

**INTELLIGENCE ARTIFICIELLE
IFT615**

Département de mathématiques et d'informatique
Université de Sherbrooke

EXAMEN FINAL, HIVER 2003

Le vendredi 25 avril 2003, 13 h 30 à 16 h 30.

ENSEIGNANT

Froduald Kabanza, PhD
Professeur titulaire

- **Téléphone** : (819) 821 8000 (ext. 2865).
- **Bureau** : D4-1010-20

LOCAUX POUR L'EXAMEN

- **D7-2015**, De Abella à Drapeau
- **D7-2016**, De Drouin P. à Ly
- **D7-2017**, De Montpetit à Thiffault

INSTRUCTIONS

L'examen dure trois heures (13 h 30 à 16 h 30).

Les notes de cours, les manuels, les notes personnelles ainsi que tout appareil électronique sont strictement interdits.

L'examen comporte 30 questions à choix multiples comptant pour 1 point chacune.

Vous devez répondre sur ce questionnaire qui sert en même temps de cahier de réponses, en **encerclant** simplement votre choix pour chaque question. *S'il n'y a aucun choix sur une question, ou s'il y a plusieurs choix, la note sur la question sera zéro.*

Ne détachez aucune feuille de ce questionnaire.

Écrivez votre nom, prénom et matricule ci-dessous.

NOM : _____ **PRÉNOM :** _____

MATRICULE: _____

Dans les questions suivantes A^* désigne « l'algorithme A^* vu dans le cours ». Vous devez supposer que les coûts des transitions sont toujours positifs et que la fonction heuristique retourne toujours une valeur positive. Soit :

- un espace d'états $\{S_0, S_1, \dots, S_9\}$;
- une fonction *successeurs* qui prend un état v comme argument et retourne la liste de ses successeurs avec les coûts des transitions correspondants sous la forme $\{(v_0, c_0), \dots, (v_n, c_n)\}$ où c_i est le coût de la transition (v, v_i) , pour $1 \leq i \leq n$;
- une fonction *but* de test de but, qui prend un état comme argument et retourne « vrai » si cet état est le but (c'est-à-dire l'état qu'on veut atteindre); sinon elle retourne « faux »;
- trois fonctions heuristiques h_1 , h_2 et h_3 ; chacune prend un état comme argument et retourne un nombre réel positif.
- Toutes ces fonctions sont définies dans la table suivante :

État	successeurs	but	h_1	h_2	h_3
S_0	$\{(S_0,1),(S_1,1),(S_8,1)\}$	faux	2	2	2
S_1	$\{(S_3,1),(S_5,1)\}$	faux	6	1	6
S_2	$\{(S_2,1)\}$	vrai	0	0	0
S_3	$\{(S_4,1)\}$	faux	20	0	20
S_4	$\{(S_7,1)\}$	vrai	0	0	0
S_5	$\{(S_6,2)\}$	faux	5	2	5
S_6	$\{(S_7,2)\}$	faux	4	3	4
S_7	$\{(S_4,2)\}$	faux	1	1	1
S_8	$\{(S_9,1)\}$	faux	5	5	2
S_9	$\{(S_0,1),(S_5,1)\}$	faux	4	7	2

1. **d.** Étant donné les fonctions définies dans la table précédente et l'état initial S_0 :
 - a. Seule la fonction heuristique h_1 est admissible.
 - b. Seules les fonctions heuristiques h_1 et h_2 sont admissibles.
 - c. Seule la fonction heuristique h_3 est admissible.
 - d. Aucune des fonctions heuristiques h_1 , h_2 et h_3 n'est admissible.
 - e. Toutes les fonctions heuristiques h_1 , h_2 et h_3 sont admissibles.

2. **e.** Une fonction heuristique est plus efficace qu'une autre si elle donne une solution optimale en ayant exploré une partie de l'espace d'états plus petite ou égale que la partie explorée par l'autre fonction, quelque soit l'état initial. Étant donné les fonctions définies dans la table précédente¹ :
 - a. La fonction heuristique h_1 est plus efficace que la fonction heuristique h_2 .
 - b. La fonction heuristique h_2 est plus efficace que la fonction heuristique h_3 .
 - c. La fonction heuristique h_2 est plus efficace que la fonction heuristique h_1 .
 - d. La fonction heuristique h_3 est plus efficace que la fonction heuristique h_2 .
 - e. Aucune des trois fonctions heuristiques n'est plus efficace que l'autre.

