

TP02 : Recherche dans un espace d'états

Livrables attendus : un fichier lisp + un rapport présentant et argumentant les réponses aux questions

Date de remise : 16 novembre 2015 à 18h

Enoncé du problème

Nous considérons un monde avec 4 pions (A,B,C,D) non superposables. Ils peuvent être arrangés dans n'importe quel ordre, sauf A qui ne peut pas être plus à droite que D. Par exemple, ABCD et CBAD sont deux états possibles du monde, tandis que DCBA et CDAB ne sont pas possibles.

Le monde peut être manipulé par une action de la forme `echange(x, y)` qui échange les pions des positions `x` et `y`. Par exemple, `echange(1, 2)` transforme BCAD dans CBAD. Seules les actions `echange(1, 2)`, `echange(2, 3)` et `echange(2, 4)` sont autorisées. Elles donnent un successeur uniquement si la situation atteinte est possible.

- 1) Dessiner le graphe d'états.
- 2) Ecrire en Lisp la fonction `echange(x y)`.
- 3) Ecrire la fonction `successeurs(etat)` qui retourne la liste des successeurs (autorisés) d'un état donné.
- 4) En utilisant les fonctions précédentes, écrire un algorithme puis une fonction de recherche en profondeur d'abord.
- 5) Déterminer à l'aide de cette fonction le chemin menant de l'état de départ ADBC à l'état CBAD.
- 6) Déterminer une heuristique permettant de choisir, parmi les successeurs d'un état donné, l'état qui semble le plus prometteur pour amener à l'état final.
- 7) Programmer la fonction `choixEtat(listeEtats)` correspondante, ainsi que la fonction de recherche correspondante.
- 8) Comparer les chemins obtenus avec les deux fonctions de recherche. Discuter l'intérêt de l'heuristique choisie.