

Projet LO21

A faire seul ou par groupe de deux étudiants (pas plus)

Un système expert est composé d'une base de connaissances (une expertise sur un domaine donné), une base de faits (des faits certains concernant un domaine donné) et un moteur d'inférence (un système permettant de déduire de nouveaux faits certains à partir de la base de connaissances et de la base de faits).

Dans ce projet, nous proposons de réaliser un mini système expert, dans lequel, les connaissances sont des propositions logiques pouvant avoir la valeur « vrai » ou « faux ».

1. Définition des composantes d'un système expert

- Une **base de connaissances** est une liste de règles.
- Une **règle** est composée de deux parties :
 - une partie prémisses contient des propositions reliées par l'opérateur logique « et »
 - une partie conclusion contient une seule proposition.

Exemple d'une règle : A et B et C => D

Dans cette règle, la prémisse est : A et B et C, la conclusion est D.

Une règle se lit de la manière suivante : si la prémisse est vraie alors on déduit que la conclusion est vraie.

On considère une règle comme une liste de propositions, dont le dernier élément est la conclusion de la règle.

- Une **base de faits** est une liste de propositions considérées comme vraies.

Base de connaissances :

R1. ¬ réservoirVide et pharesFonctionnent et ¬ moteurDemarre => problemeBougie

R2. ¬ moteurDemarre et ¬ pharesFonctionnent => problemeBatterie

R3. ¬ moteurDemarre et pharesFonctionnent => problemeStarter

Base de faits : ¬moteurDemarre, pharesFonctionnent

2. Questions

1. Définir un type abstrait de données appelé **Regle** permettant de définir les règles d'une base de connaissances. Le type abstrait Regle est muni des opérations suivantes :
 - Créer une règle vide,
 - Ajouter une proposition à la prémisse d'une règle, l'ajout se fait en queue
 - Créer la conclusion d'une règle

- Tester si une proposition appartient à la prémisse d'une règle, **de manière récursive**
- Supprimer une proposition donnée de la prémisse d'une règle
- Tester si la prémisse d'une règle est vide
- Accéder à la proposition se trouvant en tête d'une prémisse
- Accéder à la conclusion d'une règle

Pour chacune de ces opérations, donner sa signature (son interface) et son algorithme en utilisant les opérations proposées par le type abstrait liste vu en cours. L'algorithme de l'opération qui teste l'appartenance d'une proposition à une règle doit être récursive.

2. Définir un type abstrait **BC** permettant de définir des Bases de Connaissances comme une liste de règles. Ecrire l'algorithme et le sous-programme C correspondant aux opérations suivantes :
 - Créer une base vide
 - Ajouter une règle à une base, l'ajout peut se faire en queue
 - Accéder à la règle se trouvant en tête de la base
3. Ecrire l'algorithme et le programme C réalisant le moteur d'inférence (comme expliqué en cours). Un moteur d'inférence permet de déduire des faits certains (vrais) en partant de la base de faits et en appliquant les règles de la base de connaissance. L'algorithme et le programme doivent utiliser les opérations des 2 questions précédentes et éventuellement, les autres opérations prédéfinies sur le type abstrait liste étudié en cours.

3. Rapport

Le rapport devra contenir les éléments suivants :

- Description des choix de conception et d'implémentation relatifs aux structures de données utilisées et à la démarche adoptée.
- Algorithmes des sous programmes (en utilisant des opérations abstraites) et leurs explications.
- Jeux d'essais.
- Commentaires sur les résultats.

Le rapport ne devra pas contenir de listing du programme C.

4. Remarques

- Un programme C et du pseudocode ne sont pas des algorithmes.
- Un algorithme nécessite de donner son profil (données, résultat, lexique).
- Un programme doit être commenté (de façon utile).
- La qualité d'implémentation est prise en compte.

Rapport à rendre et démonstration à réaliser sur ordinateur devant l'enseignant.