

Examen médian IA01 A06
Université de Technologie de Compiègne

Durée : Deux Heures

Les documents ne sont pas autorisés

Tous ordinateurs, toutes communications sont interdits

Utilisez trois copies séparées :

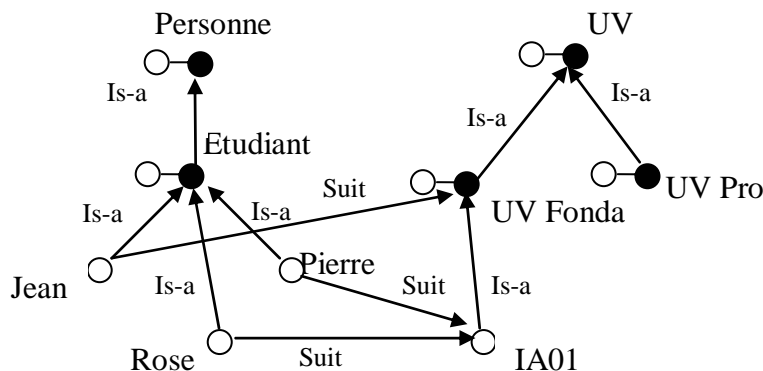
- une copie pour la partie I
- une autre copie pour la partie II
- une autre copie pour la partie III

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

1. Qu'est-ce qu'un réseau sémantique ?
2. Donnez trois problèmes liés à l'exploitation des réseaux sémantiques.
3. Qu'est-ce qu'un frame ? Faire une comparaison par rapport à un réseau sémantique.
4. Qu'est-ce qu'un système expert ? Donnez ses composants, la signification de ses différents ordres et différents chaînages.
5. Qu'est-ce qu'un langage fonctionnel ?

II NETL (7 points)

Soit le réseau suivant :



1. Connaissances représentées (1 point)

Quelles connaissances représentent le réseau précédent ?

2. Notion de rôle (2 points)

- a. Représentez les étudiants de l'Université de Compiègne au moyen des concepts Compiègne, université et étudiants. (1 point)
- b. Représentez : Jean, Rose et Pierre sont des étudiants de l'UTC. (1 point)

3. Processus d'interrogation (3 points)

Précisez le processus de marquage permettant de répondre aux questions :

a. Quels sont les étudiants suivant des UVs fonda ? (2 points)

b. Quels sont les étudiants de l'UTC ? (1 point)

4. Détection d'incohérence (1 point)

Complétez le réseau précédent afin de préciser qu'une UV ne peut être à la fois pro et fonda

III Système expert (8 points)

Dans ce qui suit, on va représenter des connaissances sur les animaux d'un zoo. L'idée est de réaliser un système expert permettant de reconnaître des animaux.

1. Base de connaissances (3 points)

Le système expert va avoir des règles du style

```
Si l'animal a des sabots, alors c'est un ongulé.  
Si l'animal est un ongulé, a une couleur noire et blanche et des  
rayures, alors c'est un zèbre
```

Les règles seront mises sous forme d'une liste qui sera la valeur d'un symbole identifiant la règle. Cette liste sera de la forme :

```
(SI <premise>+ ALORS <consequence>+)
```

où + indique la présence d'un élément ou plus. Si plusieurs éléments sont présents, ils correspondent à un ET logique.

Exemple:

```
? R1  
(SI (ROBE A-COULEUR NOIR-BLANC) (FAMILLE EST ONGULES) (DESSIN-ROBE  
EST RAYURES)  
ALORS (TYPE EST ZEBRE) (REGIME-ALIMENTAIRE EST HERBIVORE))
```

La base de faits sera réalisée sous forme d'une liste de triplets (<nom> <predicat> <valeur>) affectée à la variable globale *faits*.

```
(defParameter *faits* '((ROBE A-COULEUR BLANC) (FAMILLE EST  
ONGULES) (DESSIN-ROBE EST RAYURES)))
```

a. Écrire une fonction make-rule qui crée des règles à partir d'une liste de prémisses et de conséquences de la forme

```
? (make-rule '((COU EST LONG) (ROBE EST TACHETEE))'((TYPE EST GIRAFE))  
Rxx
```

où xx est un numéro affecté par le système (on pourra pour cela utiliser la primitive gentemp qui retourne un nouvel atome à chaque appel).

La fonction mettra également le nom de la règle dans une liste affectée à la variable globale *regles*

b. Écrire une fonction `premisses` permettant d'extraire la liste des prémisses d'une règle à partir du nom de la règle.

```
? (premisses 'R1)
((ROBE A-COULEUR NOIR-BLANC) (FAMILLE EST ONGULES) (DESSIN-ROBE EST
RAYURES))
```

c. Écrire une fonction `consequences` permettant d'extraire la liste des conséquences à partir du nom de la règle.

```
? (consequences 'R1)
((TYPE EST ZEBRE) (REGIME-ALIMENTAIRE EST HERBIVORE))
```

2. Moteur d'inférence (5 points)

On se propose maintenant d'écrire un moteur d'inférence qui fonctionne en chaînage arrière et en profondeur d'abord. Il utilisera les fonctions et les variables globales définies précédemment.

a. Écrire une fonction `regles-candidates` qui retourne la liste des règles concluant sur un but donné.

```
? (setq but '(TYPE EST ZEBRE))
? (regles-candidates but)
(R1 ... Rx ....)
```

b. Écrire une fonction `appartient` permettant de déterminer si un but est présent ou non dans la base de faits.

c. Écrire **en français** l'algorithme correspondant au moteur d'inférence. Le système ne posera pas de questions à l'utilisateur.