
IA01 : Rapport du TP2

Auteurs :
Camille GERIN-ROZE
Thomas PERRIN

Le 15 Novembre 2015

Table des matières

1	Graphe d'états	3
---	----------------	---

Introduction

Le but de ce TP est de réfléchir à une recherche dans un espace d'états avec le langage Lisp. Au départ, nous effectuerons une recherche basique en profondeur, pour ensuite considérer une heuristique afin d'optimiser cette recherche. Dans ce rapport, nous répondrons aux différentes questions posés et nous expliquerons de quel manière nous avons abordé les problèmes et pourquoi. Nous utiliserons plusieurs fonctions dites « de service » :

- `get_symbol` : Permet d'obtenir le symbole à l'index en paramètre.
- `myMember` : Implémentation différente de la fonction `member` avec la fonction `EQUAL` au lieu de `EQ`.
- `etat_correct` : Permet de déterminer si un état est correct en vérifiant la présence de A, B, C, D, que la structure soit de taille 4 et que A soit placé avant D.

Chapitre 1

Graphe d'états

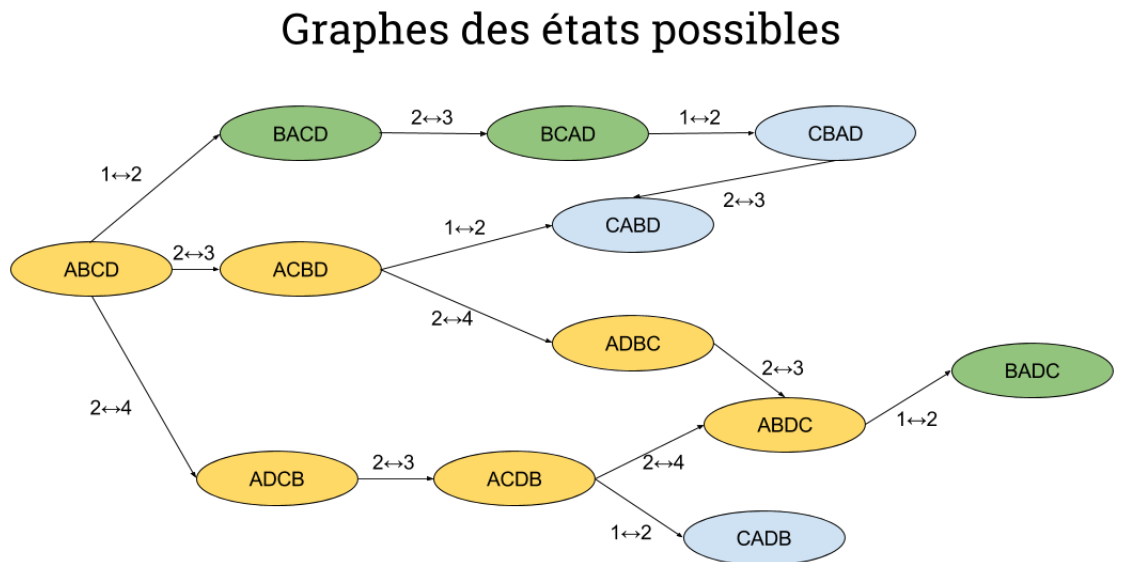


FIGURE 1.1 – Graphe représentant les états et leurs successeurs

De base, seules trois actions sont possibles à chaque état. Ensuite, à chaque successeur, on retire la même action qui va uniquement revenir en arrière. De plus, certaines actions peuvent être interdites puisqu'ils modifieraient les pions dans un état non autorisé. Ainsi, ce graphe d'états permet de représenter clairement les différentes possibilités en indiquant unitairement chaque état.

Chapitre 2

Successeurs d'un état

2.1 La fonction `echange(x,y)`

La fonction `echange` reforme les 4 pions à partir d'une liste vide. A chaque itération, on ajoute le pion correspondant, mis à part lorsque l'on est aux deux index précisés en paramètre. On vérifie ici uniquement que les deux index sont accessibles, on limitera les actions possibles dans la fonction `successeurs`.

2.2 La fonction `successeurs`

La fonction `successeurs` effectue les trois actions autorisés par l'énoncé sur l'état en paramètre : c'est-à-dire `echange(1,2)`, `echange(2,3)` et `echange(2,4)`. Si le successeur est valide (vérifié grâce à la fonction de service `etat_correct`), *on l'ajoute à la liste des successeurs. Ainsi*