχείου δίνεται παρακάτω, στο Σχήμα 2. Για διαφορετικά αρχεία κατόψεων θα υπολογίσουμε το χρόνο εκτέλεσης του αλγόριθμου σαν συνάρτηση του αριθμού των καταστάσεων που ανοίξαμε για πιθανή επέκταση.

2. Μορφή Αρχείου Κάτοψης Χώρου

Θεωρούμε ότι η κάτοψη της αίθουσας δίνεται με τη μορφή ενός αρχείου κειμένου. Αυτό το αρχείο ορίζει τις διαστάσεις της αίθουσας, την αρχική θέση του Robot, τη τελική θέση που πρέπει να βρεθεί το Robot, καθώς και τα σημεία με τα εμπόδια στο χώρο. Όλες οι συντεταγμένες δίνονται σε χώρο δύο διαστάσεων (x, y). Ένα παράδειγμα τέτοιου αρχείου για το χώρο που περιγράφεται στο Σχήμα 1, δίνεται παρακάτω στο Σχήμα 2. Σε αυτό το παράδειγμα, ο χώρος έχει διαστάσεις 20 X 13, η αρχική θέση του Robot είναι η (13, 9) και η τελική θέση είναι η (2, 9). Θεωρούμε ότι η αρχική και η τελική θέση είναι σημεία που δεν είναι κατευλημμένα από εμπόδια. Στο αρχείο της κάτοψης σημειώνουμε με O τα κενά σημεία που δεν καταλαμβάνονται από κάποιο εμπόδιο, και με X τα σημεία του χώρου που καταλαμβάνονται από κάποιο εμπόδιο.

```
20 13
13 9
2 9
XOooooooooXoooooooooXX
XOooooooooooooooooooooXX
XOooooooooooooooooooooXX
XOooooooooooooooooooooO
OooooooooXooooooooooooX
OooooooooXXooooooooooooX
OooooooooXXXXXoooooooooX
OooooooooXXooooooooooooX
OooooooooXooooooooooooX
XOooooooooooooooooooooO
XOooooooooooooooooooooXX
XOooooooooooooooooooooXX
XOooooooooooooooooooooXX
```

Σχήμα 2. Μορφή αρχείου

3. Σχόλια για την Υλοποίηση του Έργου

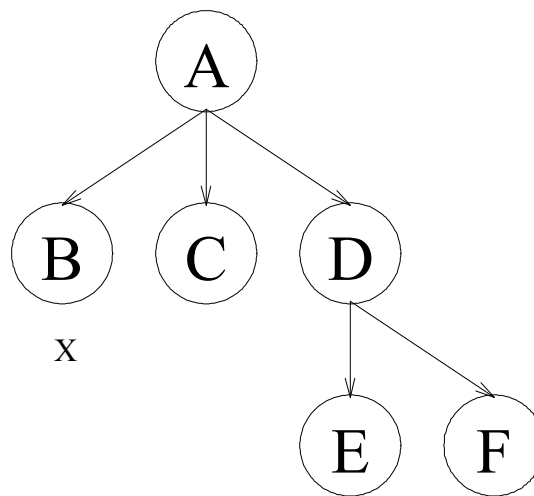
Η επίλυση του προβλήματος έχει τρία συστατικά μέρη:

1. *Τη μοντελοποίηση του χώρου των καταστάσεων.* Η μοντελοποίηση του χώρου των καταστάσεων θα γίνει με τη μορφή μιας δομής δεδομένων όπως περιγράφεται παρακάτω.
2. *Τους τελεστές μετακίνησης από μια κατάσταση σε μια άλλη.* Οι τελεστές μετακίνησης θα υλοποιηθούν με συναρτήσεις που παράγουν συνεπείς ακόλουθες καταστάσεις από μια δεδομένη κατάσταση.
3. *Τη στρατηγική αναζήτησης λύσης στο χώρο καταστάσεων.* Αυτή η στρατηγική υλοποιείται από κάποιο αλγόριθμο αναζήτησης όπως ο A* ή ο Hill Climbing.

