UNIVERSITÉ DE SHERBROOKE
Département d'informatique

IFT 615
Intelligence artificielle

SOLUTIONS

Examen final
Hiver 2009

Le vendredi 24 avril, 9 h à 12 h, au D7-2023

PROFESSEUR
Froduald Kabanza
http://www.dmi.usherb.ca/~kabanza

INSTRUCTIONS

L'examen dure trois heures (9 h à 12 h).

Les notes du cours (copie des présentations), le manuel (livres de référence) et les calculatrices sont autorisés. **Les autres documents et les appareils électroniques sont strictement interdits**.

L'examen comporte cinq questions pour quarante points au total.

Répondez sur ce questionnaire qui sert en même temps de cahier de réponses dans les endroits indiqués.

Des feuilles de brouillon vous sont fournies.

Ne détachez aucune feuille de ce questionnaire.

Écrivez votre nom, prénom et matricule ci-dessous.

NOM : _____ **PRÉNOM :** _____

MATRICULE : _____ **SIGNATURE :** _____

	Q1 / 5	Q2 /10	Q3 / 9	Q4 / 8	Q5 / 8	Total / 40
Note						

Question 1 (5 points) – Unification

Dans ce qui suit, les symboles commençant par un ? sont des variables; sinon ce sont des constantes; *f* et *g* sont des symboles fonctionnels; *p* est un symbole de prédicat. Les prédicats et les fonctions sont en notation préfixée.

a. (1 point) Calculez l'unificateur le plus général (upg) pour les littéraux **(p ?x (f ?x))** et **(p ?x ?y)**. Si l'unificateur n'existe pas, écrivez « n'existe pas » comme réponse.

((?y (f ?x)))

b. (1 point) Calculez l'upg pour les littéraux **(p ?x (f ?x) ?y)** et **(p ?y ?z ?u)**. Si l'unificateur n'existe pas, écrivez « n'existe pas » comme réponse.

((?x ?u) (?z (f ?u)) (?y ?u))

ou *((?y ?x) (?z (f ?x)) (?u ?x))* si on permute les arguments

c. (1 point) Calculez l'upg pour les littéraux **(p (f ?y) ?x)** et **(p ?x ?y)**. Si l'unificateur n'existe pas, écrivez « n'existe pas » comme réponse.

N'existe pas.

d. (1 point) Calculez l'upg pour les littéraux **(p ?x ?z)** et **(p ?z (f ?x))**. Si l'unificateur n'existe pas, écrivez « n'existe pas » comme réponse.

N'existe pas.

e. (1 point) Calculez l'upg pour les littéraux **(p (f ?y) ?z)** et **(p ?z (f ?x))**. Si l'unificateur n'existe pas, écrivez « n'existe pas » comme réponse.

((?z (f ?x)) (?y ?x))

ou *((?z (f ?y)) (?x ?y))* si on permute les arguments.

Question 2 (10 points) – Résolution

Dans la comédie de *Monty Python et le Sacré Graal* (écrite et produite en 1975 par le groupe de comédiens britanniques Monthy Python), le Roi Arthur et Sir Bedevere participent à un débat avec des paysans pour prouver qu'une certaine dame (disons Madame A) est une sorcière. Leur argument tient à une comparaison avec un canard (disons le canard D). Voici en peu de mots en quoi tient l'argumentation :

1. Toute personne faite en bois est une sorcière.
2. Tous les canards sont faits en bois.
3. Toute chose qui pèse la même chose qu'un canard est faite en bois.
4. La dame (A) pèse la même chose que le canard (D).

Naturellement, on prend pour acquis les prémisses suivantes, relevant du sens commun :

5. Le Roi Arthur est une personne.
6. Sir Bedevere est une personne.
7. La dame (A) est une personne.
8. D est un canard
9. Le canard (D) n'est pas une personne.

Utilisez la logique du premier ordre et la résolution pour prouver que la dame est une sorcière. Pour ce faire, vous devez d'abord exprimer l'argumentation précédente en logique du premier ordre, ensuite en forme normale conjonctive. Vous pouvez utiliser seulement la résolution et l'unification pour prouver que la dame est une sorcière. On ne fait pas de distinction entre le féminin et le masculin.

a. (2 points) Exprimez l'argumentation et les prémisses précédentes en logique du premier ordre.