3. **a ou b.**² Étant donné les fonctions définies dans la table précédente, soit la fonction $h_4 = \min(h_1, h_2, h_3)$, c'est-à-dire que, étant donné un état s comme argument, $h_4(s)$ retourne le minimum entre $h_1(s)$, $h_2(s)$ et $h_3(s)$.
 - a. La fonction h_4 est une fonction heuristique et elle est plus efficace que h_1 , h_2 et h_3 .
 - b. La fonction h_4 est une fonction heuristique mais elle est moins efficace que h_1 , h_2 et h_3 .
 - c. La fonction h_4 n'est pas une fonction heuristique parce qu'elle zigzague entre plusieurs valeurs, de sorte qu'elle pourrait causer des instabilités dans A^* .
 - d. Aucune des trois affirmations précédentes n'est correcte.

4. **c.** Étant donné les fonctions *successeurs*, *but* et h_1 dans la table précédente, ainsi que l'état initial S_0 :
 - a. A^* retourne le chemin suivant comme solution $(S_0, S_8, S_9, S_0, S_1, S_3, S_4)$.
 - b. A^* retourne le chemin suivant comme solution (S_0, S_1, S_3, S_4) .
 - c. A^* retourne le chemin suivant comme solution $(S_0, S_1, S_5, S_6, S_7, S_4)$.
 - d. A^* boucle indéfiniment.
 - e. Aucune des quatre affirmations précédentes n'est vraie.

5. **c.** Étant donné les fonctions *successeurs*, *but* et h_3 dans la table précédente, ainsi que l'état initial S_0 :
 - a. A^* retourne le chemin suivant comme solution $(S_0, S_8, S_9, S_5, S_6, S_7, S_4)$.
 - b. A^* retourne le chemin suivant comme solution (S_0, S_1, S_3, S_4) .
 - c. A^* retourne le chemin suivant comme solution $(S_0, S_1, S_5, S_6, S_7, S_4)$.
 - d. A^* boucle indéfiniment.
 - e. Aucune des quatre affirmations précédentes n'est vraie.

¹ Changer la question la prochaine fois car elle est un peu ambiguë. En fait, h_3 est plus efficace que h_2 sauf que h_3 n'est pas admissible.

² Accorder les deux vu que h_4 est admissible alors que les autres ne le sont pas, ce qui prête à confusion. Si les autres étaient tous admissibles, h_4 serait considéré moins efficace selon ma définition précédente; par contre $\max(h_1, h_2, h_3)$ serait plus efficace.

6. **a.** Peu importe le nombre d'états, si on utilise la fonction heuristique qui retourne toujours 0 et des coûts de transitions toujours égaux à 1 :
- A* retourne toujours une solution optimale si elle existe et que les ressources de calcul sont suffisantes.
 - A* retourne toujours une solution si elle existe et que les ressources de calcul sont suffisantes, par contre cette solution pourrait ne pas être optimale.
 - A* risque de boucler.
 - Aucune des trois affirmations précédentes n'est vraie.
7. **a.** Étant donné une fonction heuristique admissible :
- A* retourne toujours éventuellement une solution optimale lorsqu'elle existe et que les ressources de calcul sont suffisantes.
 - A* retourne toujours une solution si elle existe et que les ressources de calcul sont suffisantes, mais cette solution n'est pas nécessairement optimale.
 - A* se comporte comme l'algorithme de Dijkstra utilisé pour la recherche de chemin optimal dans un graphe.
 - A* n'explore jamais le même état deux fois durant la recherche d'une solution.
 - Aucune des quatre affirmations précédentes n'est vraie.
8. **a.** Soit $g(s)$ le coût du chemin le moins coûteux jusque là exploré par A* et partant de l'état initial jusqu'à s (c-à-d., le chemin enregistré par les pointeurs vers les parents); et soit $g^*(s)$ le coût du chemin optimal entre l'état initial et s si on considérait tout l'espace d'états. Alors, durant l'exécution de A* :
- Pour tout état s exploré par A*, on a toujours $g(s) \geq g^*(s)$.
 - Pour tout état s exploré par A*, on a toujours $g(s) \leq g^*(s)$.
 - Pour tout état s exploré par A*, on a toujours $g(s) = g^*(s)$.
 - Aucune des trois affirmations précédentes n'est vraie.
9. **a.** Selon les mêmes définitions de $g(s)$ et de $g^*(s)$ que pour la question précédente :
- Lorsque A* atteint un état s satisfaisant le but, avec une fonction heuristique admissible, on a toujours $g(s) = g^*(s)$.
 - Lorsque A* atteint un état s satisfaisant le but on a toujours $g(s) = 0$.
 - A* ne peut jamais atteindre un état s tel que $g(s) = g^*(s)$.
 - Aucune des trois affirmations précédentes n'est vraie.

