Solution Examen final

Intelligence Artificielle II (IFT-17587)

Jeudi 29 avril 2004

De 8h30 à 11h20 en salles PLT-2551

- *Tout document est permis.*
- *Le nombre de points accordés à chacune des questions est inscrit entre parenthèses.*
- *Le questionnaire a 6 questions sur 3 pages.*

----

1 (10 pts) Dans l'environnement simple suivant, une souris doit se promener dans l'environnement en essayant d'éviter de rencontrer un chat. La souris a appris à sentir la présence d'un chat une case d'avance. Donc, lorsqu'il y a un chat sur une case, la souris perçoit une odeur dans les cases adjacentes. Dans le problème suivant, on considère que la souris a commencé dans la case (1,1), quelle a fait un déplacement vers la droite en (2,1) et un mouvement vers le bas en (2,2). Dans les deux premières cases qu'elle a parcourues, elle n'a rien perçue. Toutefois, en (2,2), elle a perçu une odeur. Partant de ses observations, la souris aimerait déterminer les cases sécuritaires, c'est-à-dire, quelles sont les cases qui contiennent un chat parmi les quatre cases non visitées. En utilisant l'algorithme d'inférence de vérification de modèles, dites (en expliquant) si la base de connaissances découlant des perceptions de la souris permet d'inférer les phrases suivantes.

a) Il n'y a pas de chat en (1,2).

b) Il n'y a pas de chat en (3,2).

1	Rien	Rien	?
2	?	Odeur	?
3		?	
	1	2	3

La présence où non d'un chat dans les quatre cases peut être représentée par quatre variables booléennes. On peut représenter la base de connaissance de l'agent ainsi que la véracité des deux phrases à l'aide de la table de vérité suivante où sont listées tous les modèles possibles.

Cas à un seul chat :

(1,2)	(2,3)	(3,1)	(3,2)	BC	a)	b)
V	V	V	V	F	F	F
V	V	V	F	F	F	V
V	V	F	V	F	F	F
V	V	F	F	F	F	V
V	F	V	V	F	F	F
V	F	V	F	F	F	V
V	F	F	V	F	F	F
V	F	F	F	F	F	V
F	V	V	V	F	V	F
F	V	V	F	F	V	V
F	V	F	V	F	V	F
F	V	F	F	V	V	V
F	F	V	V	F	V	F
F	F	V	F	F	V	V
F	F	F	V	V	V	F
F	F	F	F	F	V	V

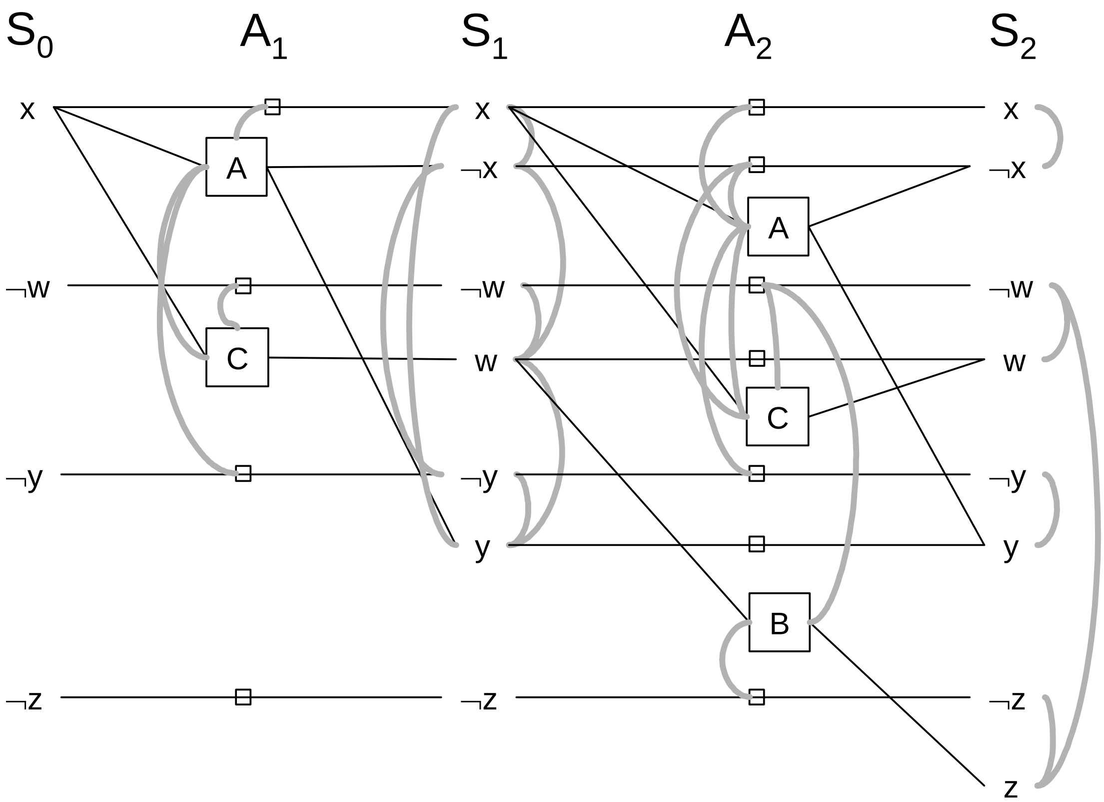
Cas où il peut y avoir plus d'un chat :