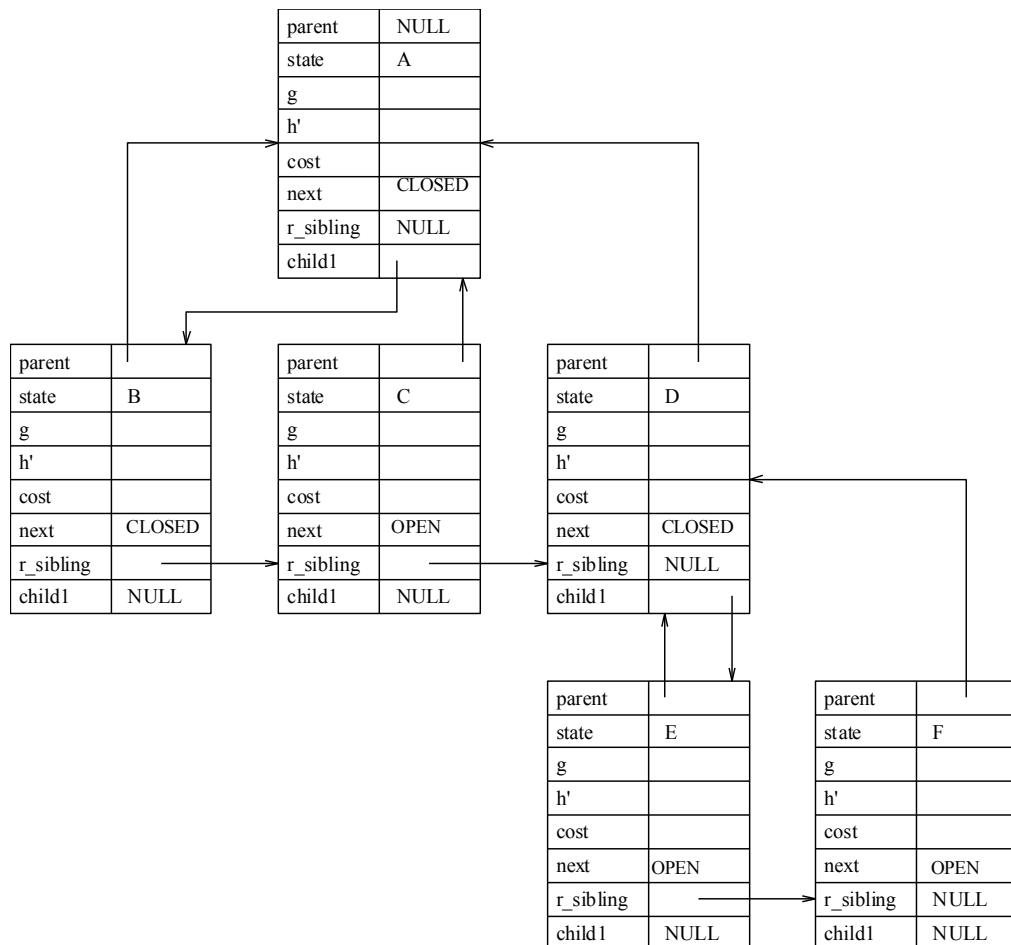
Η βασική δομή δεδομένων που προτείνεται για τη μοντελοποίηση των καταστάσεων του χώρου και του δένδρου αναζήτησης ορίζεται παρακάτω στη γλώσσα προγραμματισμού C, αλλά είναι στη διακριτική σας ευχέρεια να υλοποιήσετε το χώρο με άλλη δομή δεδομένων ή και σε άλλη γλώσσα προγραμματισμού όπως Java ή C++. Μπορείτε να χρησιμοποιήσετε τις δικές σας δομές δεδομένων ή να επεκτείνεται τις δοθείσες.

```
typedef struct Node {  
    struct Node *parent; /* Parent of Current Node*/  
    state_type state; /* State of Problem*/  
    g_type g; /* g-value of current node - κόστος μετάβασης */  
    h_type h; /* h'-value of current node - υπολειπόμενη απόσταση*/  
    cost_type cost; /* cost of last operator*/  
    struct Node *next; /* OPEN/CLOSED -  
                        /* είναι CLOSED όταν ο κόμβος δεν επεκτείνεται άλλο*/  
    struct Node *r_sibling; /*Pointer to Right Sibling*/  
    struct Node *child1; /*Pointer to Leftmost Child*/  
} SearchGraphNode;  
  
SearchGraphNode *OPEN, *CLOSED; /* Χρησιμοποιείται σαν flag για την */  
/* επέκταση ή όχι του κόμβου */
```

Ένα παράδειγμα για το πώς μοντελοποιείται το δένδρο αναζήτησης του Σχήματος 3, με τη χρήση της προτεινόμενης δομής, παρουσιάζεται παρακάτω στο Σχήμα 4.



Σχήμα 3. Παράδειγμα δένδρου αναζήτησης



Σχήμα 4. Παράδειγμα μοντελοποίησης δένδρου αναζήτησης

4. Παραδοτέα

Για αυτό το έργο θα παραδώσετε σε έντυπη μορφή μέχρι τις 7 Φεβρουαρίου μια σύντομη έκθεση (περίπου 6 σελίδες) που περιγράφει:

1. Το πρόβλημα που επιλύετε.
2. Τις δομές δεδομένων που επιλέξατε.
3. Τις βασικές συναρτήσεις-τελεστές που σχεδιάσατε για την υλοποίηση του A^* .
4. Τις ευριστικές μεθόδους που θεωρήσατε (ένα υπο-εκτιμητή – admissible heuristic και ένα υπερ-εκτιμητή non-admissible heuristic).
5. Στατιστικά αποτελέσματα από έξι διαφορετικά αρχεία εισόδου με τη μορφή γραφήματος όπου στον άξονα X θα έχετε τον συνολικό αριθμό των κόμβων του δένδρου (δένδρων) που παράχθηκαν κατά την αναζήτηση, και στον άξονα Y τον συνολικό χρόνο εκτέλεσης του προγράμματος. Τα στατιστικά αποτελέσματα θα παραχθούν μια φορά (έξι αρχεία) με τη χρήση υπο-εκτιμητή (admissible heuristic) και μια φορά (έξι αρχεία) με τη χρήση υπερ-εκτιμητή (non-admissible heuristic).
6. Συνοπτική ανάλυση και σχολιασμό των αποτελεσμάτων
7. Τον πηγαίο κώδικα του προγράμματός σας. **Ο πηγαίος κώδικας και η εργασία θα παραδοθούν και σε ηλεκτρονική μορφή σύμφωνα με τις οδηγίες που θα αναρτηθούν στην ιστοσελίδα του μαθήματος.**