Université d'ELOUED

Faculté de sciences exactes

Département d'Informatique — *3 Licence Informatique*

Durée 01h — *Date 24/05/2022*

Contrôle du module Intelligence Artificielle

Questions (5 pts)

1. Donner deux exemples d'applications de l'intelligence artificielle

- Jeux, Systèmes experts, Traitement du langage naturel, système de vision artificielle, reconnaissance de la parole …..

2. Citer les participants aux développement d'un système expert
   - Expert du domaine
   - Ingénieur de connaissances (cogniticien)
   - Programmeur
   - Directeur de projet
3. Donner le cycle d'un moteur d'inférence.

Saisie des faits initiaux

Début

Phase de filtrage => Détermination des règles applicables.

Tant que ensemble de règles applicables n'est pas vide ET que le pb n'est pas résolu Faire

Phase de choix => Résolution des conflits

Appliquer la règle choisie (exécution)

Modifier (éventuellement) l'ensemble des règles applicables

Fin Faire

Fin

Exercice 1

Soit la base de règle suivante :

R1 : **SI** Responsabilité **ET** Langue-facile **ET** Néerlandais-parlé **ALORS** Dynamique
R2 : **SI** Langue-facile **ET** Anglais-parlé **ALORS** Adaptabilité
R3 : **SI** Slave **ET** Dynamique **ALORS** Adaptabilité
R4 : **SI** Responsabilité **ALORS** Leadership
R5 : **SI** Langue-facile **ALORS** Néerlandais-parlé
R6 : **SI** Adaptabilité **ET** Leadership **ALORS** Accepté
R7 : **SI** Slave **ALORS** Langue-facile
R8 : **SI** Leadership **ET** Slave **ALORS** Adaptabilité

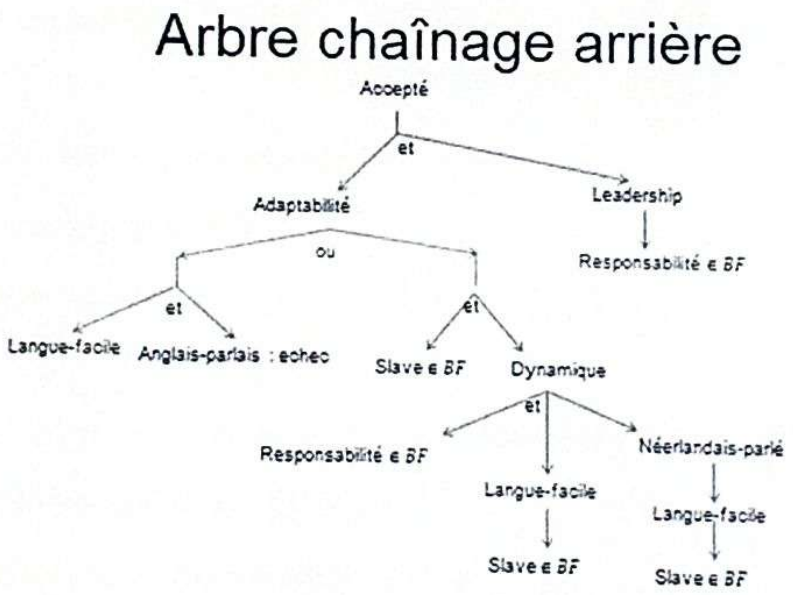
Base de faits initiale : Slave, Responsabilité

1. Prouver le fait « Accepté » par Chaînage avant en largeur d'abord **(4pts)**

Conflits	Base de fait
R4, R7	Slave, Responsabilité, Leadership, Langue-facile
R5, R8	Slave, Responsabilité, Leadership, Langue-facile, Néerlandais-parlé, Adaptabilité
R1, R6	Slave, Responsabilité, Leadership, Langue-facile, Néerlandais-parlé, Adaptabilité, Dynamique, Accepté

2. Utiliser l'arbre Chaînage arrière pour vérifier le même but « Accepté » **(6pts)**

Pour la résolution de conflits utiliser la stratégie qui préfère la règle ayant le plus petit indice.



Exercice 2 (5pts)

Le codage d'une règle est le suivant: regle = [['cond1','cond2',...,'condn'], consequence]
Ecrire les sous-programmes python qui permettent de :

1. Retourner les conditions d'une règle.
2. Retourner la conséquence d'une règle
3. Indiquer si un fait passé en paramètre satisfait une des conditions d'une règle ou non

1. Retourne les conditions d'une regle.

```
def conditionsRegle( regle ):
    return regle[0]
```

2. ## Retourne la consequence d'une regle

```
def consequenceRegle( regle ):
    return regle[1]
```

3. ## Indique si un fait passe en parametre satisfait une des conditions d'une regle ou non

```
def satisfaitUneCondition( regle, le_fait ):
    return ( le_fait in conditionsRegle(regle) )
```

Bon courage