

Projet LO21 – Explication détaillée de l'énoncé

Ce document résume clairement ce que vous devez coder pour chaque question.

1. Type abstrait REGLE

Vous devez définir un TAD (type abstrait de données) pour représenter une règle d'un système expert.

Une règle contient :

- Une prémisse : liste de propositions (A, B, C...)
- Une conclusion : une proposition unique

Vous devez coder les opérations suivantes :

1. Créer une règle vide
2. Ajouter une proposition à la prémisse (en queue)
3. Créer/définir la conclusion
4. Tester si une proposition appartient à la prémisse (récursif)
5. Supprimer une proposition de la prémisse
6. Tester si la prémisse est vide
7. Accéder à la première proposition de la prémisse
8. Accéder à la conclusion

Chaque opération doit être écrite :

- avec sa signature (interface)
- avec un algorithme utilisant uniquement les opérations du TAD liste vues en cours

2. Type abstrait BC (Base de Connaissances)

Une base de connaissances est une liste de règles.

Vous devez coder :

1. Créer une base vide
2. Ajouter une règle à la base (en queue)
3. Accéder à la règle en tête de la base

Pour chacune :

- algorithme
- sous-programme C

3. Moteur d'inférence

Vous devez coder un moteur capable de :

- Prendre la base de faits (propositions vraies au départ)
- Parcourir les règles
- Vérifier si la prémisse d'une règle est vraie grâce aux faits
- Si oui : ajouter la conclusion à la base de faits
- Continuer tant que des faits nouveaux sont déduits

Vous devez fournir :

- L'algorithme complet
- Le programme C correspondant
- L'utilisation obligatoire des TAD Regle et BC

4. Rapport

Le rapport doit contenir :

- Choix de conception des structures de données
- Algorithmes (pas de C dans cette partie)
- Explications
- Jeux d'essais
- Commentaires sur les résultats

Interdictions :

- Ne pas mettre le code C dans le rapport
- Pseudocode ou C \neq algorithmes (il faut vrai algorithme avec lexique)