(1,2)	(2,3)	(3,1)	(3,2)	BC	a)	b)
V	V	V	V	F	F	F
V	V	V	F	F	F	V
V	V	F	V	F	F	F
V	V	F	F	F	F	V
V	F	V	V	F	F	F
V	F	V	F	F	F	V
V	F	F	V	F	F	F
V	F	F	F	F	F	V
F	V	V	V	F	V	F
F	V	V	F	F	V	V
F	V	F	V	V	V	F
F	V	F	F	V	V	V
F	F	V	V	F	V	F
F	F	V	F	F	V	V
F	F	F	V	V	V	F
F	F	F	F	F	V	V

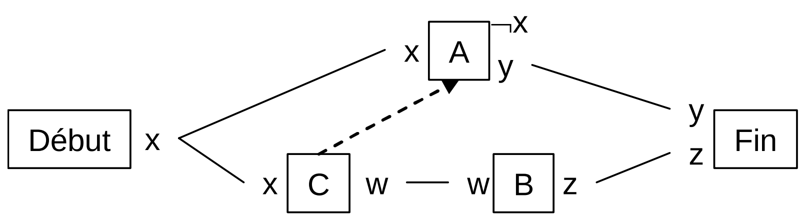
Le résultat est le même qu'il y ait un ou plusieurs chats. Partout où la base de connaissance est vraie, la phrase en a) est vraie, donc M(BC) $\subseteq$ M(a)), par conséquent, BC permet d'inférer a). Il y a au moins un cas où la base de connaissance est vraie et où b) est fausse, donc M(BC) $\not\subset$ M(b)), par conséquent, BC ne permet pas d'inférer b).

2 (18 pts) En considérant la description du problème ici-bas, utilisez l'algorithme « Graphplan » pour trouver le plan ou les plans permettant d'atteindre le but tel que précisé ci-dessous. Vous devez présenter toutes les étapes de l'algorithme clairement.

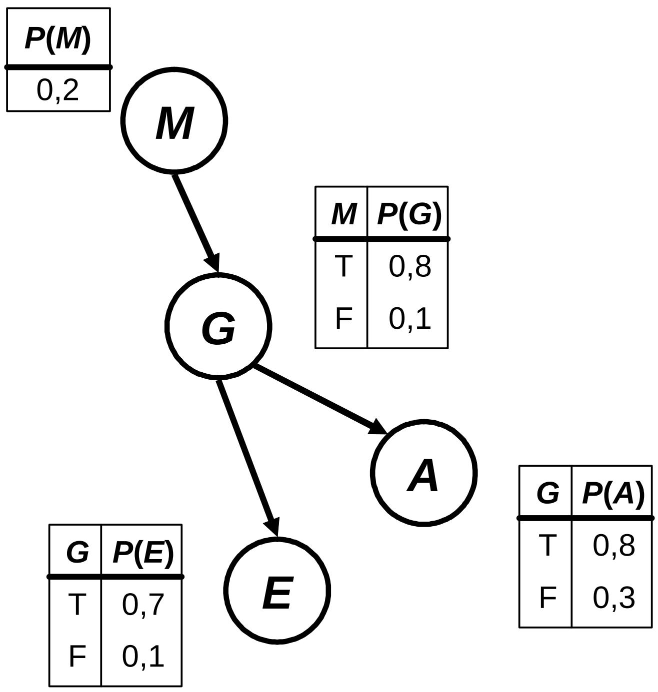
INITIAL(x)
BUT($y \wedge z$)
Action(A, PRECOND : x, EFFET : $\neg x \wedge y$)
Action(B, PREDOND: w, EFFET : z)
Action(C, PREDOND: x, EFFET : w)



