UNIVERSITÉ DE SHERBROOKE
Département d'informatique

IFT 615
Intelligence artificielle

Examen périodique
Hiver 2008

Le vendredi 29 février, 10 h 30 à 12 h 20, au D7-2017

PROFESSEUR
Froduald Kabanza, PhD
http://www.planiart.usherbrooke.ca/kabanza

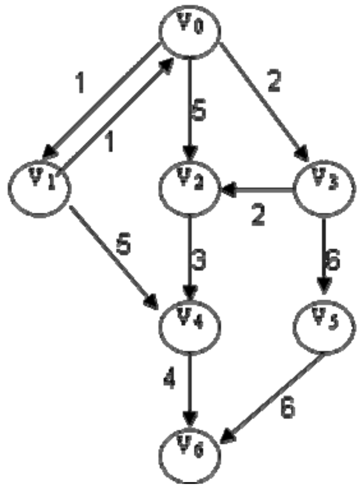
ASSISTANT
Éric Beaudry, Msc
http://www.planiart.usherbrooke.ca/eric

INSTRUCTIONS

SOLUTIONS

Question 1 (3 points) – Algorithme A*

Soit l'espace d'états (v_1, …, v_6) et la fonction de transition représentés par le graphe suivant : un arc allant de l'état v_i à l'état v_j signifie que v_j est parmi les successeurs de v_i; le nombre étiquetant l'arc est le coût de la transition entre les deux états.



Soit la fonction heuristique *h* définie sur l'espace d'états précédent comme suit :

État	v_0	v_1	v_2	v_3	v_4	v_5	v_6
h	6	6	3	7	8	3	0

a. (2 points) Donnez, ci-après, une trace d'exécution de A* avec comme entrées : la fonction de transition décrite par le graphe précédent; l'état initial v_0; la fonction de but qui retourne vrai pour l'état v_6 et faux pour tout autre argument; et la fonction heuristique *h*. Votre trace doit indiquer avec exactitude le contenu des listes *open* et *close*, à chaque itération de A*. Pour chaque état dans ces listes, indiquez dans l'ordre et entre parenthèses, les composantes (*état*, *valeur g*, *valeur f*, *parent*), séparées par une virgule. L'ordre des données dans chaque liste doit correspondre à l'implémentation de A*.

Itération	*open*	*close*
0	(V_0, 0, -)	
1	(v_1, 1, 7, v_0) (v_2, 5, 8, v_0) (v_3, 2, 9, v_0)	(v_0, 0, -)
2	(v_2, 5, 8, v_0) (v_3, 2, 9, v_0) (v_4, 6, 14, v_1)	(v_1, 1, 7, v_0) (v_0, 0, -)

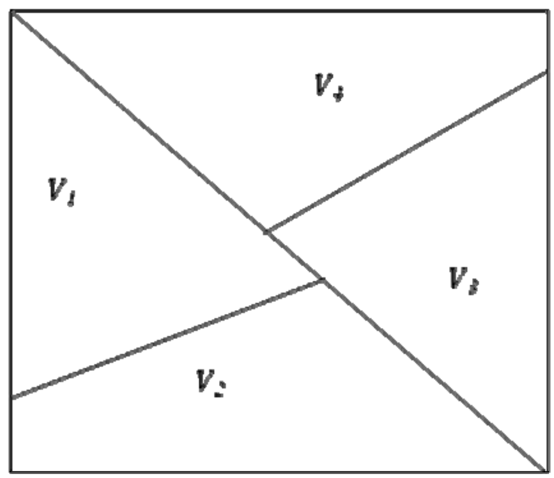
3	(v_3, 2, 9, v_0) (v_4, **6**, **14**, $\mathbf{v_1}$)	(v_2, 5, 8, v_0) (v_1, 1, 7, v_0) (v_0, 0, -)
4	**($\mathbf{v_2}$, 4, 7, $\mathbf{v_3}$)** (v_5, 8, 11, v_3) (v_4, **6, 14,** $\mathbf{v_1}$)	(v_3, 2, 9, v_0) (v_1, 1, 7, v_0) (v_0, 0, -)
5	(v_5, 8, 11, v_3) (v_4, 6, 14, v_1)	(v_2, 4, 7, v_3) (v_3, 2, 9, v_0) (v_1, 1, 7, v_0) (v_0, 0, -)
6	(v_6, 14, 14, v_5) (v_4, 6, 14, v_1)	(v_5, 8, 11, v_3) (v_2, 4, 7, v_3) (v_3, 2, 9, v_0) (v_1, 1, 7, v_0) (v_0, 0, -)
7	(v_4, 6, 14, v_1) Arrêt! V6 est le but!	(v_6, 14, 14, v_5) (v_5, 8, 11, v_3) (v_2, 4, 7, v_3) (v_3, 2, 9, v_0) (v_1, 1, 7, v_0) (v_0, 0, -)
8	Note : c'est correcte d'arrêter à l'étape 6. C'est correcte aussi de visiter v_4 avant v_6 puisque $f(v_4)=f(v_6)=14$.	
9		
10		
11		

