

Représentation des connaissances

Evaluation Ecrite

Durée : 2h

Matériel autorisé : une feuille A4 recto/verso de notes personnelles

1 Algorithmes de recherche [3pt]

Q. 1 : Qu'est-ce qu'une heuristique admissible ?

Q. 2 : Sous quelles conditions l'algorithme A se comportera comme l'algorithme de recherche en largeur d'abord ?

2 Alpha-Beta [4pt]

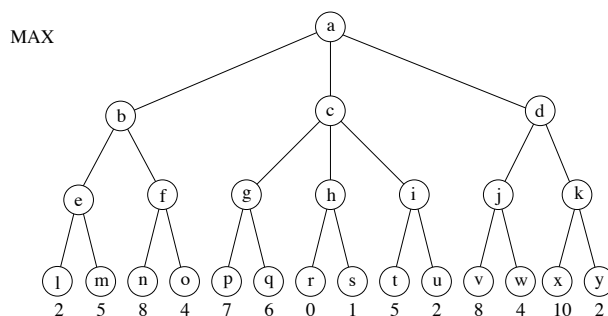


Figure 1: Arbre de jeu. La racine est un nœud maximisant. Chaque lettre identifie un nœud. Chaque chiffre correspond à l'évaluation d'une feuille.

1. Appliquer l'algorithme α - β à l'arbre de jeu donné par la figure 1. Supposer que les fils des nœuds sont parcourus de droite à gauche et indiquer tous les nœuds qui ne sont pas considérés par l'algorithme.

3 IDA* [6 pt]

Le jeu du Taquin (aussi appelé 8-puzzle en anglais) est composé de 8 pièces numérotées et d'une case vide que l'on peut déplacer. Le but est partant d'une configuration mélangée de retrouver la configuration *but* illustrée par la figure 4.

Appliquer l'algorithme IDA* au problème du Taquin avec la meilleure heuristique vue en cours. Vous trouverez ci-dessous l'algorithme A* pour vous aider dans cette démarche.

Algorithme 1 Alpha-Beta

```

 $\alpha$ - $\beta$ (Node  $s$ , int  $\alpha$ , int  $\beta$ ) //La valeur d'un nœud est encadrée par  $\alpha$  et  $\beta$ .
si isTerminal( $s$ ) alors
    return (GainEvaluation( $s$ ))
sinon
    // appliquer alpha-beta sur les fils de  $s$  et mémoriser leurs valeurs
     $s$ .alpha =  $-\infty$ ;
     $s$ .beta =  $+\infty$ ;
    si isComputerTurn( $s$ ) alors
        pour tout successeur  $si$  de  $s$  faire
             $val = \alpha$ - $\beta$ ( $si$ , max( $\alpha$ ,  $s$ .alpha),  $\beta$ )
             $s$ .alpha = max( $s$ .alpha,  $val$ )
            si  $s$ .alpha  $\geq \beta$  alors
                break
            fin si
        fin pour
        return  $s$ .alpha
    sinon
        // tour de l'adversaire
        pour tout successeur  $si$  de  $s$  faire faire
             $val = \alpha$ - $\beta$ ( $si$ ,  $\alpha$ , min( $\beta$ ,  $s$ .beta))
             $s$ .beta = min( $s$ .beta,  $val$ )
            si  $s$ .beta  $\leq \alpha$  alors
                break
            fin si
        fin pour
        return  $s$ .beta
    fin si
fin si

```

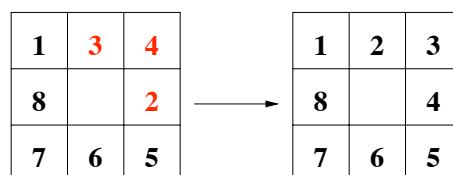


Figure 2: Un état initial (à gauche) et l'état-but (à droite) du jeu du taquin

Algorithme 2 A (A^* si h est admissible)

s_0 est l'état initial
OUVERTS est la liste des nœuds à étudier
FERMES est la liste des nœuds étudiés
la fonction tête(OUVERTS) rend le premier élément de la liste OUVERTS
la fonction chemin(s) appelle chemin(pere(s)) si $s \neq s_0$ puis rend s

OUVERTS $\leftarrow \{s_0\}$
FERMES $\leftarrow \emptyset$
 $g(s_0) \leftarrow 0$
 $s \leftarrow s_0$
tant que OUVERTS $\neq \emptyset$ et tête(OUVERTS) non terminal **faire**
 Supprimer s de OUVERTS et le mettre dans FERMES
 pour tout nœud s' successeur de s **faire**
 si $s' \notin (OUVERTS \cup FERMES)$ ou $g(s') > g(s) + \text{cout}(s, s')$ **alors**
 $g(s') \leftarrow g(s) + \text{cout}(s, s')$
 $f(s') \leftarrow g(s') + h(s')$
 $\text{pere}(s') \leftarrow s$
 Ranger s' dans OUVERTS dans l'ordre f croissant
 si $s' \in FERMES$ **alors**
 FERMES $\leftarrow FERMES \setminus \{s'\}$
 fin si
 fin si
 fin pour
 si OUVERTS $\neq \emptyset$ **alors**
 $s \leftarrow \text{tete}(OUVERTS)$
 fin si
fin tant que
si OUVERTS $= \emptyset$ **alors**
 le problème n'admet pas de solution
sinon
 le chemin(s) est solution
fin si

4 Construction d'un arbre de décision [4pt]

On dispose d'un échantillon de 200 patients pour lesquels on a relevé trois informations : l'état de leur gorge (irritée ou non), leur température (≥ 37.5 ou non) et si ils étaient malades ou sains (d'après un médecin). Le tableau de données individus x variables

	Gorge	Température	Classe
patient 1	irritée	basse (< 37.5)	Sain
patient 2	non irritée	basse (< 37.5)	Sain
patient 3	irritée	haute (≥ 37.5)	Malade
...	...	...	...

peut être ici résumé par :

	gorges irritée	gorge non irritée
température < 37.5	(6 S , 37 M)	(91 S, 1 M)
température ≥ 37.5	(2 S, 21 M)	(1 S, 41 M)

avec S pour Sain et M pour Malade. On constate que 100 patients sont malades et 100 autres bien portants.

1. Quel est le premier test le plus discriminant pour construire l'arbre de décision de type CART (arbre binaire utilisant l'indice d'impureté de Gini) associé aux exemples ci-dessus ? Contentez vous d'écrire les formules instanciées.

5 Logique propositionnelle [3pt]

Si la température et la pression sont constantes, alors il ne pleut pas.
La température était constante.

A partir du texte précédent, est-il vrai que s'il ne pleuvait pas, alors la pression était constante ?

1. Transformez le texte ci-dessus en formules logiques.
2. Convertissez les expressions en forme normale conjonctive
3. Construire une preuve formelle en utilisant les règles d'inférence.