À partir de ce Graph de planification, on peut extraire le plan suivant :



3 (18 pts) En considérant le réseau bayésien suivant, calculez la distribution de probabilités **P**($G|e$) à l'aide l'algorithme d'inférence par élimination de variables. Vous devez montrer tous les calculs.



Réponse :

$$P(G|e) = \alpha \sum_a \sum_m P(m)P(G|m)P(e|G)P(a|G)$$

$$P(G|e) = \alpha P(e|G) \sum_m P(m)P(G \mid m) \sum_a P(a \mid G)$$

$$P(G|e) = \alpha P(e|G) \sum_m P(m)P(G \mid m)$$

$$f_G(G,M) = \begin{pmatrix} P(g,m) & P(g,\neg m) \\ P(\neg g,m) & P(\neg g,\neg m) \end{pmatrix} = \begin{pmatrix} 0,8 & 0,1 \\ 0,2 & 0,9 \end{pmatrix}$$

$$f_M(M) = \begin{pmatrix} P(m) \\ P(\neg m) \end{pmatrix} = \begin{pmatrix} 0,2 \\ 0,8 \end{pmatrix}$$

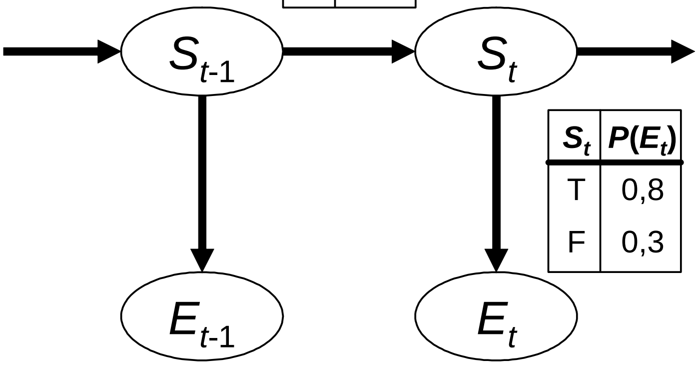
$$f_{\overline{M}G}(G) = \sum_m f_M(m) \times f_G(G,m) = f_M(m) \times f_G(G,m) + f_M(\neg m) \times f_G(G,\neg m)$$

$$= \begin{pmatrix} f_{\overline{M}G}(g) \\ f_{\overline{M}G}(\neg g) \end{pmatrix} = \begin{pmatrix} 0,2 \times 0,8 + 0,8 \times 0,1 \\ 0,2 \times 0,2 + 0,8 \times 0,9 \end{pmatrix} = \begin{pmatrix} 0,24 \\ 0,76 \end{pmatrix}$$

$$f_E(G) = \begin{pmatrix} P(e \mid g) \\ P(e \mid \neg g) \end{pmatrix} = \begin{pmatrix} 0,7 \\ 0,1 \end{pmatrix}$$

$$P(G|e) = \alpha\, f_E(G) \times f_{\overline{M}G}(G) = \alpha < 0.7 \times 0.24, 0.1 \times 0.76 >= \alpha < 0.168, 0.076 >=< 0.689, 0.311 >$$

4 (18 pts) En considérant le réseau bayésien dynamique suivant, montrez tous les calculs du processus de filtrage pour les deux premiers états (S_1 et S_2). Pour cela, il faudra considérer que $\mathbf{P}(S_0) = <0.4, 0.6>$, E_1 = *vrai* et E_2 = *faux*.



