

IFT-2008 – Algorithmes et structures de données ÉTÉ 2017

Travail pratique 1 (individuel)

À remettre, par le portail du cours (ENA) avant 17h00 le mercredi 07 juin 2017

1 Mise en contexte

Nous vous demandons dans ce premier travail pratique d'implanter un système expert en utilisant les notions que nous avons vues dans le cours concernant le type abstrait « liste ordonnées ».

Un système expert est un logiciel fournissant dans un domaine particulier les mêmes conclusions qu'un être humain expert dans ce domaine. Par exemple, il peut fournir un diagnostic médical à partir d'un ensemble de symptômes, classer des espèces animales ou végétales à partir d'observations, etc.

En effet, un système expert doit être capable :

- D'enregistrer les faits initiaux (les symptômes d'un malade, les observations sur l'animal en cours d'examen, etc.). On parle alors d'ensemble de faits.
- D'enregistrer des règles composées de prémisses et de conclusions.
- D'appliquer ces règles pour en déduire de nouveaux faits non connus initialement.

La construction d'un système expert consiste dans l'intégration de l'ensemble des faits avec l'ensemble des règles et de l'implantation des moteurs d'inférence, algorithmes pour vérifier la véracité de faits et de déduction de nouveaux faits.

Nous commençons par vous donner un exemple illustrant l'utilisation d'un système expert. Il est à noter que cet exemple n'est pas utilisé pour expliquer les systèmes experts en détail. Les règles mentionnées ci-dessous sont données à titre indicatif, sans prétention quant au domaine de l'expert, pour illustrer le comportement du système.

Voici un exemple de base de règles (numérotés par Ri) que pourrait avoir un système expert pour classer des espèces animales (Les règles sont de la forme Si « prémisses », alors « conclusions ») :

- R1. Si Animal « a des poils », alors Animal « est un mammifère »
- R2. Si Animal « nourrit ses petits au lait », alors Animal « est un mammifère ».
- R3. Si Animal « a des plumes », alors Animal « est un oiseau »
- R4. Si Animal « vole » et « pond des œufs », alors Animal « est un oiseau »
- R5. Si Animal « est un mammifère » et « mange de la viande », alors Animal « est un carnivore »
- R6. Si Animal « est un mammifère » et « a les yeux dirigés en avant » et « a des dents pointues » et « a des griffes », alors Animal « est un carnivore »

- R7. Si Animal « est un mammifère » et « a des ongles », alors Animal « est un ongulé »
- R8. Si Animal « est un carnivore » et « est brun » et « a des taches noires », alors Animal « est un guépard »
- R9. Si Animal « est un carnivore » et « est brun » et « a des raies noires », alors Animal « est un tigre »
- R10. Si Animal « est un ongulé » et « a un long cou » et « est brun » et « a des taches noires », alors Animal « est une girafe »
- R11. Si Animal « est un ongulé » et « est blanc » et « a des raies noires », alors Animal « est un zèbre »
- R12. Si Animal « est un oiseau » et « ne vole pas », alors Animal « est un oiseau non volant »
- R13. Si Animal « est un oiseau » et « nage », alors Animal « est un oiseau non volant »
- R14. Si Animal « est un oiseau non volant » et « a des pattes longues » et « a un long cou » et « est noir et blanc », alors Animal « est une autruche »
- R15. Si Animal « est un oiseau non volant » et « a des pattes palmées » et « est noir et blanc », alors Animal « est un pingouin »
- R16. Si Animal « est un oiseau » et « vole remarquablement », alors Animal « est un albatros »

Voici maintenant un exemple de base de faits (F1) que notre système expert pourrait utiliser :

- F1-1. Ficelle « a des poils »
- F1-2. Ficelle « a des ongles »
- F1-3. Ficelle « a un long cou »
- F1-4. Ficelle « est brun »
- F1-5. Ficelle « a des taches noires »

L'algorithme de déduction de nouveaux faits à partir des règles et des faits connus est dit un moteur d'inférence par « chaînage avant ». Le principe est de saturer la base de faits que nous avons, c'est-à-dire d'arriver à avoir une base de faits de laquelle nous ne pouvons plus déduire d'autres faits avec les règles connues. Le processus d'inférence est expliqué dans l'algorithme ci-dessous. Prenons par exemple la base de faits F1 constituée de 5 faits. Un animal nommé Ficelle est couvert de poils, a des ongles, a un long cou, est de couleur brune et a des taches noires.