10. **c.** Quelque soit la fonction heuristique utilisée, en supposant des ressources de calcul suffisantes:
- Si une solution optimale existe, alors A* termine toujours avec une solution optimale.
 - Si une solution existe, alors A* termine uniquement si la fonction heuristique est admissible.
 - Aucune des deux affirmations précédentes n'est vraie.
 - Toutes les deux affirmations a. et b. sont vraies.

Notons $\{(x_1, t_1), \dots, (x_n, t_n)\}$ la substitution qui consiste à remplacer les variable x_i par les termes t_i dans une formule de logique du premier ordre, pour $1 \leq i \leq n$. Supposons aussi que les variables sont des symboles commençant par « ? ».

11. **c.** Étant donné les littéraux $P(?x, f(?x))$ et $P(a, ?z)$.
- Leur unificateur le plus général est $\{(a, ?x), (f(?x), f(?z))\}$.
 - Leur unificateur le plus général est $\{(?x, a), (?z, f(?x))\}$.
 - Leur unificateur le plus général est $\{(?x, a), (?z, f(a))\}$.
 - Il n'existe pas d'unificateur le plus général pour ces littéraux.
 - Aucune des quatre affirmations précédentes n'est vraie.

12. **d.** Étant donné les littéraux $P(f(?x), f(?x))$ et $P(?y, a)$.
- Leur unificateur le plus général est $\{(?y, f(?x)), (a, f(?x))\}$.
 - Leur unificateur le plus général est $\{(?y, f(?x)), (f(?x), a)\}$.
 - Leur unificateur le plus général est $\{(?y, f(a)), (f(?x), a)\}$.
 - Il n'existe pas d'unificateur le plus général pour ces littéraux.
 - Aucune des quatre affirmations précédentes n'est vraie.

13. **b.** Étant donné les littéraux $P(?x, f(?x))$ et $P(?y, ?z)$.
- Leur unificateur le plus général est $\{(?x, ?y), (?z, f(?x))\}$.
 - Leur unificateur le plus général est $\{(?x, ?y), (?z, f(?y))\}$.
 - Leur unificateur le plus général est $\{(?x, ?y), (f(?x), ?z)\}$.
 - Il n'existe pas d'unificateur le plus général pour ces littéraux.
 - Aucune des quatre affirmations précédentes n'est vraie.

14. **a.** La composition des substitutions $\{(?x, a), (?y, ?b), (?z, c)\}$ et $\{(?x, f(?y)), (?y, ?z)\}$ est :
- $\{(?x, a), (?y, b), (?z, c)\}$.
 - $\{(?x, f(b)), (?y, c), (?z, c)\}$.
 - $\{(?x, f(?y)), (?y, ?z), (?x, a), (?y, b), (?z, c)\}$.
 - Il est impossible de composer les deux substitutions.
 - Aucune des quatre affirmations précédentes n'est vraie.

15. **b.** La composition des substitutions $\{(?x, f(?y)), (?y, ?z)\}$ et $\{(?x, a), (?y, b), (?z, c)\}$ est :
- $\{(?x, a), (?y, b), (?z, c)\}$.
 - $\{(?x, f(b)), (?y, c), (?z, c)\}$.
 - $\{(?x, f(?y)), (?y, ?z), (?x, a), (?y, b), (?z, c)\}$.
 - Il est impossible de composer les deux substitutions.
 - Aucune des quatre affirmations précédentes n'est vraie.
16. **b.** Étant donné les clauses $\neg r(?z, ?x) \vee p(?x, ?y, ?z) \vee q(a)$ et $r(a, b) \vee s(?x)$, leur résolvant obtenu en appliquant la règle de résolution est :
- La clause vide.
 - $p(b, ?y, a) \vee q(a) \vee s(b)$.
 - $p(b, ?y, a) \vee r(a, b) \vee q(a) \vee s(b)$.
 - $\neg p(b, ?y, a) \vee \neg q(a) \vee s(b)$.
 - Il n'existe pas de résolvant pour les deux clauses.
 - Aucune des cinq affirmations précédentes n'est vraie.
17. **c.** Étant donné les clauses $\neg p(?x) \vee r(?x)$ et $\neg q(?u) \vee p(?u)$, leur résolvant obtenu en appliquant la règle de résolution est :
- La clause vide.
 - $q(?u) \vee r(?u)$.
 - $\neg q(?u) \vee r(?u)$.
 - Il n'existe pas de résolvant pour les deux clauses.
 - Aucune des cinq affirmations précédentes n'est vraie.
18. **b.** La base de connaissance d'un système expert :
- Décrit le moteur d'inférence de base du système expert.
 - Décrit l'expertise utilisée par le moteur d'inférence.
 - Est constituée de la fonction A* et de l'unification.
 - Aucune des trois affirmations précédentes n'est correcte.
 - Toutes les trois affirmations de a. à c. sont correctes.

