

Rappels : Aucun document n'est autorisé. Le barème n'est donné qu'à titre indicatif.

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, $\forall i \in [1, n-1]$ 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 sur la table. 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 sur D_2).

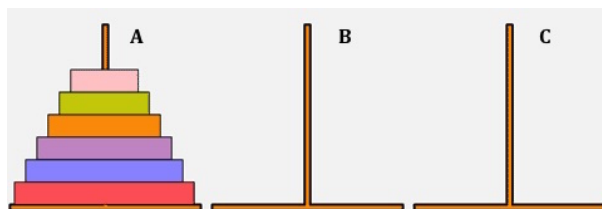


FIGURE 1 – Tours de Hanoï avec 6 disques

Exercice 1 Planification avec STRIPS (Noté sur 7 points)

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,
- un ensemble d'actions.

Nous nous plaçons ici dans le cas où le nombre de disques n est égal à 3 et nous faisons appel à 4 prédicats

- $pos(d_i, p_j)$, qui indique que le disque d_i a pour position le piquet p_j .
- $sur(d_i, d_j)$, qui indique que le disque d_i est sur le disque d_j . Ce prédicat peut aussi correspondre à $sur(d_i, table)$ lorsque le disque d_i est le dernier.
- $vide(p_j)$ qui indique que le piquet p_j est vide.
- $sommet(d_i, p_j)$ qui indique que le disque d_i se trouve sur le sommet du piquet p_j .

Enfin, on fera appel au prédicat $sup(i, j)$ qui exprime que l'entier i est supérieur à l'entier j et au prédicat $diff(i, j)$ qui exprime que i est différent de j . Les faits de ces prédicats sont considérés comme prédéfinis.

1. *Etat initial* : Formuler l'état initial dans le formalisme STRIPS.
2. *Etat terminal* : Formuler l'état terminal toujours dans le formalisme STRIPS.
3. *Opérateurs* : Formuler trois opérateurs en STRIPS : le premier exprime que l'on déplace un disque vers un piquet vide, le second que l'on déplace un disque posé sur la table vers un piquet non vide et le dernier que l'on déplace un disque non posé sur la table vers un piquet non vide.
4. *Ébauche de résolution* : Simuler, en STRIPS, trois déplacements successifs de disques à partir de la situation initiale en indiquant, à chaque étape, l'évolution de la pile de buts et de l'état.
Remarque : on demande juste d'indiquer grossièrement les mouvements de disques et l'état de la pile de buts.

Exercice 2 Planification en ASP (Noté sur 7 points)

On reprend ici la programmation de la planification en ASP. Pour cela, nous rappelons qu'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 en ASP 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 bornés par une constante appelée fin .
Remarque : on choisira arbitrairement $fin = 7$.
2. *Représentation des états* : Décrire l'état initial avec les deux prédicats $pos(D, P, T)$ qui indique que le disque D se trouve sur le piquet P à l'instant T et $sur(D_i, D_j, T)$ qui précise que le disque D_i se trouve au dessus du disque D_j à l'instant T .
3. *Représentation du but* : Représenter le but à l'aide de contraintes d'intégrité, faisant intervenir la constante fin . 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 le piquet 3.
4. *Programmation de l'effet des mouvements* : À l'aide d'un prédicat $deplace(D, PP, P, T)$ qui indique qu'à la date T , le disque D situé au piquet PP est déplacé vers le piquet P , programmer en ASP les effets d'un tel mouvement.
5. *Spécification des contraintes* : Programmer en ASP les contraintes du mouvement, à savoir (i) qu'un disque ne peut se trouver sous un disque plus grand que lui et (ii) qu'un disque ne peut être déplacé s'il y a un disque au dessus.
6. *Inertie* : Programmer en ASP l'inertie des deux prédicats pos et sur lorsque les littéraux ne sont pas affectés par l'action.
7. *Règle de génération* : Programmer en ASP le choix d'un 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 Extraction de règles d'association (Noté sur 6 points)

Dans le tableau ci-dessous, les E_i correspondent à différentes transactions, autrement dit à des exemples d'apprentissage, tandis que A, B, C et $Classe$ représentent trois attributs qui possèdent respectivement 3, 3, 3 et 2 valeurs.

	A	B	C	Classe
E_1	a_1	b_1	c_3	-
E_2	a_1	b_3	c_1	+
E_3	a_2	b_2	c_1	-
E_4	a_3	b_1	c_3	+
E_5	a_3	b_2	c_3	-
E_6	a_3	b_1	c_2	-
E_7	a_3	b_3	c_2	-
E_8	a_1	b_3	c_3	+
E_9	a_1	b_3	c_2	+

Ici, nous faisons appel à l'algorithme *Apriori* pour engendrer tous les ensembles fréquents de l'ensemble d'apprentissage E donné dans le tableau avec $minsupp = 30\%$.

1. Donner la liste des items booléens de l'ensemble E , autrement dit, représenter l'ensemble E avec une ensemble d'attributs booléens sur lesquels on peut appliquer l'algorithme *Apriori*.
2. Rappeler les étapes de l'algorithme *Apriori*
3. Engendrer tous les ensembles fréquents à l'aide de l'algorithme *Apriori*. À chaque étape de l'algorithme, vous énoncerez l'ensemble des itemsets candidats et l'ensemble des itemsets fréquents.
4. Indiquer comment on extrait les règles d'association à partir d'un ensemble fréquent
5. À partir des ensembles fréquents de taille supérieure à 1 trouvés à la question précédente, engendrer toutes les règles d'association de classification, c'est-à-dire qui concluent sur la classe, dont la confiance est supérieure à 70% et le support supérieur à 30%.