- R1: Ficelle « a des poils » → Ficelle « est un mammifère »
Le fait Ficelle « est un mammifère » est ajouté à F1 comme fait numéro F1-6.
- R7: Ficelle « est un mammifère » et « a des ongles » → Ficelle « est un ongulé »
Le fait Ficelle « est un ongulé » est ajouté à la base de faits F1.
- R10: Ficelle « est un ongulé » et « a un long cou » et « est brun » et « a des taches noires » →
Ficelle « est une girafe »
Le fait Ficelle « est une girafe » est ajouté à la base de faits F1.

Comme vous pouvez le constater, nous avons déduit (inférer) que Ficelle est un mammifère, est un ongulé et est une girafe. Ces trois faits s'ajoutent à la base de 5 faits, que nous avions au départ. Nous ne pouvons plus déduire d'autres nouveaux faits, car aucune autre des prémisses des règles n'est satisfaite. Nous avons alors saturé la base de faits et nous pouvons même affirmer que nous avons classé Ficelle qui est bien sûr une girafe.

Voici donc de façon formelle l'algorithme utilisé :

Jusqu'à ce qu'aucune règle ne produise de nouvelles assertions :

```
{
    Pour chaque règle
    {
        Essayer de vérifier chacune des prémisses de la règle en essayant de l'assortir aux faits connus.

        Si toutes les prémisses de la règle sont vérifiées, faire de la conclusion une nouvelle assertion (un nouveau fait) à moins qu'une assertion identique n'existe déjà.
    }
}
```

Votre mandat est d'implanter un tel système expert qui peut déduire de nouveaux faits en appliquant les règles aux faits observés par chaînage avant.

2 Modélisation du système expert

L'ensemble des classes que vous devez utiliser pour réaliser ce travail pratique sont définies dans les fichiers que nous vous fournissons :

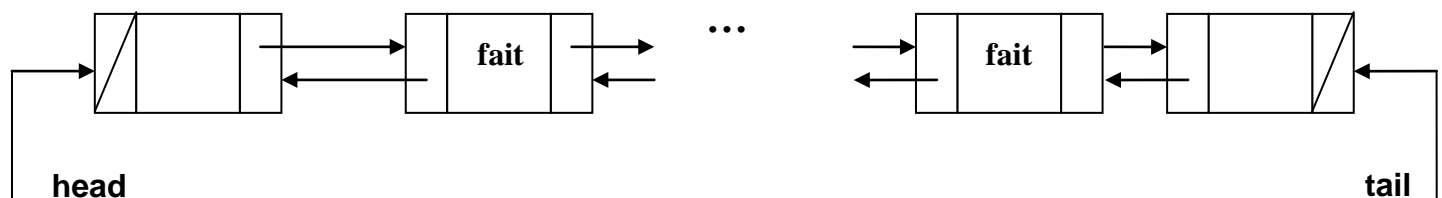
- ListeCirculaire.h et ListeCirculaire.hpp;
- Regle.h et Regle.cpp;
- Principal.cpp;
- SystemeExpert.h et SystemeExpert.cpp;
- AssertionException.h

Il est interdit de modifier les fichiers .h fournis à deux exceptions près. Vous pouvez éventuellement enrichir les commentaires d'interface en format Doxygen dans ces fichiers. De plus, il est permis d'ajouter des méthodes privées dans les classes pour simplifier votre développement. **Ne touchez pas s.v.p. aux variables privées afin de simplifier la correction !**

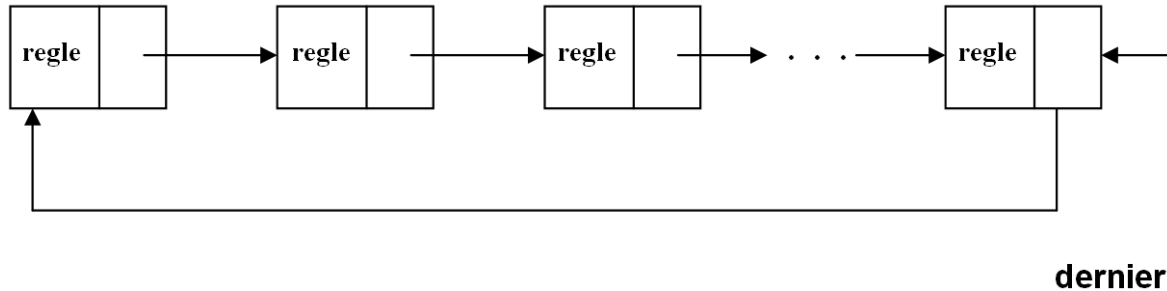
PS. Le fichier AssertionException.h est une version simplifiée de ContratException. Vous pouvez vérifier les préconditions et les postconditions de vos méthodes en ajoutant simplement des assertions (voir les exemples dans le fichier ListeCirculaire.hpp). Vous n'avez donc pas à gérer les invariants. De plus, vous n'êtes pas obligé d'ajouter des « m_ » aux données membres des classes, des « p_ » aux paramètres des méthodes, des « req » aux accesseurs ou des « asg » aux modificateurs.

