

Examen médian IA01 A08
Université de Technologie de Compiègne

Durée : 1H30

Les documents ne sont pas autorisés

Tous ordinateurs, toutes communications sont interdits

Utilisez trois copies séparées :

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

I Qu'avez-vous retenu du cours (6 points) ?

1. Donnez trois approches de l'IA en précisant leurs particularités.
2. Qu'est ce qu'un réseau sémantique ? Donnez deux problèmes liés à l'exploitation d'un tel réseau.
3. En quoi consiste une recherche dans un espace d'états ? Donnez la définition formelle d'un problème de recherche.
4. Donnez et expliquez les principales caractéristiques du langage Lisp.
5. Qu'est-ce qu'une a-liste ? Donnez un exemple. Qu'est-ce qu'une p-liste ? Comparez ces deux types de liste.
6. Dessinez la structure interne des listes suivantes :
(A B C) ((A . B) . C) ((A B) . C) (A B . C)

II Programmation Lisp (7 points)

1. Que font les fonctions suivantes ? Donnez des exemples simples d'exécution de ces fonctions avec les résultats correspondants. Existe-t-il des primitives équivalentes en Lisp (3 points) ?

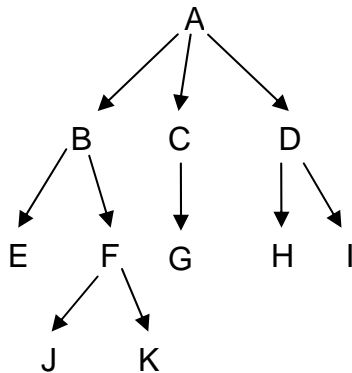
```
(defun F1 (A L)
  (cond
    ((atom L) nil)
    ((eq (car L) A) L)
    ((F1 A (cdr L))) ) )
```

```
(defun F2 (L) (if L (cons (car L) (F2 (cddr L)))) )
```

```
(defun F3 (L)
  (cond
    ((null L) 0)
    ((atom L) 1)
    ((+ (F3 (car L)) (F3 (cdr L))) ) ) )
```

2. On considère un arbre représentant une classification.

Exemple :



On choisit de représenter les arbres de ce type de la façon suivante :

- chaque nœud est représenté par son nom suivi du nom de ses fils,
- on ne représente pas les feuilles.

Ainsi, avec l'exemple, on obtient :

((A B C D)(B E F)(C G)(D H I)(F J K))

2.1 Ecrire une fonction Lisp **descendant?** permettant de tester si un élément est un descendant d'un autre. (2 points)

Exemples :

```
> (descendant? 'J 'B)
> t

> (descendant? 'J 'C)
> nil

> (descendant? 'F 'A)
> t

> (descendant? 'J 'K)
> nil
```

2.2 Proposer une amélioration de la représentation et écrire la fonction **descendant2?** correspondante. (1 point)

2.3 Ecrire une fonction **ascendants** qui retourne tous les ascendants d'un élément. Utiliser pour cela l'une ou l'autre des deux représentations, en précisant la représentation choisie (1 point).

Exemples :

```
> (ascendants 'J)
> (F B A)
> (ascendants 'A)
> nil
```

III Système expert (7 points)

Fervent amateur du compositeur baroque François Couperin, j'aimerais savoir s'il me serait loisible de jouer ses pièces avec un instrument de musique ressemblant à une grande caisse en bois pesant environ 69 kilogrammes, et muni d'un mécanisme à base de sautereaux... Pour m'éclairer, je vais faire appel à un système expert spécialisé en organologie (qui est la science de la classification des instruments de musique).

Données du problème

Base de règles

R1	[catégorie = aérophones] ET [résonateur = tuyaux] -> [nombreDeClaviers >= 1]
R2	[nombreDePédales = 7] ET [nombreDeClaviers = 0] -> [famille = harpes]
R3	[famille = harpes] -> [catégorie = cordesPincées]
R4	[résonateur = caisse] ET [mécanismes = sautereaux] ET [poids >= 30] -> [famille = citharesAPlanche]
R5	[catégorie = cordesPincées] ET [nombreDeClaviers >= 1] -> [instrument = clavecin]
R6	[instrument = clavecin] -> [jouerCouperin = possible]
R7	[famille = citharesAPlanche] -> [catégorie = cordesPincées]
R8	[poids >= 30] ET [résonateur = caisse] -> [nombreDeClaviers >= 1]

Base de faits

[poids = 69], [résonateur = caisse], [mécanismes = sautereaux]

But

[jouerCouperin = possible]

Questions

1. Précisez l'ordre dont doit être un système expert capable d'exploiter ces connaissances. Vous rappellerez alors la signification de cette notion d'ordre, et vous illustrerez vos propos par des exemples simples pour chacun des cas. (2 points)
2. Dessinez l'arbre de recherche qui matérialiserait le cheminement d'un moteur d'inférence en chaînage arrière à qui l'on soumettrait ce problème. Prenez bien soin de préciser les règles applicables. (1 point)
3. Exposez une représentation LISP pertinente de l'ensemble des connaissances constituant les données du problème. (1 point)
4. Proposez un algorithme (*clairement présenté et écrit en français*) effectuant le parcours en profondeur d'abord de l'arbre que vous avez tracé dans la question 2. Vous devrez utiliser les fonctions de service suivantes :

termes-condition

(obtenir les prémisses d'une règle)

terme-conclusion

(obtenir la conclusion d'une règle)

connu?

(savoir si un fait est dans la base de faits)

regles-candidates

(déterminer les règles concluant sur un but déterminé)

Note : cet algorithme ne questionnera pas l'utilisateur. (1,5 points)

5. Ecrivez désormais une implémentation LISP de chacune de ces quatre fonctions. (1,5 points)