1. $\forall x\ Person(x) \wedge Wood(x) \rightarrow Witch(x)$
2. $\forall x\ Duck(x) \rightarrow Wood(x)$
3. $\forall x,y\ Duck(x) \wedge Equals(Weight(y), Weight(x)) \rightarrow Wood(y)$
4. $Equals(Weight(A), Weight(D))$
5. $Person(Arthur)$
6. $Person(Bedevere)$
7. $Person(A)$
8. $Duck(D)$
9. $\neg\ Person(D)$

Note : pour être précis, il faudrait ajouter un axiome pour la commutativité du prédicat *Equals*. Sinon, s'assurer que le prédicat *Equals* à la ligne 4 a ses arguments dans l'ordre sous-entendu par le prédicat *Equals* à la ligne 3.

b. (3 points) Convertissez les formules de l'étape précédente sous forme normale conjonctive. Ne donnez pas les étapes de conversion. Numérotez chacune des clauses pour pouvoir y faire référence dans la sous-question suivante.

1. $\neg$ *Person* (x_1) $\vee \neg$ *Wood*(x_1) $\vee$ *Witch*(x_1)
2. $\neg$ *Duck*(x_2) $\vee$ *Wood*(x_2)
3. $\neg$ *Duck*(x_3) $\vee \neg$ *Equals*(*Weight*(x_4), *Weight*(x_3)) $\vee$ *Wood*(x_4)
4. *Equals*(*Weight*(*A*), *Weight*(*D*))
5. *Person*(*Arthur*)
6. *Person*(*Bedevere*)
7. *Person*(*A*)
8. *Duck*(*D*)
9. $\neg$ *Person*(*D*)

c. (3 points) Utilisez la preuve par résolution pour prouver que la dame est une sorcière.

Exprimer l'assertion en logique du premier ordre

Witch(*A*)

La nier et la mettre dans une forme clausale

10. $\neg$ *Witch*(*A*)

Effectuer la preuve par résolution :

11. $\neg$ *Duck*(*D*) $\vee$ *Wood*(*A*)	3, 4, {x_3=*D*, x_4=*A*}
12. *Wood*(*A*)	8,11
13. $\neg$ *Person* (*A*) $\vee$ *Witch*(*A*)	1,12 {x_1=*A*}
14. *Witch*(*A*)	7,13
15. *False*	10,14

d. (2 points) Supposons qu'au début on ne sache pas vraiment qui de la dame, de Sir Bedevere ou du canard est la sorcière. Utilisez la preuve par résolution pour trouver qui est la sorcière.

Exprimer en logique du premier ordre l'assertion « Il existe une sorcière »

$\exists x \ Witch(x)$

La nier, la mettre dans une forme clausale et ajouter un littéral pour retenir la trace de l'unification comme réponse.

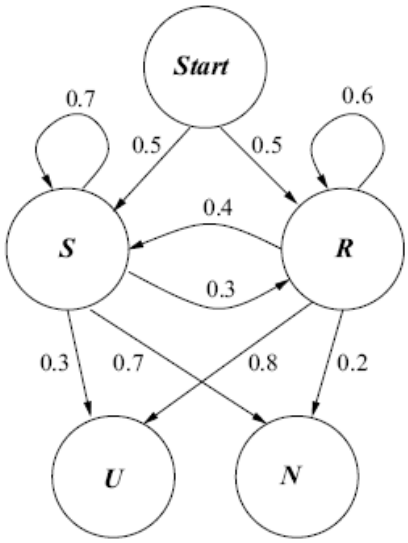
10. $\neg Witch(x_5) \vee Answer(x_5)$

Effectuer la preuve par résolution :

11. $\neg Duck(D) \vee Wood(A)$	3, 4, $\{x_3=A, x_4=D\}$
12. $Wood(A)$	8,11
13. $\neg Person(A) \vee Witch(A)$	1,12 $\{x_1=A\}$
14. $Witch(A)$	7,13
15. $Answer(A)$	10,14 $\{x_5=A\}$

Question 3 (9 points) – Chaînes cachées de Markov

Soit la chaîne cachée de Markov suivante, version simplifiée de la prévision du temps en fonction de l'observation d'un parapluie. C'est un modèle pour un prisonnier qui ne sort jamais, n'a aucun accès à l'extérieur ni aux nouvelles. Sa seule façon de prévoir le temps, c'est en observant si son gardien entre avec un parapluie.



Le temps est la variable cachée et elle est notée *W* (pour « *Weather* »). Elle peut prendre les valeurs *S* (pour « *Sunny* ») ou *R* (pour « *Rainy* »). La variable observée est le parapluie et elle est notée *O*. Elle peut prendre les valeurs *U* (pour « *Umbrella* ») et *N* (pour *no « Umbrella »*). La valeur d'une variable *X* au temps t est notée X_t. On suppose que chaque transition dans la chaîne de Markov dure une unité de temps.