Un système expert comprend les attributs suivants :

- **baseFaits:** une liste doublement chaînée utilisant le conteneur list de la STL. Le nœud typique contient un « Typefait » qui est une chaîne de caractère représentant la description d'un fait.



- **baseRegles**: une liste circulaire dont chacun des éléments est une règle. Cette dernière contient un premier ensemble de faits représentant les prémisses de la règle. Elle contient également un deuxième ensemble de faits représentant les conclusions de la règle. Les prémisses et les conclusions utilisent également le conteneur list de la STL.



3 Travail à faire

Il s'agit essentiellement d'implémenter dans les fichiers `Regle.cpp` et `SystemeExpert.cpp` toutes les méthodes des classes définies dans les fichiers `.h`. Vous allez aussi devoir ajouter ou enrichir les commentaires d'interface et d'implémentation en format Doxygen dans les divers fichiers `.h`, `.cpp` et `.hpp`. Ne les oubliez pas !

Nous vous fournissons un fichier texte. Le fichier `animal.txt` que pouvez utiliser pour effectuer vos tests en chargeant en mémoire le système expert par l'intermédiaire de la fonction « chargerSE » que vous devez implanter.

Le format du fichier texte est le suivant :

Regle_1_Premisse_1		
Regle_1_Premisse_2		
...		
Regle_1_Premisse_N		PREMIÈRE
!>		RÈGLE
Regle_1_Conclusion_1		
Regle_1_Conclusion_2		
...		
Regle_1_Conclusion_N		
!%		
Regle_2_Premisse_1		
...		
Regle_2_Premisse_N		DEUXIÈME
!>		RÈGLE
Regle_2_Conclusion_1		
...		
Regle_2_Conclusion_N		
!%		
... ..		
!%		
Regle_N_Premisse_1		
...		

Regle_N_Premisse_N	N ième
!>	RÈGLE
Regle_N_Conclusion_1	
...	
Regle_N_Conclusion_N	
!!	
Fait_1	
Fait_2	BASE DE
...	FAITS
Fait_3	

Nous vous fournissons également un fichier Principal.cpp comprenant un main() qui vous permettra de vous aider à tester votre système expert. Nous avons déjà mis quelques exemples dans le main, mais il en faut normalement d'autres. Par ailleurs, voici ce que le fichier principal.cpp est censé donner comme résultat dans la console de la VM :

```

----- Menu -----
1 - Ajouter une regle au systeme expert.
2 - Ajouter un fait a analyser au systeme expert.
3 - Supprimer une regle du systeme expert.
4 - Supprimer un fait du systeme expert.
5 - Charger un systeme expert depuis un fichier texte.
6 - Tirer des conclusions a partir de la base de faits.
7 - Copier un systeme expert dans un autre
8- Sauvegarder un systeme expert dans un fichier texte
0 - Quitter.
-----

```

```

Entrer s.v.p. votre choix (0 a 8):? 1
La regle suivante a ete ajoutez:
Si Animal << a des poils>>, alors Animal <<est un mammifere>>
Nombre de premisses de cette regle: 1
Nombre de conclusions de cette regle: 1
Affichage du systeme expert:
Base de regles:
Element numero 1:
Premisses:
a des poils
Conclusions:
est un mammifere
Base de fait:
La liste est vide

```

```

Entrer s.v.p. votre choix (0 a 8):? 2
Le fait suivant a ete ajoute: a des poils
Nombre de faits dans la base de faits: 1
Affichage du systeme expert:
Base de regles:
Element numero 1:
Premisses:
a des poils

