UV IA41 – Responsable : Fabrice Lauri
Concepts fondamentaux en Intelligence Artificielle et langages dédiés

Examen Final
Juin 2014

Coefficient : 40%.
Aucun document autorisé.

Remarques et conseils :

1) Lisez **attentivement** chaque question avant d'y répondre.
2) Indiquez clairement sur votre copie le numéro de l'exercice avant d'y répondre.
3) Lorsque vous définissez un prédicat, son profil et sa définition formelle doivent être au moins indiqués. Un jeu d'essais est facultatif.
4) Expliquez autant que possible vos choix lors de la définition d'un prédicat ou d'une fonction.
5) Le barème défini ci-après est susceptible d'être modifié.

Exercice 1 : Questions de cours (4 points)

1. Donnez une définition de l'Intelligence Artificielle. **(1 point)**
2. L'un des quatre courants de pensée en IA est la pensée rationnelle. Rappelez de quoi il s'agit. Est-ce que Prolog est issu de ce courant de pensée ? **(1 point)**
3. A quel type de problème appartient le problème du taquin ? Quels sont les données du problème et l'inconnue ? Rappelez la terminologie employée pour ce type de problème. Quel(s) algorithme(s) vu(s) en cours permet de le résoudre ? **(2 points)**

Exercice 2 : Prolog (6 points)

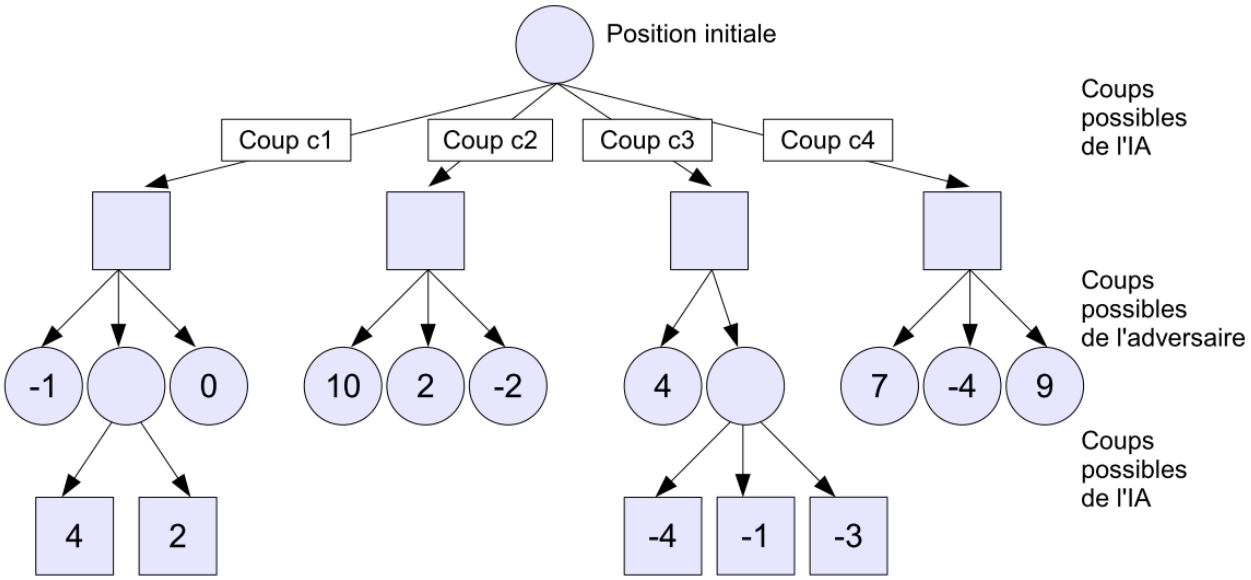
Soit un ensemble d'objets caractérisés par leur couleur (rouge, vert, bleu, jaune ou orange), leur forme (carré ou circulaire) et leur taille (longueur d'un côté du carré ou rayon du cercle).

