ENSC 2ème année

Jean-Marc Salotti

Projet du module Intelligence Artificielle

Objectif général : Il s'agit de réaliser un programme informatique permettant de tester et évaluer de façon automatique les connaissances d'une personne dans le domaine de l'intelligence artificielle, sur la base des enseignements du module.

Projet à réaliser par groupe de 2 ou 3 élèves

Partie I :

Réalisez un programme en C# Windows Forms qui présente des questions simples à l'utilisateur pour évaluer ses connaissances en I.A. en reprenant des concepts clés vus en cours. Exemples :

1) L'algorithme MinMax est :
   a. Utilisé en I.A. pour jouer à deux contre un adversaire.
   b. Utilisé en I.A. pour trouver le plus court chemin dans un graphe.
   c. Utilisé en reconnaissance des formes pour détecter un isomorphisme de graphe
   d. Utilisé pour la détection de contours en vision par ordinateur
2) Soit M une matrice d'adjacence de taille NxN remplie avec des « a », des « b » ou éventuellement avec des « 0 » en cas d'absence de relation. On considère le code ci-dessous :

```
int cpt = 0 ;
for (int i=0 ; i<n ; i++)
   for (int j=0 ; j<n ; j++)
         X
```

   Par quoi faut-il remplacer X pour compter le nombre de relations non symétriques du graphe ?

   a. if ((M[i,j] == "a") || (M[i,j]== "b")) cpt++ ;

   b. if ((M[i,j] != "0") && (M[j,i] != M[i,j])) cpt++ ;

   c. Ni a, ni b, il faut d'abord modifier les lignes précédentes.

Toutes les questions et réponses doivent être stockées dans un fichier xml, de sorte qu'il soit possible d'en rajouter de nouvelles, sans avoir besoin de recompiler le programme. A l'exécution, le programme choisit au hasard une question, la présente à l'utilisateur. Puis, il évalue la réponse et indique la correction éventuelle. Au cours d'une même session, il faut pouvoir poser 20 questions différentes. A la fin des 20 questions, une note est donnée.

Elaborez des questions aussi pertinentes et diversifiées que possible, en visant le niveau d'un élève de l'ENSC qui aurait suivi le module de manière assidue. Si une question nécessite un temps de réflexion ou des calculs, comptez un point supplémentaire (par exemple si on doit déterminer le meilleur coup en appliquant MinMax à un arbre des coups jouables).

Partie II :

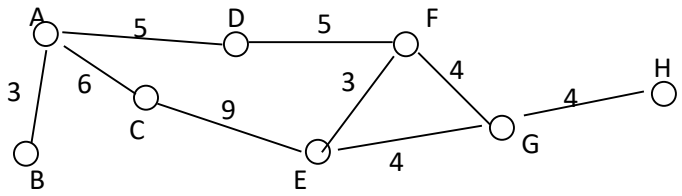
On reste dans le même cadre général de test des connaissances, on garde le même programme, mais on cherche à évaluer de manière spécifique la connaissance de l'algorithme de Dijkstra. Il s'agit de générer automatiquement la solution à un problème de recherche du plus court chemin dans un graphe et d'évaluer la réponse d'un utilisateur qui tente de faire tourner Dijkstra à la main. On attribuera 3 points à la réussite de cette question.

On pourra s'aider d'un code source C# qui permet d'appliquer l'algorithme de recherche du plus court chemin Dijkstra à partir d'un graphe décrit dans un fichier texte.

Faire tourner Dijkstra à la main consiste à indiquer l'évolution des ensembles $\mathcal{F}$ (Fermés) et $\mathcal{O}$ (Ouverts) ainsi que l'arbre de recherche final avec la distance de chaque nœud au nœud initial. On demandera donc à l'utilisateur de remplir $\mathcal{O}$ et $\mathcal{F}$ à chaque étape, jusqu'à ce que le plus court chemin menant au nœud final soit trouvé.

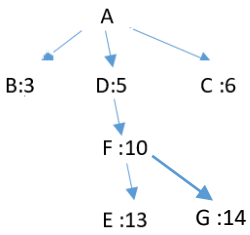
Exemple :

Appliquez Dijkstra pour trouver le plus court chemin entre A et E.



(1) $\mathcal{F}$={} $\mathcal{O}$ ={A} // ligne déjà remplie
(2) $\mathcal{F}$ ={A} $\mathcal{O}$ ={B,D,C} // lignes remplies par l'utilisateur
(3) $\mathcal{F}$ ={A,B} $\mathcal{O}$ ={D,C}
(4) $\mathcal{F}$ ={A,B,D} $\mathcal{O}$ ={C,F}
(5) $\mathcal{F}$ ={A,B,D,C} $\mathcal{O}$ ={F,E}
(6) $\mathcal{F}$ ={A,B,D,C,F} $\mathcal{O}$ ={E,G}
(7) $\mathcal{F}$ ={A,B,D,C,F,E} fin

Arbre final :



Le programme doit ensuite évaluer la proposition : à chaque étape, il faut vérifier que $\mathcal{F}$ et $\mathcal{O}$ sont justes, sans tenir compte de l'ordre (théoriquement, le meilleur est en première position de $\mathcal{O}$, mais on peut admettre que la sélection du meilleur se fasse indépendamment de sa position). Le programme doit également proposer un arbre final déjà bien structuré mais sans information sur les nœuds. Il faut alors vérifier que l'ascendant de chaque nœud est correct et que sa distance au nœud initial est juste. On notera sur 2 points le remplissage des 2 ensembles (0 en cas d'erreur) et sur 1 point l'arbre (0 en cas d'erreur). Pour l'affichage de l'arbre, il est possible (mais pas obligatoire) d'exploiter un objet Treeview de C#. Notez qu'il faut que le programme soit générique, de sorte qu'il soit possible de le faire tourner sur des graphes différents.

Evaluation :

Le livrable sera constitué des éléments suivants :

- les sources complètes du projet C# (pour pouvoir être testé par l'enseignant)
- un rapport constitué de
    - un résumé de ce qui a été fait, avec éventuellement les orientations spécifiques qui ont été prises (types de question, choix de l'interface et des interactions …)
    - la répartition détaillée des tâches au sein du groupe (une disparité trop importante peut conduire à une notation différenciée)
    - la liste des questions avec les réponses attendues
    - des copies d'écran illustrant l'interface et le bon déroulement de toutes les étapes de l'évaluation.

Vous serez notés de la façon suivante :

- 10 points sur la partie I, dont 8 sur les questions posées (pertinence et diversification) et 2 sur la réalisation (justesse et ergonomie du programme et qualité du rapport)
- 10 points sur la partie II, dont 8 sur la réalisation informatique et 2 sur l'ergonomie du programme et la qualité du rapport).