```

Conclusions:
est un mammifere
Base de fait:
a des poils

Entrer s.v.p. votre choix (0 a 8):? 3
La regle suivante a ete supprimee:
Si Animal <<a des poils>>, alors Animal <<est un mammifere>>
Affichage du systeme expert:
Base de regles:
La liste est vide
Base de fait:
a des poils

Entrer s.v.p. votre choix (0 a 8):? 4
Le fait suivant a ete supprime: a des poils
Affichage du systeme expert:
Base de regles:
La liste est vide
Base de fait:
La liste est vide

Entrer s.v.p. votre choix (0 a 8):? 5
Le systeme expert a ete charge, voici son contenu:
Base de regles:
Element numero 1:
Premisses:
a des poils
Conclusions:
est un mammifere
Element numero 2:
Premisses:
nourrit ses petits au lait
Conclusions:
est un mammifere
Element numero 3:
Premisses:
a des plumes
Conclusions:
est un oiseau
Element numero 4:
Premisses:
vole
pond des oeufs
Conclusions:
est un oiseau
Element numero 5:
Premisses:
est un mammifere
mange de la viande

Conclusions:

est un carnivore

Element numero 6:

Premises:

est un mammifere

a les yeux diriges en avant

a des dents pointues

a des griffes

Conclusions:

est un carnivore

Element numero 7:

Premises:

est un mammifere

a des ongles

Conclusions:

est un ongule

Element numero 8:

Premises:

est un carnivore

est brun

a des taches noires

Conclusions:

est un guepard

Element numero 9:

Premises:

est un carnivore

est brun

a des raies noires

Conclusions:

est un tigre

Element numero 10:

Premises:

est un ongule

a un long cou

est brun

a des taches noires

Conclusions:

est une girafe

Element numero 11:

Premises:

est un ongule

est blanc

a des raies noires

Conclusions:

est un zebre

Element numero 12:

Premises:

est un oiseau

ne vole pas

Conclusions:

est un oiseau non volant

Element numero 13:

Premises:

est un oiseau

nage

Conclusions:

est un oiseau non volant

Element numero 14:

Premises:

est un oiseau non volant

a des pattes longues

a un long cou

est noir et blanc

Conclusions:

est une autruche

Element numero 15:

Premises:

est un oiseau non volant

a des pattes palmees

est noir et blanc

Conclusions:

est un pingouin

Element numero 16:

Premises:

est un oiseau

vole remarquablement

Conclusions:

est un albatros

Base de fait:

a des poils

a des dents pointues

a des griffes

a les yeux diriges en avant

est brun

a des taches noires

Nombre de regles dans la base de regles: 16

Nombre de faits dans la base de faits: 6

Entrer s.v.p. votre choix (0 a 8):? 6

Le systeme expert a ete sature

Voici la nouvelle base de faits:

a des poils

a des dents pointues

a des griffes

a les yeux diriges en avant

est brun

a des taches noires

est un mammifere

est un carnivore

est un guepard

Voici la liste des regles utilises:

Element numero 1:

Premises:

a des poils

Conclusions:

est un mammifere

Element numero 2:

Premises:

est un mammifere

a les yeux diriges en avant

a des dents pointues

a des griffes

Conclusions:

est un carnivore

Element numero 3:

Premises:

est un carnivore

est brun

a des taches noires

Conclusions:

est un guepard

Nombre de regles utilises: 3

Entrer s.v.p. votre choix (0 a 8):? 7

Affichage du systeme expert avant la copie:

Base de regles:

La liste est vide

Base de fait:

La liste est vide

Affichage du systeme expert apres la copie:

Base de regles:

Element numero 1:

Premises:

a des poils

Conclusions:

est un mammifere

Element numero 2:

Premises:

nourrit ses petits au lait

Conclusions:

est un mammifere

Element numero 3:

Premises:

a des plumes

Conclusions:

est un oiseau

Element numero 4:

Premisses:

vole

pond des oeufs

Conclusions:

est un oiseau

Element numero 5:

Premisses:

est un mammifere

mange de la viande

Conclusions:

est un carnivore

Element numero 6:

Premisses:

est un mammifere

a les yeux diriges en avant

a des dents pointues

a des griffes

Conclusions:

est un carnivore

Element numero 7:

Premisses:

est un mammifere

a des ongles

Conclusions:

est un ongule

Element numero 8:

Premisses:

est un carnivore

est brun

a des taches noires

Conclusions:

est un guepard

Element numero 9:

Premisses:

est un carnivore

est brun

a des raies noires

Conclusions:

est un tigre

Element numero 10:

Premisses:

est un ongule

a un long cou

est brun

a des taches noires

Conclusions:

est une girafe

Element numero 11:

Premisses:

est un ongule

est blanc
a des raies noires
Conclusions:
est un zebre
Element numero 12:
Premisses:
est un oiseau
ne vole pas
Conclusions:
est un oiseau non volant
Element numero 13:
Premisses:
est un oiseau
nage
Conclusions:
est un oiseau non volant
Element numero 14:
Premisses:
est un oiseau non volant
a des pattes longues
a un long cou
est noir et blanc
Conclusions:
est une autruche
Element numero 15:
Premisses:
est un oiseau non volant
a des pattes palmées
est noir et blanc
Conclusions:
est un pingouin
Element numero 16:
Premisses:
est un oiseau
vole remarquablement
Conclusions:
est un albatros
Base de fait:
a des poils
a des dents pointues
a des griffes
a les yeux dirigés en avant
est brun
a des taches noires
est un mammifère
est un carnivore
est un guépard

Entrer s.v.p. votre choix (0 a 8):? 8

Le système expert a été sauvegardé dans le fichier: SystemeExpert.txt

Entrer s.v.p. votre choix (0 a 8):? 0

Merci et au revoir !

4 Ce que vous devez rendre

Vous devez rendre un fichier .zip comportant uniquement les fichiers suivants :

- L'ensemble des fichiers .h contenant les commentaires d'interface enrichis en format Doxygen.
 - Regle.h; SystemeExpert.h; ListeCirculaire.h et AssertionException.h
- Les fichiers .cpp et .hpp complétés avec les commentaires d'implémentation en format Doxygen.
 - Regle.cpp; SystemeExpert.cpp; ListeCirculaire.hpp et Principal.cpp
- Animaux.txt et NoteTp1-IFT2008.xls (un fichier Excel contenant le barème de correction. Il faut juste ajouter votre nom et matricule sur la première ligne).

Vous ne devez en aucun cas ajouter un autre fichier ou répertoire. N'incluez aucun exécutable, ni aucun fichier généré par votre environnement de développement. De plus, vous n'avez pas à générer la documentation Doxygen en format html pour l'ensemble du projet (pour ne pas envoyer un gros fichier zip). Le correcteur pourra le faire lors de la correction.

Le nom du .zip doit respecter le format suivant : TP1-Matricule.zip. Nous vous rappelons qu'il est important de faire la remise par voie électronique uniquement en vous connectant à: <http://monportail.ulaval.ca/> (aucun travail envoyé par courriel n'est accepté). Il est toujours possible de faire plusieurs remises pour le même travail. Vous pouvez donc remettre votre travail plus tôt afin d'être sûr qu'il a été remis en temps, et remettre par la suite une version corrigée avant l'heure limite de remise. De plus, il est de votre responsabilité de vérifier après la remise que vous nous avez envoyé les bons fichiers (**non vides et non corrompus**), sinon vous pouvez avoir un zéro.

Attention ! Tout travail remis en retard se verra pénalisé de -25% de la note. Le retard débute dès la limite de remise dépassée (dès la première minute). Un retard excédant une journée provoquera le rejet du travail pour la correction et la note de 0 pour ce travail.

5 Remarques importantes

Normes de programmation :

Vous devez respecter les normes de programmation du cours. Vous devez également appliquer la théorie du contrat et utiliser les balises Doxygen pour commenter votre code.

Tolérance zéro vis-à-vis des erreurs de compilation :

Vos programmes doivent contenir zéro erreur de syntaxe. Un programme qui ne compile pas, peu importe la raison, se verra attribuer la note zéro.

Portabilité des programmes :

Vos programmes doivent absolument pouvoir être compilés et exécutés sans erreurs sous Eclipse sur la machine virtuelle du cours. Autrement, vous pouvez avoir la note zéro pour votre travail. Testez intensivement votre programme. Pour cela, nous recommandons l'utilisation de Google Test. **Par contre, SVP, ne pas remettre vos testeurs.** Rappelez-vous qu'un programme qui ne plante pas n'est pas nécessairement sans bogues, mais qu'un programme qui plante même une seule fois comporte nécessairement un bogue. Si vous ne testez pas suffisamment votre travail, il risque de provoquer des erreurs à l'exécution lors de la correction. La moindre erreur d'exécution rend la correction extrêmement difficile et vous en serez donc fortement pénalisés.

Bon travail.