b. (1 point) Avec le graphe précédent, indiquez la solution qui serait retournée par l'algorithme de Dijkstra pour l'état initial v_0 et l'état final v_6.

V

Question 2 (2 points) – Satisfaction des contraintes

Soit la carte suivante décrivant les frontières entres quatre villes (V_1, …, V_4). On voudrait colorier la carte en utilisant seulement les couleurs rouge, bleu et vert, de sorte que V_1 soit en rouge ou en vert; V_2 et V_3 soient en bleu ou en vert; et V_4 soit en vert. Toutefois, deux villes adjacentes ne peuvent avoir la même couleur.



(2 points) Donnez le résultat de l'algorithme AC-3 sur ce problème. Que concluez-vous de ce résultat?

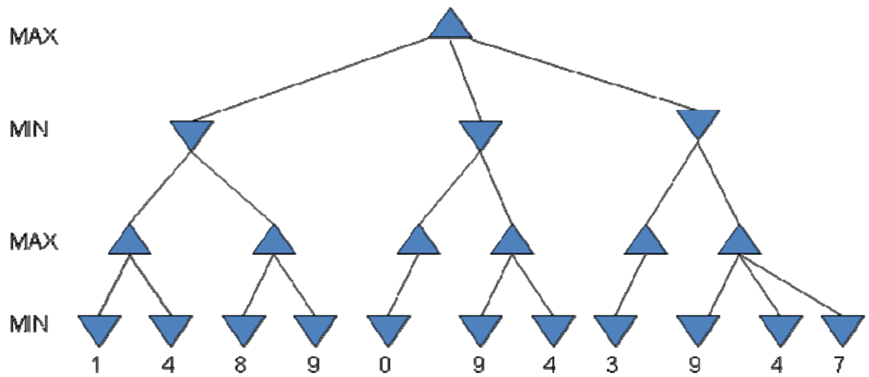
V_1	V_2	V_3	V_4	*AC-3*
{R, V}	{B,V}	{B,V}	{V}	
{R}	{B,V}	{B,V}	{V}	$V_1 – V_4$
{R}	{B,V}	{B }	{V}	$V_3 – V_4$
{R}	{V}	{B}	{V}	$V_2 – V_3$

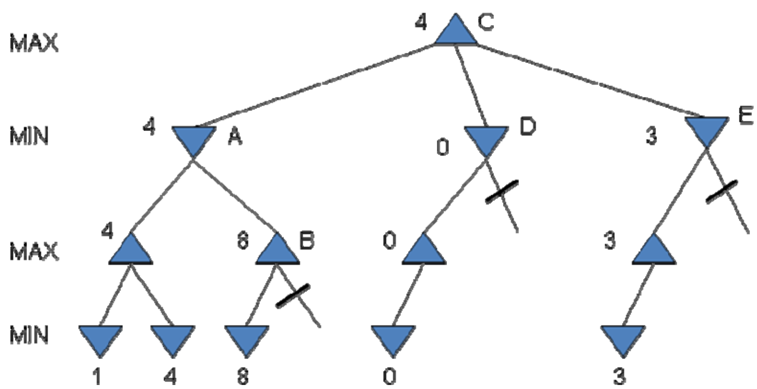
Conclusion :

Le domaine de chaque variable est réduit exactement à une seule valeur. Comme il n'y a que des contraintes binaires, on a une solution.

Question 3 (6 points) – *Alpha-Beta Pruning*

a. (5 points) Soit l'espace d'états suivant modélisant les actions de deux joueurs (MAX et MIN). Les feuilles correspondent aux états terminaux du jeu. Les valeurs des états terminaux sont indiquées en bas de chaque état. Dessinez la partie de l'espace d'états qui serait explorée par l'algorithme *alpha-beta pruning*, en supposant qu'il explore l'espace d'états de la gauche vers la droite. Dessinez seulement les états explorés et les transitions correspondantes. Indiquez, à côté de chaque état exploré, la valeur correspondante à la terminaison de l'algorithme.





