

Examen Médian

Mercredi 4 Novembre 2009, de 8h à 10h, en salles B422 et A200, site de Belfort

Coefficient : 30%.**Aucun document autorisé.****Remarques et conseils :**

- Lisez **attentivement** chaque question avant d'y répondre.
- Indiquer clairement sur votre copie le numéro de l'exercice avant d'y répondre.
- Lorsque vous définissez un prédicat, son profil et sa définition formelle doivent être au moins indiqués. Un jeu d'essais est facultatif.
- Expliquez autant que possible vos choix lors de la définition d'un prédicat.
- Le barème défini ci-après est susceptible d'être modifié.

Les parties I et II devront être rédigées sur une même copie. La partie III devra être rédigée sur une copie séparée des parties I et II.

Partie I : Questions de cours (4 points)

1. Qu'est-ce que l'Intelligence Artificielle ? Donnez en au moins une définition.
2. Qu'est-ce qui caractérise un problème résolu de manière non déterministe ?
3. Dans un programme Prolog, à quoi sert l'opérateur de coupure ? Quelle différence y a-t'il entre une coupure verte et une coupure rouge ?

Partie II : Logique propositionnelle (4 points)

Soit la liste suivante d'énoncés :

- (E1) Les hauts fonctionnaires ne pratiquent pas le népotisme.
- (E2) Le népotisme n'existe pas.
- (E3) Si nous vivons dans une société contemporaine alors le népotisme existe.
- (E4) Dans une société démocratique il y a des hauts fonctionnaires.
- (E5) Nous vivons dans une société démocratique.

1. Utilisez le langage propositionnel pour traduire les énoncés de ce raisonnement.
2. Transformez les formules précédentes en clauses de Horn, si nécessaire.
3. Déterminez lequel des énoncés (E1), (E2), (E3), (E4) ou (E5) représente une conclusion qui rend le raisonnement composé des 4 autres énoncés et de cette conclusion valide.
Indice : utilisez la méthode de résolution sur chacun des 5 ensembles possibles d'hypothèses.

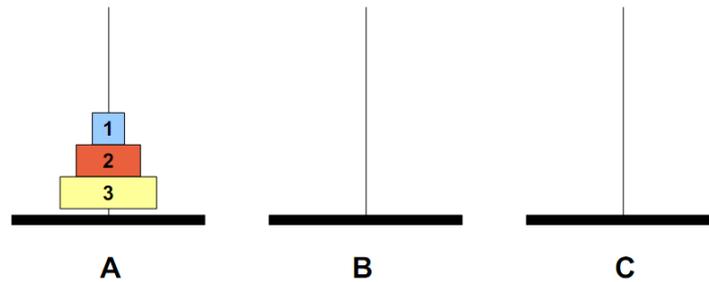
Partie III : Définition de prédicats Prolog (12 points)

Le problème des tours de Hanoï consiste à déplacer des disques de diamètres différents d'un piquet de départ vers un piquet d'arrivée en utilisant un piquet intermédiaire généralement situé au milieu des deux autres (figure 1). Les règles à respecter pour déplacer les disques sont les suivantes :

- on ne peut déplacer qu'un seul disque à la fois
- on ne peut placer un disque 1 sur un autre disque 2 que si le disque 1 a un diamètre plus petit que le disque 2.

On veut résoudre le problème des tours de Hanoï en Prolog, en considérant n disques, numérotés de

1 à n dans l'ordre croissant de leur diamètre. On propose de modéliser les disques empilés sur les 3 piquets par une liste de 3 éléments. Chaque élément sera un couple avec comme première valeur le nom du piquet (a, b ou c), et comme deuxième valeur, la liste des disques empilés sur ce piquet dans l'ordre croissant des disques. La liste représentant la situation initiale du problème (figure ci-dessous) sera donc la suivante : $L_i = [[a, [1,2,3]] , [b, []] , [c, []]]$.



- 1) Définir le prédicat «affichePile(+L)» qui affiche une liste d'entiers, qui seront séparés par des virgules à l'affichage. On utilisera pour cela le prédicat « write » qui permet d'afficher des entiers ou des chaînes de caractères entre quotes.

Exemple :

```
?- affichePile([1,2,3]).
1,2,3
```

- 2) Définir le prédicat «afficheHanoi(+L)» qui affiche l'état des tours de Hanoi représenté par L, en écrivant le nom du piquet, suivi de ':', suivi de la liste des disques empilés, du plus petit au plus grand.

Exemple :

```
?- afficheHanoi([ [a,[1,2,3]] , [b,[]] , [c,[]] ]).
a:1,2,3 - b: - c:
```

- 3) Définir le prédicat « depile(+L1,+T,-D,-L2) » qui modifie l'état du problème représenté par L1 en un état intermédiaire L2, en enlevant le disque D se trouvant au sommet du piquet T.

Exemple :

```
?- depile([ [a, [1,2,3]] , [b, []] , [c, []] ], a, D, L).
D = 1,
L = [ [a, [2, 3]] , [b, []] , [c, []] ].
```

- 4) Définir le prédicat « empile(+L1,+T,+D,-L2) » qui modifie l'état intermédiaire L1 en un état L2, en empilant le disque D sur le piquet T. On ne fera pas de test sur la faisabilité de cet empilement, car on supposera que le mouvement satisfait les contraintes des tours de Hanoi.

Exemple :

```
?- empile([ [a, [2, 3]] , [b, []] , [c, []] ], c, 1, L).
L = [ [a, [2, 3]] , [b, []] , [c, [1]] ].
```

- 5) Définir le prédicat « deplace(+L1,+T1,+T2,-L2) » qui modifie l'état L1 du problème en L2, en déplaçant le disque situé au sommet du piquet T1 sur le piquet T2.

Exemple :

```
?- deplace([ [a, [1,2,3]] , [b, []] , [c, []] ], a, c, L).
L = [ [a, [2, 3]] , [b, []] , [c, [1]] ].
```

- 6) Définir le prédicat « hanoi(N,X,Y,Z,L) » qui résout le problème des tours de hanoi pour N disques. X, Y, Z sont les piquets respectivement de départ, intermédiaire et d'arrivée. L est la liste représentant l'état initial du problème. A chaque déplacement de disque, on demande à ce que :

- un message écran indique quel disque est déplacé, ainsi que son piquet origine et le piquet

destination

- l'état du problème soit affiché de façon à visualiser les disques empilés sur chaque piquet.
Pour vous aider, l'algorithme récursif pour déplacer n disques de la tour a vers la tour c , en utilisant b comme pivot, est donné par :

1. déplacer les $n-1$ premiers disques de a vers b , en utilisant c comme pivot
2. déplacer le disque n de a vers c ,
3. déplacer les $n-1$ disques de b vers c , en utilisant a comme pivot

Exemple :

```
?- hanoi(3,a,b,c,[[a,[1,2,3]],[b,[]],[c,[]]]).  
deplacement disque 1 de a vers c : a:2,3 - b: - c:1  
deplacement disque 2 de a vers b : a:3 - b:2 - c:1  
deplacement disque 1 de c vers b : a:3 - b:1,2 - c:  
deplacement disque 3 de a vers c : a: - b:1,2 - c:3  
deplacement disque 1 de b vers a : a:1 - b:2 - c:3  
deplacement disque 2 de b vers c : a:1 - b: - c:2,3  
deplacement disque 1 de a vers c : a: - b: - c:1,2,3
```