



Cours 2 - Représentation symbolique et langage Lisp

MARIE-HÉLÈNE ABEL

Table des matières

I - Représentation d'une base de données au moyen de listes	5
II - Représentation et manipulation de graphes	7
A. Représentation d'un graphe.....	7
B. Manipulation d'un graphe.....	8
III - Recherche dans un espace d'états	11
A. Définition formelle d'un problème de recherche.....	11
B. Exemple : le jeu du TAQUIN.....	11

Représentation d'une base de données au moyen de listes



Remarque

Il est possible de représenter très facilement des informations courantes à l'aide de simples listes.



Exemple

```
>(setq evenement '(Albertine va au marche))
(ALBERTINE VA AU MARCHE)
>(setq Albertine '(personne de sexe feminin))
(PERSONNE DE SEXE FEMININ)
>(setq personne '(etre_humain))
(ETRE_HUMAIN)
>
```



Méthode

Les informations sont liées automatiquement entre elles par le fait même qu'un élément d'une liste est un atome qui reçoit une valeur.



Remarque

La primitive **symbol-value** permet d'afficher la valeur d'un atome.



Exemple

```
>evenement
(ALBERTINE VA AU MARCHE)
>(car evenement)
ALBERTINE
>(symbol-value (car evenement))
(PERSONNE DE SEXE FEMININ)
>(symbol-value (car (symbol-value (car evenement))))
(ETRE_HUMAIN)
>
```



Remarque

On voit qu'il est très facile de construire des listes d'atomes représentant des situations ou des relations entre objets.

Représentation et manipulation de graphes

Représentation d'un graphe

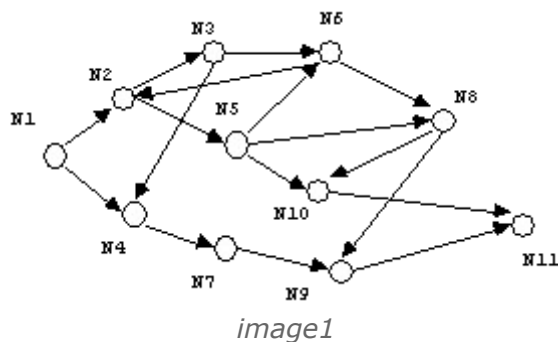
7

Manipulation d'un graphe

8

A. Représentation d'un graphe

Soit le graphe orienté *G1* suivant :



Remarque

On peut représenter le graphe *G1* par la liste de ses sommets



Exemple

```
>(setq *G1* '(N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11))
(N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11)
>
```



Méthode : Principes de représentation

Chaque sommet est représenté à son tour par un atome dont la valeur est la liste de ses successeurs.



Exemple

```
>(setq N1 '(N2 N4)
      N2 '(N3 N5)
      N3 '(N4 N6)
      N4 '(N7)
      N5 '(N6 N8 N10))
```

```
N6 '(N2 N8)
N7 '(N9)
N8 '(N9 N10)
N9 '(N11)
N10 '(N11)
N11 '()
NIL
>
```

B. Manipulation d'un graphe

Préambule

Il est souvent utile (et fortement recommandé) de définir des fonctions de service permettant de s'affranchir du format exact choisi pour la représentation du graphe ou des sommets.

=> On réalise ainsi une abstraction de l'information.



Méthode

On peut par exemple écrire :

- Une fonction permettant d'obtenir les successeurs d'un sommet particulier,
- Une fonction qui teste si un sommet particulier fait partie du graphe spécifié,
- Une fonction qui donne le sommet suivant un sommet donné dans l'ordre de définition de ceux-ci.



Exemple

```
>(defun successors (node)
(symbol-value node))
SUCCESSORS
>(defun node-list (graph)
(symbol-value graph))
NODE-LIST
>(defun in-graph? (node graph)
"Teste si un sommet fait partie du graphe"
(member node (symbol-value graph)))
IN-GRAPH?
>(defun next-node (node graph)
"Donne le sommet suivant un sommet donné dans l'ordre de définition des
sommets du graphe. (next-node nil <graphe>) retourne le premier sommet du
graphe"
(if node (cadr (member node (symbol-value graph))) (car (symbol-value graph))))
NEXT-NODE
>
```

Recherche dans un espace d'états



Définition formelle d'un problème de recherche

11

Exemple : le jeu du TAQUIN

11

Considérons un système interagissant avec son milieu :

- On représente l'ensemble des situations que le système peut rencontrer par un ensemble d'états accessibles S .
- Dans chaque état, le système est susceptible d'effectuer un certain nombre d'actions appartenant à l'ensemble A .
- Après la réalisation d'une certaine action depuis l'état $s \in S$, le système se retrouve dans l'état $s' \in S$.

A. Définition formelle d'un problème de recherche



Définition

- Q un ensemble non vide d'états
- $S \subseteq Q$ un ensemble non vide d'états initiaux
- $G \subseteq Q$ un ensemble non vide d'états-solutions (pouvant être définis explicitement ou implicitement par un test solution)
- A un ensemble d'actions
- Successeurs : $Q \times A \rightarrow Q$
- Fonction coût : $Q \times A \times Q \rightarrow \mathbb{R}^+$ (défini uniquement pour des états qui sont successeurs l'un de l'autre)

B. Exemple : le jeu du TAQUIN

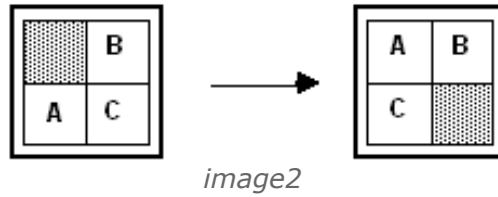


Méthode : Principes de base

Modification de configurations par déplacement d'une tuile dans un damier.