S_{t-1}	$P(S_t)$
T	0,9
F	0,2

S_t	$P(E_t)$
T	0,8
F	0,3

E_1 = *vrai*, donc la prédiction du temps $t = 0$ à $t = 1$ est :

$$P(S_1) = \sum_{s_0} P(S_1 \mid s_0)P(s_0) = <0.9, 0.1> \times 0.4 + <0.2, 0.8> \times 0.6 = <0.48, 0.52>$$

La mise à jour avec l'évidence au temps $t = 1$ est :

$$P(S_1 \mid e_1) = \alpha P(e_1 \mid S_1)P(S_1) = \alpha <0.8, 0.3><0.48, 0.52> = \alpha <0.384, 0.156> = <0.71, 0.29>$$

E_2 = *faux*, donc la prédiction du temps $t = 1$ à $t = 2$ est :

$$P(S_2 \mid e_1) = \sum_{s_1} P(S_2 \mid s_1)P(s_1 \mid e_1) = <0.9, 0.1> \times 0.71 + <0.2, 0.8> \times 0.29 = <0.697, 0.303>$$

La mise à jour avec l'évidence au temps $t = 2$ est :

$$P(S_2 \mid e_1, e_2) = \alpha P(e_2 \mid S_2)P(S_2 \mid e_1) = \alpha <0.2, 0.7><0.697, 0.303>$$
$$= \alpha <0.1394, 0.2121> = <0.397, 0.603>$$

5 (18 pts) Bob désire s'acheter une maison. La maison a une valeur sur le marché de 100 000$, mais il pourrait l'obtenir pour 90 000$. Toutefois, avant d'acheter la maison, il aimerait bien la faire évaluer pour être certain de faire un bon choix. L'évaluateur lui charge 1000$ pour faire l'évaluation. Si jamais la maison n'est pas en bon état, le coût des réparations pourrait aller jusqu'à 15 000$. Bob estime que la maison a 80% de chance d'être en bon état. La probabilité que l'évaluateur dise que la maison est en bon état si elle est effectivement en bon état est de 90%, de plus la probabilité qu'il dise qu'elle est en bon état si elle n'est pas en bon état (donc qu'il ne voit pas le problème) est de 15%.

a) Quel est le gain espéré de Bob en achetant cette maison sans faire de test?

EU(acheter) = P(bonEtat) Gain(bonEtat) + P(¬bonEtat) Gain(¬bonEtat)
= 0.8 * (100 000 – 90 000) + 0.2 * (100 000 – 105 000) = 7000$

b) Quel est le gain espéré de Bob en achetant cette maison si l'évaluateur dit qu'elle est en bon état?

P(testOK) = P(testOK|bonEtat) * P(bonEtat) + P(testOK|¬bonEtat) * P(¬bonEtat)
= 0.9 * 0.8 + 0.15 * 0.2 = 0.75
P(bonEtat|testOK) = P(testOK|bonEtat) * P(bonEtat) / P(testOK) = 0.9 * 0.8 / 0.75 = 0.96
EU(acheter|testOK) = P(bonEtat|testOK) Gain(bonEtat) + P(¬bonEtat|testOK) Gain(¬bonEtat)
= 0.96 * (100 000 – 90 000) + (1 - 0.96) * (100 000 – 105 000)= 0.96 * 10000 + 0.04 * -5000 = 9400$

c) Quel est le gain espéré de Bob en achetant cette maison si l'évaluateur dit qu'elle n'est pas en bon état?

P(¬testOK) = 1 – P(testOK) = 1 – 0.75 = 0.25
P(bonEtat|¬testOK) = P(¬testOK|bonEtat) * P(bonEtat) / P(¬testOK) = 0.1 * 0.8 / 0.25 = 0.32
EU(acheter|¬testOK) = P(bonEtat|¬testOK) Gain(bonEtat) + P(¬bonEtat|¬testOK) Gain(¬bonEtat)
= 0.32 * 10000 + (1 – 0.32) * -5000 = -200$

d) Est-ce que ça vaut la peine de faire évaluer la maison. Quelle est la valeur de cette information?