19. **c.** L'algorithme A^* peut être utilisé pour implémenter:
- Un moteur d'inférence à chaînage arrière seulement.
 - Un moteur d'inférence à chaînage avant seulement.
 - Aussi bien un moteur d'inférence à chaînage arrière qu'un moteur d'inférence à chaînage avant.
 - Aucune des trois affirmations précédentes n'est correcte.
20. **a.** L'algorithme d'apprentissage symbolique ID3
- Est dans la catégorie des algorithmes d'apprentissage supervisé.
 - Est dans la catégorie des algorithmes d'apprentissage non supervisé.
 - Aucune des deux affirmations précédentes n'est correcte.
21. **b.** L'algorithme d'apprentissage symbolique ID3
- Procède par élimination de candidats.
 - Produit un arbre de décision à partir d'une base de données.
 - Utilise un système expert de prise de décision.
 - Aucune des trois affirmations précédentes n'est correcte.
22. **d.** L'algorithme d'apprentissage symbolique ID3
- A besoin des opérations de généralisation définies par l'utilisateur.
 - A besoin des opérations de spécialisation définies par l'utilisateur.
 - Les deux affirmations précédentes sont vraies.
 - Aucune des affirmations a. et b. n'est vraie.
23. **a.** Un réseau de transition augmenté
- Peut reconnaître des langages sensibles au contexte.
 - Ne reconnaît que des langages hors-contexte.
 - Reconnaît seulement les langages réguliers vu que c'est un automate.
 - Aucune des trois affirmations précédentes n'est correcte.
24. **c.** Dans un réseau de transition augmenté :
- Les états représentent les symboles terminaux de la grammaire et les transitions représentent les symboles non terminaux.
 - Les états représentent les symboles non terminaux de la grammaire et les transitions les symboles terminaux.
 - Les transitions représentent les symboles non terminaux et parfois les symboles terminaux.
 - Aucune des trois affirmations précédentes n'est correcte.

25. **b.** Le domaine de l'intelligence artificielle :
- A comme unique objectif de programmer des systèmes experts.
 - Vise à construire des systèmes qui agissent comme des humains, raisonnent comme des humains, agissent rationnellement ou raisonnent rationnellement.
 - Aucune des deux affirmations précédentes n'est correcte.
26. **a.** Le problème de planification déterministe consiste à :
- Trouver une séquence d'actions ou de transitions qui mène d'un état initial à un état final.
 - Calculer un arbre de décision optimal à partir d'une base de données.
 - Propager les signaux dans un réseau de neurones avec une architecture de Hopfield.
 - Aucune des affirmations précédentes n'est vraie.
 - Toutes les affirmations de a. à c sont vraies.
27. **b.** Choisissez l'affirmation la moins fausse ou la plus correcte. Un algorithme génétique :
- Est un algorithme qui utilise l'ADN pour trouver des solutions.
 - Est un algorithme qui utilise une recherche aléatoire inspirée des opérations de sélection, de croisement et de mutation pour trouver une solution.
 - Est un algorithme ayant un code génétique pour des problèmes complexes.
 - Est un algorithme qui combine la bio-informatique et la physique quantique.
28. **a.** Choisissez l'affirmation la moins fausse ou la plus correcte. La différence principale entre un algorithme génétique et un programme génétique est que :
- L'algorithme génétique opère sur des chaînes de bits alors qu'un programme génétique opère sur des programmes.
 - Il n'y a pas de différence entre les deux, il s'agit de deux terminologies différentes pour dire la même chose.
 - L'algorithme génétique est toujours plus rapide.
29. **a.** Choisissez l'affirmation la moins fausse ou la plus correcte.
- Un réseau de neurones de type *perceptron* à une couche ne peut apprendre à classifier des données non linéairement séparables.
 - Un réseau de neurones de type *perceptron* à une couche peut apprendre à classifier des données non linéairement séparables.
 - Un réseau de neurones de type *perceptron* à une couche peut apprendre à classifier des données non linéairement séparables à condition qu'on utilise au moins deux neurones.
 - Aucune des trois dernières affirmations n'est vraie.
 - Toutes les trois affirmations a. à c. sont vraies.

30. **a.** Choisissez l'affirmation la moins fausse ou la plus correcte.

- a. Un réseau de neurones peut être combiné avec un algorithme génétique ou un système expert floue.
- b. Un réseau de neurones ne peut être combiné avec un algorithme génétique parce que les deux sont incompatibles.
- c. Aucune des deux affirmations précédentes n'est vraie.