A a β = 4 (A ne sera jamais plus grand que 4)

B est β-coupé puisque 8 > β = 4

C a α = 4 (C ne sera jamais plus petit que 4)

D est α-coupé puisque 0 < α = 4

E est α-coupé puisque 3 < α = 4

C a la valeur 4.

Rappel :

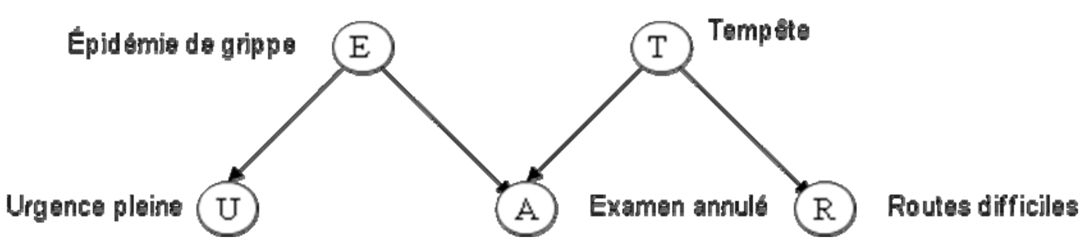
- α : meilleure (plus grande) valeur de MAX jusqu'ici; elle ne décroît jamais; MAX ne considère jamais les nœuds (en bas de lui – successeurs) ayant des valeurs plus petites que α.
- β : meilleure (plus petite) valeur de MIN jusqu'ici; elle ne croît jamais; MIN ne considère jamais les nœuds (en bas de lui – successeurs) ayant des valeurs plus grandes que β.

b. (1 point) Supposons que l'algorithme *alpha-beta pruning* explore le même espace d'états, mais cette fois-ci de la droite vers la gauche. À la terminaison, l'état à la racine aura-t-il la même valeur que dans l'exploration précédente? Répondez par oui ou non et ensuite expliquez clairement pourquoi.

Oui. Parce que *alpha-beta pruning* donne toujours le même résultat que minimax; par contre, les états explorés ou coupés ne seront pas forcément les mêmes selon qu'on explore l'espace de gauche vers la droite ou de droite vers la gauche.

Question 4 (4 points) – Réseaux bayésiens

Soit le réseau bayésien (RB) suivant dans lequel les nœuds représentent des variables booléennes. Le RB établit une relation de causalité entre une épidémie de grippe et l'annulation d'un examen et le fait que l'urgence du CHUS devienne pleine. Il y a aussi une corrélation entre une tempête de neige et l'annulation d'un examen de même que des conditions routières difficiles.



Soit les distributions de probabilité conditionnelle suivantes (*v* indique vrai ou « true » et *f* indique « faux » ou « false ») :

E	P(U|E)
v	0.2
f	0.02

E	T	P(A|E,T)
v	v	0.70
v	f	0.20
f	v	0.50
f	f	0.01

T	P(R|T)
v	0.8
f	0.1

P(E)
0.05

P(T)
0.4

a. (1 point) Indiquez la façon la plus simple possible de calculer la distribution jointe P(E, A, U, T, R).

P(E, A, U, T, R) = P(A|E,T)P(U|E)P(R|T)P(E)P(T)

b. (1 point) Calculez la probabilité P(e,a,u,t,r), c.-à-d., P(E=v, A=v, U=v, T=v, R=v).

P(e,a,u,t,r) = 0.70*0.2*0.8*0.05*0.4 = 0.00224

c. (1 point) Indiquez la façon la plus simple possible de calculer P(U|E,T).

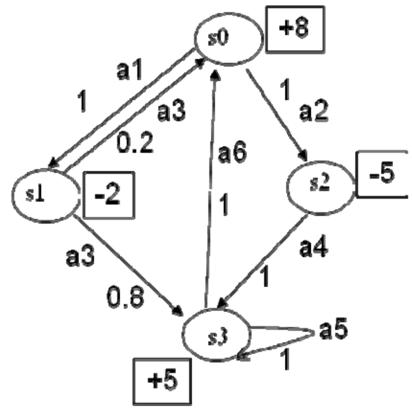
P(U|E,T) = P(U|E)   Parce que U est (conditionnellement) indépendant de T étant donné E.

d. (1 point) Calculez la probabilité P(u|e,t), c.-à-d., P(U=v | e=v, t=v).