La meilleure action à faire si le test est OK, c'est d'acheter la maison, car l'utilité d'acheter la maison (9400$) est plus grande que l'utilité de ne pas l'acheter (0$). La meilleure action à faire si le test n'est pas OK est de ne pas acheter la maison, parce que l'utilité de ne pas acheter la maison (0$) est plus grande que l'utilité de l'acheter (-200$). La meilleure action si Bob ne fait pas de test est d'acheter, car l'utilité d'acheter (7000$) est plus grande que l'utilité de ne pas acheter (0$).

VPI(test) = (P(testOK) * EU(acheter|testOK) + P(¬testOK) * EU(¬acheter|¬testOK)) – EU(acheter)
= (0.75 * 9400 + 0.25 * 0) – 7000 = 50$

Donc, ça ne vaut pas la peine de faire le test, parce que la valeur de l'information est plus petite que son coût.

6 (18 pts) On considère un environnement avec deux actions possibles : se déplacer vers la gauche ou se déplacer vers la droite. Si l'agent rencontre un mur, il reste sur place. Pour chaque déplacement, le modèle de transition spécifie les probabilités suivantes :

- 0.7 : l'agent se déplace dans la direction désirée.
- 0.2 : L'agent reste sur place.
- 0.1 : L'agent se déplace dans la direction opposée.

Fonction de récompense :

-0.05	-0.05	+1
1	2	3

Politique à l'itération i ($\pi_i(s)$) :

←	←	←
1	2	3

Valeurs des utilités au temps i :

0.8	0.6	0.3
1	2	3

Appliquez à cet environnement l'algorithme modifié d'itération de politique (« policy iteration ») en utilisant la fonction de mise à jour simplifiée de Bellman. **Montrez tous les calculs pour deux itérations.**

Fonction simplifiée de Bellman :

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') U_i(s')$$

Pour l'exemple, le paramètre γ vaut 0.8.

$U_{i+1}(1)$ = -0.05 + 0.8 * (0.9*0.8 + 0.1*0.6) = 0.574
$U_{i+1}(2)$ = -0.05 + 0.8 * (0.7*0.8 + 0.2*0.6 + 0.1*0.3) = 0.518
$U_{i+1}(3)$ = 1 + 0.8 * (0.7*0.6 + 0.3*0.3) = 1.408

Maintenant, il faut calculer la nouvelle politique. Pour l'état 1 :

Action Gauche : (0.9*0.574 + 0.1*0.518) = 0.5684
Action Droite : (0.7*0.518 + 0.3*0.574) = 0.5348

L'action Gauche est donc l'action optimale en 1.

Pour l'état 2 :

Action Gauche : (0.7*0.574 + 0.2*0.518 + 0.1*1.408) = 0.6462
Action Droite : (0.7*1.408 + 0.2*0.518 + 0.1*0.574) = 1.1466

L'action Droite est donc l'action optimale en 2.

Pour l'état 3 :

Action Gauche : (0.7*0.518 + 0.3*1.408) = 0.785
Action Droite : (0.9*1.408 + 0.1*0.518) = 1.319

L'action Droite est donc l'action optimale en 3.

$U_{i+2}(1) = -0.05 + 0.8 * (0.9*0.574 + 0.1*0.518) = 0.40472$
$U_{i+2}(2) = -0.05 + 0.8 * (0.7*1.408 + 0.2*0.518 + 0.1*0.574) = 0.86728$
$U_{i+2}(3) = 1 + 0.8 * (0.9*1.408 + 0.1*0.518) = 2.0552$

Maintenant, il faut calculer la nouvelle politique. Pour l'état 1 :

Action Gauche : (0.9*0.40472 + 0.1*0.86728) = 0.450976
Action Droite : (0.7*0.86728 + 0.3*0.40472) = 0.728512

L'action Droite est donc l'action optimale en 1.

Pour l'état 2 :

Action Gauche : (0.7*0.40472 + 0.2*0.86728 + 0.1*2.0552) = 0.66228
Action Droite : (0.7*2.0552 + 0.2*0.86728 + 0.1*0.40472) = 1.652568

L'action Droite est donc l'action optimale en 2.

Pour l'état 3 :

Action Gauche : (0.7*0.86728 + 0.3*2.0552) = 1.223656
Action Droite : (0.9*2.0552 + 0.1*0.86728) = 1.936408

L'action Droite est donc l'action optimale en 3.