

Tours de Hanoï

Nous avons n disques D_1, D_2, \dots, D_n avec un trou au centre et trois piquets, A, B et C sur lesquels les n disques peuvent être placés. Le disque D_n est plus grand que le disque D_{n-1} qui lui-même est plus grand que le disque D_{n-2} , de sorte que, pour tout $\forall i \in [1, n-1]$ D_i est plus petit que D_{i+1} . Initialement, les disques se trouvent sur le piquet A , avec D_1 au dessus de D_2 qui est lui-même au-dessus de D_3 et ainsi de suite jusqu'à D_n qui se trouve par terre. Nous voulons les déplacer sur le piquet C dans la même configuration (D_1 sur D_2 qui est lui-même sur D_3 et ainsi de suite jusqu'à D_n). Les règles suivantes s'appliquent :

1. seul le disque qui se trouve au sommet d'un piquet peut être déplacé
2. un disque ne peut pas être placé sur un disque plus petit (par exemple, D_3 ne peut pas être placé sur D_1 , ni même sur D_2)

Remarque : on fera appel à 2 prédicats $sur(Disque, Piquet)$ et $au-dessus(disque_i, disque_j)$



FIGURE 1 – Tours de Hanoï avec 6 disques

Exercice 1 Planification avec STRIPS

Rappelons qu'une instance d'un problème de planification se décrit en STRIPS de la façon suivante :

- Un état initial
- La spécification de l'état terminal? situation que la planificateur essaie d'atteindre
- Un ensemble d'actions dont la description comprend, pour chacune,
- Une précondition
- Une liste de destruction
- Une liste d'ajout

Remarque : nous nous placerons ici dans le cas où le nombre de disque n est égal à 3.

1. *Etat initial*
Formulez l'état initial dans le formalisme STRIPS.
2. *Etat terminal*
Formulez l'état terminal toujours dans le formalisme STRIPS.
3. *Opérateurs*
Formulez le(s) différent(s) opérateur(s) en STRIPS
4. *Ebauche de résolution*
Simuler les trois premiers déplacements de disques en STRIPS
Remarque : on indiquera l'évolution de la pile de buts et de l'état à chaque étape.

Exercice 2 Planification en ASP

On reprend ici la programmation de la planification en ASP telle qu'elle avait été vue en cours. Pour cela, il y a quatre composants : (1) représentation des objets et des états, (2) buts, (3) spécifications des mouvements, (4) règle de génération.

1. *Représentation des objets*

Représenter les objets, à savoir les instants, les 3 disques et les 3 piquets, en supposant qu'on les représente tous avec des entiers naturels. Noter que les instants sont bornées par une constante appelée *fin*.

2. *Représentation des états*

Décrire l'état initial avec deux prédicats, $sur(D, P, T)$ qui indique que le disque D se trouve sur le piquet P à l'instant T .

3. *Représentation du but*

Représenter le but à l'aide de contraintes du type : $\neg notpred(I, J)$. lorsque $pred(I, J)$ est un prédicat que l'on souhaite satisfaire.

Rappelons que si, à l'instant initial, les disques se trouvent tous sur le piquet 1, à l'instant final, ils doivent se trouver sur un autre piquet, par exemple 3.

4. *Programmation de l'effet des mouvements*

Sachant qu'il n'y a qu'un seul type de mouvement qui correspond au déplacement d'un piquet sur un autre, programmer, en ASP, l'effet d'un mouvement.

5. *Inertie*

Donner, en ASP, les clauses qui spécifient l'inertie des prédicats qui ne sont pas affectés par l'action.

6. *Spécification des contraintes*

Indiquer, en ASP, les clauses qui contraignent les mouvements, par exemple qu'un disque ne peut se trouver sous un disque plus grand, ou qu'un disque ne peut être déplacé s'il y a un disque au dessus.

7. *Règle de génération*

Programmer en ASP une clause qui engendre les mouvement, sachant qu'il n'y en a pas plus d'un à chaque instant. Pour cela, on fera appel à une énumération contrainte, comme cela a été vu en cours.

Exercice 3 Ensembles fréquents

Dans un premier temps, nous ferons appel aux algorithmes *Apriori* et *FP – Growth* vus en cours pour engendrer tous les ensembles fréquents de l'ensemble d'apprentissage A donné dans le tableau 1 avec $SupMin = 0,3$.

1. Donner la liste des items booléens de l'ensemble A .
2. Rappeler les étapes de l'algorithme *Apriori*
3. Engendrer tous les ensembles fréquents à l'aide de l'algorithme *Apriori*
4. Rappeler les étapes de l'algorithme *FP – Growth*
5. Ordonner les items booléens de l'ensemble A
6. Construire le *FP – tree* de l'ensemble A
7. Indiquer comment on peut extraire les ensemble fréquents à partir de ce *FP – tree*.

Exercice 4 Extraction de règles d'association

1. Indiquer comment on extrait les règles d'association à partir d'un ensemble fréquent
2. Engendrer trois règles d'association de confiance 1,00 à partir des ensembles fréquents trouvés à la question précédente.

Exercice 5 Construction d'un arbre de décision

1. Rappeler comment on détermine le gain d'un attribut à partir de l'entropie pondérée par le nombre d'éléments et expliquer à quoi correspond cette formule.

	A	B	C	Classe
E_1	V_a	V_a	V_c	+
E_2	V_a	V_c	V_a	+
E_3	V_b	V_b	V_a	-
E_4	V_c	V_a	V_c	+
E_5	V_c	V_b	V_c	-
E_6	V_c	V_a	V_b	-
E_7	V_c	V_c	V_b	-
E_8	V_a	V_c	V_c	+

TABLE 1 – Ensemble d'apprentissage A

2. Construire l'arbre de décision qui discrimine tous les exemples de l'ensemble d'apprentissage A (voir tableau 1) pour les deux classes '+' et '-' en faisant appel à la méthode de construction employée par l'algorithme ID3.

Remarque : on s'aidera du tableau 2 fourni en annexe : il donne, pour un ensemble E qui contient p éléments de la classe '+' et n éléments de la classe '-', l'entropie pondérée, à savoir $-(n+p) * [\frac{p}{n+p} \cdot \log_2(\frac{p}{n+p}) + \frac{n}{n+p} \cdot \log_2(\frac{n}{n+p})] = |E| * [Pr('+') \cdot \log_2(Pr('+')) + Pr('-') \cdot \log_2(Pr('-'))]$

Annexe

Représentation des actions en STRIPS.

Les schémas d'action sont vus comme des opérateurs permettant de passer d'un état à un autre. Le langage utilisé pour décrire les actions est un fragment très restreint de la logique du premier ordre.

action(paramètres)	
precond	conjonction de littéraux positifs sans symbole de fonction qui exprime ce qui doit être vrai pour que l'action puisse être exécutée
delete	conjonction de littéraux positifs sans symbole de fonction qui exprime ce qui n'est plus vrai après la réalisation de l'action
precond	conjonction de littéraux positifs sans symbole de fonction qui exprime ce qui est rendu vrai par l'exécution de l'action

De plus, l'hypothèse cruciale suivante est faite : *toute formule non mentionnée dans la description de l'action reste inchangée.*

Entropie pondérée.

Le tableau donne l'entropie pondérée d'un ensemble E qui contient p éléments de la classe '+' et n éléments de la classe '-'.

p/n	1	2	3	4	5
5	3,90	6,04	7,63	8,92	10,00
4	3,61	5,51	6,90	8,00	
3	3,25	4,85	6,00		
2	2,75	4,00			
1	2,00				

TABLE 2 – Entropie pondérée