

Examen médian IA01 A07
Université de Technologie de Compiègne

Durée : Deux Heures

Les documents ne sont pas autorisés

Tous ordinateurs, toutes communications sont interdits

Utilisez quatre copies séparées :

- *une copie pour la partie I*
- *une autre copie pour la partie II*
- *une autre copie pour la partie III*
- *une autre copie pour la partie IV*

I Qu'avez-vous retenu du cours (5 points) ?

1. A quoi sert la représentation des connaissances ? Donnez deux formalismes de représentation.
2. Donnez trois problèmes liés à l'exploitation des réseaux sémantiques.
3. Qu'est-ce qu'un frame ? Situez le par rapport à un réseau sémantique.
4. En quoi consistent les logiques de description ? Citez un langage appartenant à cette famille.
5. Expliquez la spécificité principale d'une programmation fonctionnelle. Citez un langage fonctionnel.

II Lisp et la représentation des connaissances (4 points)

1. Fonctions mystères (1,5 points)

Soient les deux fonctions suivantes :

```
(defun F1 (L)
  (cond
    ((atom L) 0)
    ((+ 1 (F1 (cdr L))) ) ) )
```

```
(defun F2 (N L)
  (cond
    ((atom L) nil)
    ((eq N 1) (car L))
    ((F2 (- N 1) (cdr L))) ) )
```

- Faites tourner les instructions suivantes : (0,5 point)
 - (F1 '(1 3 7 8 10)), (F1 nil)
 - (F2 4 '(1 4 6 10 7 6)), (F2 4 '(1 4))
- Expliquez ce que font les fonctions F1 et F2 (1 point).

2. Représentation des connaissances (2,5 points)

On cherche à gérer tous les étudiants d'IA01 (automne 2007). On considère les notes de TP (TP1, TP2, TP3), les notes de TD, les notes de Médian, les notes de Final. On souhaite, par exemple, retrouver la liste de tous les étudiants, la moyenne d'un examen (Médian ou Final), les étudiants qui ont plus d'une certaine note à l'examen, etc.

- Proposer une représentation de la promotion de IA01 de A07 (0,5 point).
- A partir de la représentation établie au 2.1, écrire :
 - une fonction lisp qui affiche tous les noms des étudiants d'IA01 de A07 (0,5 point),
 - une fonction lisp qui fait la moyenne du Médian (0,5 point),
 - une fonction lisp qui affiche les notes d'un étudiant donné (0,5 point),
 - une fonction lisp qui affiche tous les étudiants de IA01 qui ont plus de 15 au final et renvoie la liste de ces derniers (0,5 point).

III Système expert (5,5 points)

Je suis dans un restaurant bourguignon. Je voudrais, d'une part, découvrir le vignoble de Maranges, et d'autre part, que le vin ne soit pas servi trop froid (par exemple, 15°). Je me demande alors s'il est envisageable d'accompagner ce vin d'une viande en sauce locale...

Je vais donc utiliser un système expert ayant des connaissances en œnologie pour déterminer si mon choix de mets est pertinent au regard de mes exigences sur le vin.

Données du problème

Base de règles :

```
R1    [température >= 12] -> [couleur = rouge]
R2    [région = Alsace] ET [mets = gibier] -> [cépage = Pinot noir]
R3    [village = Sancerre] ET [garde >= 6] -> [couleur = rouge]
R4    [district = Côte de Beaune] -> [région = Bourgogne]
R5    [village = Sancerre] ET [couleur = rouge] -> [cépage = Pinot noir]
R6    [caractère = puissant] -> [mets = viande en sauce]
R7    [village = Maranges] -> [district = Côte de Beaune]
R8    [cépage = Pinot noir] ET [région = Bourgogne] -> [caractère = puissant]
R9    [couleur = rouge] ET [région = Bourgogne] -> [cépage = Pinot noir]
```

Base de faits : [village = Maranges], [température = 15]

But : [mets = viande en sauce]

Questions

- De quel ordre doit être un système expert capable d'exploiter ces connaissances ? Rappelez brièvement ce que signifie cette notion d'ordre, et illustrez par des exemples simples. (1 point)

- Dessinez l'arbre de recherche d'un moteur d'inférence en chaînage arrière tentant de résoudre ce problème. Précisez bien les règles qui sont applicables. (1 point)
- Proposez une représentation LISP pertinente de ces connaissances. (1 point)
- Exposez un algorithme effectuant un parcours en profondeur d'abord d'un tel arbre. (1,5 point) Vous utiliserez les fonctions de service suivantes :

termes-condition	: obtenir les prémisses d'une règle)
terme-conclusion	: obtenir la conclusion d'une règle)
connu?	: tester si une prémisse est vérifiée à l'aide de la base de faits
regles-candidates	: déterminer les règles concluant sur un but déterminé

- Proposez une implémentation LISP de ces quatre fonctions. (1 point)

IV Recherche dans un espace d'états (5,5 points)

On considère un jeu avec 6 pièces de monnaie identiques à deux faces (P : Pile et F : Face). Le joueur doit passer d'une configuration initiale (par exemple : PPPFPF) à une configuration finale (par exemple : FFFFFFFF). Les seules opérations à sa disposition pour cela consistent à changer les faces de deux pièces consécutives de son choix. Ainsi : FF devient PP, PP devient FF, FP devient PF et PF devient FP.

Dans un premier temps, on se propose de simuler la résolution de ce problème en utilisant une recherche dans un espace d'états.

Questions :

- Qu'est-ce qu'un état ? Comment représenter un état ? Dans l'exemple, quel est l'état initial ? quel est l'état final ? (1 point)
- Ecrire un algorithme permettant d'effectuer une recherche en profondeur d'abord. On pourra supposer que l'on dispose d'une fonction "successeurs" qui retourne la liste des états successeurs d'un état donné. (2 points)
- En s'appuyant sur l'exemple, dire pourquoi la recherche en profondeur n'est pas efficace. (0,5 point)
- En fait, ce type de problème peut être traité à l'aide d'une méthode dite « de gradient » qui consiste à choisir, à chaque étape, parmi les états successeurs, l'état « le plus proche » de l'état final. Il faut pour cela disposer d'une fonction d'évaluation de la proximité d'un état par rapport à un autre.
 - Définir une fonction d'évaluation pour le problème des pièces de monnaie et donner le code Lisp associé. (1 point)
 - Donner le principe général d'un algorithme mettant en œuvre une telle méthode. (1 point)