a. (3 points) Calculez la probabilité $P(W_1=S, W_2=S, W_3=R, W_4=S)$

$P(W_1=S, W_2=S, W_3=R, W_4=S) = 0.5*0.7*0.3*0.4=$ ***0.042***

b. (3 points) Calculez la probabilité $P(W_1=R \mid O_1=N)$

$P(W_1=R \mid O_1=N) = P(W_1=R, O_1=N) / P(N)$

$= P(O_1=N \mid W_1=R)*P(W_1=R) / [P(O_1=N \mid W_1=R)*P(W_1=R) + P(O_1=N \mid W_2=S)*P(W_2=S)]$

$= 0.2*05 / [0.2*0.5 + 0.7*0.5]$

$=$ ***0.22***

c. (3 points) Calculez la probabilité $P(O_1=N, O_2=U, W_1=R, W_2=S)$

$P(O_1=N, O_2=U, W_1=R, W_2=S) = P(W_2=S \mid W_1=R) * P(W_1=R) * P(O_1=N \mid W_1=R) * P(O_2=U \mid W_2=S)$

$= 0.4*0.5*0.2*.03$

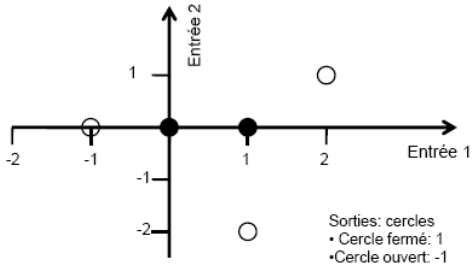
$=$ ***0.012***

Question 4 (8 points) – Réseaux de neurones

d. (3 points) Un perceptron peut-il apprendre à classifier les données suivantes? Expliquez pourquoi.

Entrée 1	Entrée 2	Sortie désirée
-1	0	0
0	0	1
2	1	0
1	0	1
1	-2	0

Non. Les données ne sont pas linéairement séparables.



a. (2 points) Expliquez pourquoi la fonction d'activation sigmoïde est préférable à la fonction signe pour un réseau de neurones à rétropropagation.

L'algorithme à rétropropagation utilise la méthode du gradient descendant pour calculer l'erreur à propager. Pour cette méthode on a besoin d'une fonction continue et dérivable en tout point, telle que la fonction sigmoïde pour pouvoir utiliser l'algorithme à rétropropagation. Or la fonction signe a une discontinuité sur la valeur seuil, ce qui fait qu'elle n'est pas dérivable à ce point.

b. (3 points) Supposons qu'on entraîne un réseau de neurones à rétropropagation avec une fonction sigmoïde et qu'on le laisse itérer suffisamment longtemps afin de minimiser l'erreur sur les données d'entraînement. Supposons qu'on recommence l'entraînement, avec les mêmes données, mais avec des poids initiaux différents. Avons-nous la garantie que le réseau va converger vers la même solution qu'avant (c.-à-d., même ensemble de poids des connexions)? Expliquez votre réponse.

Non, justement parce que l'algorithme à rétropropagation utilise la méthode du gradient descendant jusqu'à atteindre un minimum local pour l'erreur de la sortie. De cette façon, différents minima locaux peuvent être atteints en partant des points initiaux différents.

Question 5 (8 points) – Systèmes experts

Étant donné l'évolution technologique des nouveaux modèles de voitures, Nissan cherche à développer une application pour assister ses mécaniciens. La solution retenue est celle d'un système expert d'aide au diagnostique. Vous faites parti d'un groupe devant concevoir le moteur d'inférence du système expert. Vous avez décidé que ce sera un moteur d'inférence à chaînage avant. Chaque règle sera décrite par une classe **Rule**, ayant la méthode **Rule.precond()** un vecteur de prédicats décrivant la précondition de la règle (une conjonction de prédicats) et la méthode **Rule.postcond** un prédicat décrivant la postcondition de la règle.

Vous disposez d'une implémentation de l'algorithme d'unification vu en classe : **Unification.unify(p1,p2)** prend deux prédicats comme arguments et retourne leur unificateur le plus général. Il a été convenu que la mémoire de travail sera un vecteur de propositions, retourné par la méthode **WorkingMem.get()**.

a. (5 points) En partant de ces spécifications, donnez le pseudocode de la méthode **Conflictset.find()** permettant de calculer le vecteur de règles applicables à une mémoire de travail donnée.

Voir l'exemple du code scheme donné en cours. Plusieurs formulations différentes sont possibles.

b. (3 points) Nommez une technique nettement plus efficace que l'algorithme d'unification vu en classe et couramment utilisé par les systèmes experts. Expliquez brièvement comment elle fonctionne.

Algorithme Rete. Cette technique permet de mémoriser les unifications faites d'une mémoire de travail à une autre durant l'activation itérative des règles (c-à-d., l'exploration de l'espace d'états). De cette façon, des unifications déjà calculées précédemment sont évitées. En fait, on procède par mise à jour de nouvelles unifications.