P(u|e,t) = P(u|e) = 0.2   Selon la table de probabilité conditionnelle.

Question 5 (4 points) Planification par les processus de décision de Markov

Soit le processus de décision markovien suivant, dont les transitions sont étiquetées par les noms des actions et les probabilités de transitions; les états sont étiquetés par les récompenses correspondantes (exprimant une fonction d'utilité).



a. (2 points) Dans le tableau suivant, indiquez les valeurs des états à la fin de chacune des trois premières itérations de l'algorithme *value-itération*, en supposant qu'on utilise un facteur d'atténuation (*discount factor*) de 0.5 et en partant initialement (itération 0) avec des valeurs des états toutes égales à 0.

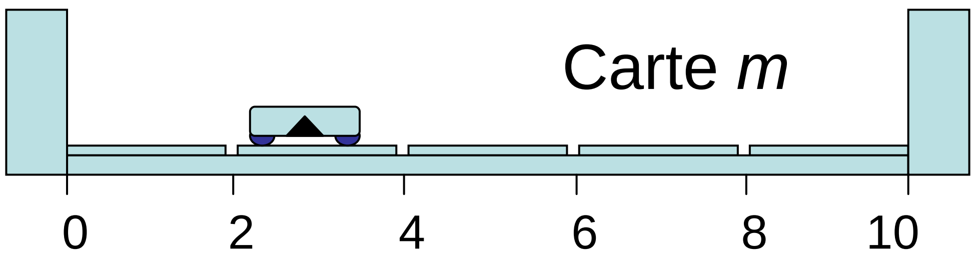
	Valeurs Itération 1	**Valeurs Itération 2**	**Valeurs Itération 3**
s_0	**8**	8 + 0.5 * 1 *-2= **7**; 8 + 0.5 * 1*-5 = 5.5	8+0.5*1*0.8= **8.4**; 8+0.5*1*-2.5=6.5
s_1	**-2**	-2 + 0.5 * (0.2*8 + 0.8*5) = **0.8**	-2+0.5*(0.2*7+0.8*9)=**2.3**
s_2	**-5**	-5 + 0.5*1*5= **-2.5**	-5+0.5*1*9= **- 0.5**
s_3	**5**	5 + 0.5*1*5= 7.5; 5+0.5*1*8=**9**	5+0.5*1*9= **9.5** ; 5+0.5*1*7= 8.5

b. (2 points) Donnez le plan d'actions (*policy*) correspondant aux valeurs à chaque itération.

	Plan Itération 1	**Plan Itération 2**	**Plan Itération 3**
s_0	**a1**	**a1**	**a1**
s_1	a3	a3	a3
s_2	a4	a4	a4
s_3	**a6**	**a5**	**a5**

Question 6 (1 point) - Exposé d'Éric Beaudry

Un fabricant de robots vient de mettre au point un robot aspirateur. Ce robot peut avancer jusqu'à 100 millimètres par seconde. Similairement à une automobile, ce robot ne peut pas tourner sur lui-même, mais en avançant, il peut tourner jusqu'à plus ou moins 50 degrés par mètre. Le système de contrôle du robot est basé sur une architecture comportementale intégrant des modules comportementaux permettant de se déplacer et d'éviter les obstacles. Pour percevoir les obstacles, il dispose de trois sonars installés à l'avant. Pour se déplacer en ligne droite, le robot est équipé d'un capteur pouvant détecter des jointures de tuiles afin de l'aider à se guider. Ce capteur est installé au centre du robot. Lorsque le robot passe sur une jointure de tuile, il a 80 % de chance de la détecter. À l'opposé, il n'y a jamais de fausse alarme. En d'autres mots, lorsque le robot est sur une tuile, il ne signalera jamais à tort une jointure. On voudrait implémenter un module de localisation à l'aide d'un filtre de Kalman ou d'un filtre à particules. Pour des fins de simplicité, une seule dimension est considérée.



Proposez (dessinez) une fonction de densité pour le modèle d'observation *p(x | d=j, m)*, signifiant la probabilité que le robot soit à la position x lorsque le détecteur perçoit une jointure (d=j) sachant la carte *m*. Seule l'allure de la courbe est corrigée, il n'est pas nécessaire d'écrire les valeurs exactes.