Questions :

1. Donnez une représentation d'un tel ensemble d'objets en Prolog.
2. Ecrivez le prédicat **objet(+L)** qui vérifie que L est un objet décrit par trois caractéristiques de couleur, forme et taille.
3. Ecrivez le prédicat **objectSet(+L)** qui vérifie que L est un ensemble d'objets décrits par trois caractéristiques de couleur, forme et taille. Chaque objet ne doit apparaître qu'une fois dans l'ensemble.
4. Ecrivez le prédicat **buildObjectSet(+L,?R)** qui est satisfait si R est l'ensemble des objets contenus dans L, sans redondance d'objets.
5. Ecrivez le prédicat **objectSubset(+L,+O,?R)** qui est satisfait si R est l'ensemble d'objets extraits de L ne contenant que des objets de caractéristiques O. Dans O, une caractéristique définie à * indique qu'elle n'est pas prise en compte dans la sélection des objets. Par exemple, si une taille de * est indiquée dans O, les objets de toute taille respectant les deux autres critères seront présents dans R.

Exercice 3 : Algorithmes Min-Max et Alpha-Bétâ (5 points)

Les algorithmes Min-Max et Alpha-Bétâ sont largement utilisés pour déterminer le meilleur coup qu'une intelligence artificielle (joueur cybernétique) doit réaliser lorsqu'elle est confrontée à un autre joueur dans le cadre d'un jeu à somme nulle.
L'arbre ci-dessous représente l'arbre de recherche d'un tel jeu à deux joueurs. L'évaluation de certaines positions pouvant être atteintes à partir de la position initiale est indiquée par des valeurs entre -10 et 10 dans cet arbre.



L'IA est supposée maximiser ses gains, tandis que l'adversaire les minimise.

Questions :

1. Décrivez aussi précisément que possible l'algorithme Min-Max.
2. En utilisant l'algorithme Min-Max, précisez pour chaque nœud son évaluation et indiquez le coup qu'effectuera l'IA.
3. Décrivez aussi précisément que possible l'algorithme Alpha-Bétâ.
4. En utilisant l'algorithme Alpha-Bétâ, précisez pour chaque nœud ses bornes finales [alpha; bétâ]. Indiquez également avec une croix les coups qui seront élagués par l'algorithme et donnez enfin le coup qu'effectuera l'IA.
5. Quel(s) critère(s) dans la construction de l'arbre de recherche permettrai(en)t d'améliorer l'efficacité de l'élagage de Alpha-Bétâ ? Si ce critère est appliqué, combien de nœuds dans l'arbre de recherche ci-dessus pourraient alors être élagués ?

Exercice 4 : Algorithme A* (5 points)

Le problème à résoudre en utilisant l'algorithme A* consiste à passer de la séquence de symboles YBRG à la séquence RGBY. Une permutation de deux lettres permet de passer d'une séquence à une autre. Par exemple, la permutation de la lettre Y avec la lettre G permet de passer de la séquence YBRG à la séquence GBRY.

Questions :

1. Donnez une représentation d'un état pour résoudre le problème exposé ci-dessus avec A*.
2. Exprimez la fonction successeur **succ(x)** pour chacun des états **x** nécessaires à la résolution.

3. Utilisez une heuristique, si possible admissible, et résolvez ce problème avec l'algorithme A* en indiquant à chaque itération de l'algorithme (voir figure ci-dessous) :
   - le contenu de la liste Open,
   - le contenu de la liste Closed,
   - les valeurs de g, de h et de f pour le nœud courant extrait de la liste Open.

   On suppose qu'à l'itération 1, la liste *Open* ne contient que le nœud correspondant à l'état initial.

Itération	**Liste *Open***	**Liste *Closed***	**Noeud N extrait de la liste *Open***	**Valeur g du noeud N**	**Valeur h du noeud N**	**Valeur f du noeud N**
1						
2						
3						
4						
...						

4. Combien de nœuds sont développés sans utiliser d'heuristique (c'est-à-dire lorsque h(x) = 0 quel que soit l'état x) ?
5. Rappelez ce qu'est une heuristique admissible et montrez que celle que vous avez utilisé dans la question 3 est ou n'est pas admissible.