Exemple



Représentation de l'espace d'états : graphe d'états

- Un état : un damier
- Objectif : un damier particulier
- Actions : déplacement d'une tuile d'une case vers la gauche, la droite, le haut ou la droite



Méthode : Ensemble des déplacements possibles : coups permis

- 1-1-D: déplacer la tuile en position 1-1 vers la droite
- 1-1-B: déplacer la tuile en position 1-1 vers le bas
- 2-1-G: déplacer la tuile en position 2-1 vers la gauche
- 2-1-B: déplacer la tuile en position 2-1 vers le bas
- 1-2-D: déplacer la tuile en position 1-2 vers la droite
- 1-2-H: déplacer la tuile en position 1-2 vers le haut
- 2-2-G: déplacer la tuile en position 2-2 vers la gauche
- 2-2-H: déplacer la tuile en position 2-2 vers le haut



Méthode : Classés en fonction de la position de l'emplacement vide

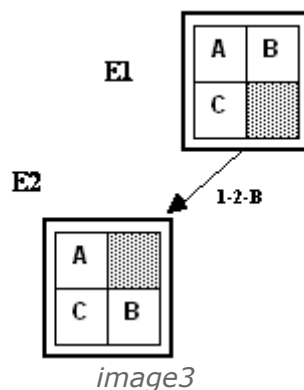
- 1-1: 1-2-G 2-1-H
- 1-2: 1-1-D 2-2-H
- 2-1: 1-1-B 2-2-G
- 2-2: 1-2-B 2-1-D

Recherche

Consiste à passer d'un état du damier à l'autre en jouant l'un des coups permis.



Exemple



Définition : Recherche en profondeur d'abord ou "depth first".

Consiste à continuer la recherche toujours à partir de la dernière position atteinte.



Exemple

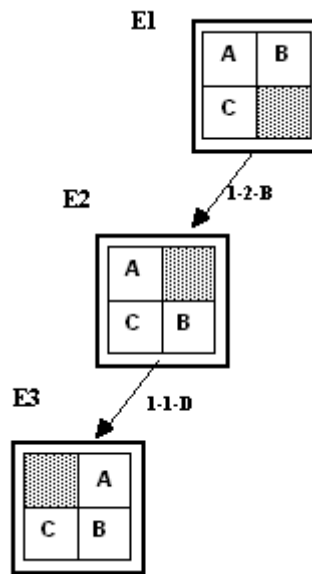


image4



Définition : Recherche en largeur d'abord ou "breadth first".

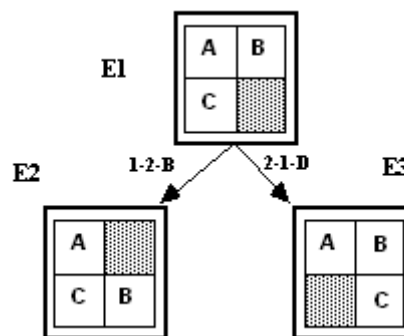


image5

Graphe des états complets du jeu

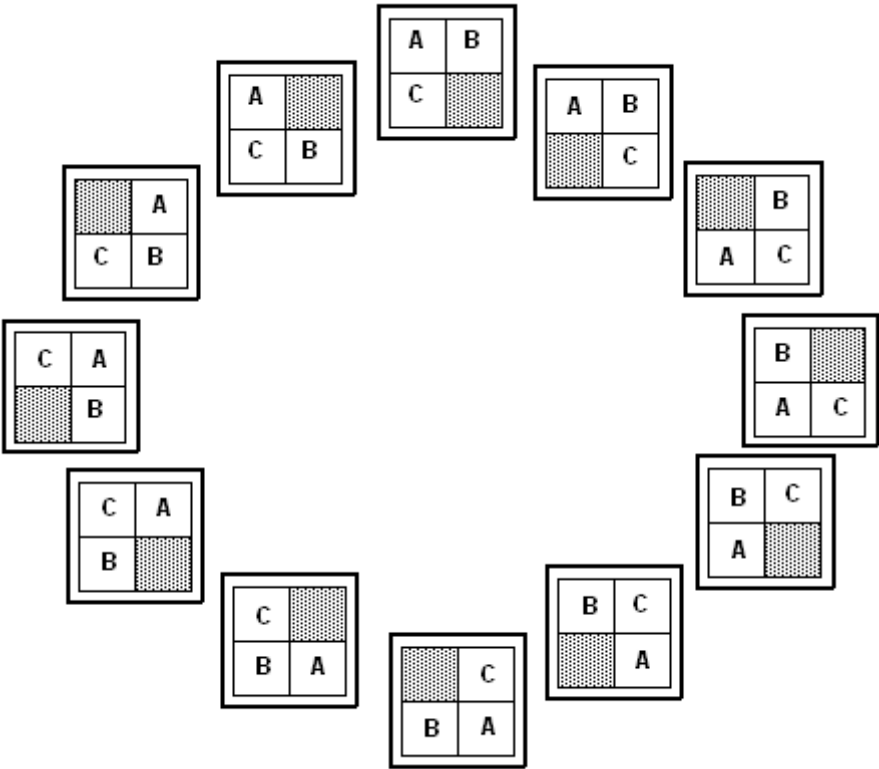


image6