

L3 informatique, L3 mathématiques

Examen

Unité M.MIM5E2 : Aide à la décision et intelligence artificielle
2 h — Tous documents autorisés

Consignes

- Il est vivement conseillé de composer en utilisant un logiciel de traitement de texte. Cependant, vous pouvez aussi composer en écrivant à la main sur des feuilles blanches. Dans ce cas, vous disposerez à l'issue de chaque épreuve de 2 minutes pour scanner (ou photographier) numériquement votre production écrite.
- Écrivez au début du fichier que vous rendez : votre nom, votre prénom et votre numéro d'étudiant.
- Pour chaque réponse, indiquez la partie et la question traitée.
- A la fin de l'examen, téléversez (i.e. upload) votre fichier à l'emplacement « Épreuve du 22 juin 2020 » sous eCampus. Remarques :
 - cet envoi doit être fait impérativement AVANT 15 h 30
 - conservez une COPIE de votre fichier et vérifiez bien que le fichier envoyé est le BON.

Note Pour représenter un arbre, on pourra utiliser un format textuel tel que

```

A
  B
    C
    D
  E

```

(pour représenter un arbre de racine A, ayant pour fils les nœuds B et E, B ayant pour fils les nœuds C et D).

Barème Le barème est sur 22 points. Si vous obtenez par exemple 15 points sur les 22, vous aurez 15/20, et si vous obtenez par exemple 21 points, vous aurez 20/20.

1 Contraintes (9 points)

On considère le CSP binaire discret $P = (X, D, C)$ suivant :

- $X = \{X1, X2, X3, X4\}$
- $D(X1) = D(X2) = \{a, b\}; D(X3) = \{b, c\}; D(X4) = \{a, c\}$
- $C = \{ \begin{array}{l} C1 : X1 \neq X2 \\ C2 : X2 \neq X3 \\ C3 : X1 \neq X4 \\ C4 : X3 \neq X4 \end{array} \}$

On s'intéresse à la résolution du problème P afin de trouver des solutions qui satisfont toutes les contraintes du CSP. Pour cela, on utilise un arbre de décision où chaque niveau de l'arbre représente une variable choisie, et chaque nœud d'un niveau représente une valeur choisie pour la variable concernée.

On choisit l'ordre statique $X1, X2, X3$ puis $X4$ d'instanciation des variables et l'ordre statique a, b puis c de choix des valeurs du domaine pour chacune des variables.

Question 1 (2 points). *Donnez l'arbre de recherche de solution sur ce problème en utilisant l'algorithme Générer et Tester.*

Question 2 (2 points). *Même question avec l'algorithme Backtrack.*

Question 3 (1 point). *Que peut-on conclure des réponses aux questions 1 et 2 ?*

On souhaite désormais utiliser des techniques de propagation de contraintes pour améliorer l'efficacité de la résolution.

Question 4 (2 points). *Appliquez l'arc-consistance sur le CSP P en précisant tous les tests effectués et le statut des valeurs.*

Question 5 (1 point). *Pour ce problème, l'arc-consistance peut-elle améliorer l'efficacité de la résolution ? Justifiez.*

Question 6 (1 point). *Que peut-on conclure des réponses aux questions 4 et 5 sur l'arc-consistance ?*

2 Fouille de données (6 points)

On considère le jeu de données \mathbf{D} représenté à la table 1. A, B, C et D sont appelés *items* et une *transaction* est une ligne de \mathbf{D} .

	Items			
1	A		C	
2	A	B	C	
3		B	C	D
4		B		D
5	A	B	C	

TABLE 1 – Jeu de données \mathbf{D} .

Question 7 (1,5 point). *À partir de \mathbf{D} , illustrez le fonctionnement de l'extraction des motifs fréquents selon l'algorithme **Apriori** (on choisira $\text{minsup} = 2$). Précisez les motifs fréquents au seuil $\text{minsup} = 2$ ainsi que les motifs non fréquents et testés par **Apriori**.*

Question 8 (2 points). *Rappelez le principe clé utilisé par les algorithmes d'extraction de motifs fréquents pour éviter de parcourir tout l'espace de recherche. Illustrez ce principe en vous appuyant sur l'exemple de \mathbf{D} . Précisez deux conditions d'élagage de l'espace de recherche qui découlent de ce principe.*

Question 9 (1 point). *Donnez toutes les règles d'association fréquentes et non triviales constituées d'au moins 3 items et extraites de \mathbf{D} .*

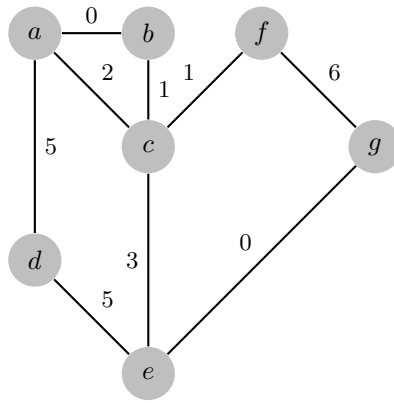
Question 10 (0,5 point). *L'algorithme **Apriori** est-il de nature heuristique ou produit-il un résultat complet ? Justifiez.*

Question 11 (0,5 point). *Une fois les paramètres fixés par l'utilisateur, l'algorithme **Apriori** donne-t-il les meilleures règles ou toutes les règles vérifiant les valeurs de ces paramètres ? Justifiez.*

Question 12 (0,5 point). *On suppose maintenant que \mathbf{D} contient 1024 transactions. Comment varie l'espace de recherche des motifs fréquents par rapport à l'état initial de \mathbf{D} (c'est-à-dire l'état donné à la table 1) ?*

3 Planification (7 points)

Considérons le graphe suivant dont les nœuds représentent les états d'un système et les arêtes les coûts de transition d'un état à l'autre par l'intermédiaire d'une action.



Question 13 (3 points). Appliquez l'algorithme de *Dijkstra* pour calculer le plus court chemin de *a* à *g*. Indiquez explicitement chaque étape de l'algorithme et la valeur courante de toutes les variables (liste des ouverts, table des distances, table des pères, nœud courant).

Le taquin est un jeu solitaire en forme de damier. Il est composé de 15 petits carreaux numérotés de 1 à 15 qui glissent dans un cadre prévu pour 16. Le jeu consiste à remettre dans l'ordre les 15 carreaux (comme par exemple sur la figure ¹ 1.A) à partir d'une configuration initiale quelconque (par exemple celle donnée en figure 1.B). Supposons que nous voulons résoudre le jeu avec l'algorithme A^* , que nous choisissons comme heuristique h_1 , qui associe à chaque état le nombre de carreaux mal placés, et supposons que le coût de n'importe quelle action est égal à 1.

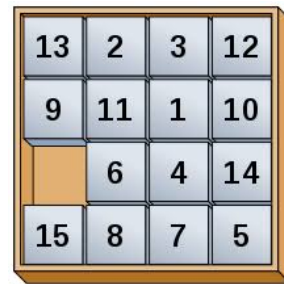


FIGURE 1 – (A) un état final; (B) un état quelconque

Question 14 (2 points). Démontrez que h_1 est une heuristique admissible.

Question 15 (2 points). Donnez (et démontrez) une heuristique plus informée que h_1 .

1. Figure tirée de Wikipedia.