

Examen final IA02/AI27 : résolution de problèmes, logique et jeux – Printemps 2023

Préliminaires

- **Durée** : 2h00.
- **Seul document autorisé** : deux feuilles A4 recto verso **manuscrites**, c.-à.-d non imprimées (aménagements possibles pour certains 1/3 temps qui en ont préalablement fait la demande).
- **Barème indicatif**, susceptible de changement sans préavis : 3 + 4 + 4,5 + 8 (+ 0,5 point pour la clarté et la présentation de la copie).
- L'élégance, la lisibilité et la concision des solutions fournies font partie **intégrante** des critères d'évaluation.
- Toute ambiguïté, erreur ou omission dans le sujet devra être résolue par le candidat et par lui seul.
- L'**Exercice 2** est réservé aux étudiant·e·s sous statut d'étudiant·e et l'**Exercice 2bis** aux étudiant·e·s sous statut d'apprenti·e·s. Tous les autres exercices sont à rendre pour tout le monde.
- Organisation des copies, à suivre **SCRUPULEUSEMENT** (des points pourront être retirés en cas de non-respect)
 - FISE : Exercice 1 et 2 sur une même copie, Exercice 3 sur une copie à part, Problème sur une copie à part.
 - FISA : Exercice 1 sur une copie à part, Exercice 2bis et Exercice 3 sur une même copie, Problème sur une copie à part.

Exercice 1 – Question de cours : recherche informée

Qu'appelle-t-on algorithmes de recherche informée ? Sur quoi se basent-ils ? En quoi peut-on dire que les algorithmes gloutons et A* sont des algorithmes de recherche informée ? Comment fonctionnent-ils et en quoi diffèrent-ils ? Quelle(s) structure(s) de données vous semble(nt) la/les plus pertinente(s) pour appliquer ces algorithmes ? Quand dit-on qu'une heuristique est admissible ? Quel rapport avec les 2 algorithmes précédemment présentés ? Quel lien existe-t-il avec l'algorithme de Dijkstra ?

20 lignes maximum.

Exercice 2 – Prolog : bègues et hamburgers (FISE)

On demande dans cet exercice de ne pas utiliser de fonctions prédéfinies.

1. Écrire un prédicat `hamburger(+L1, -L2)` qui, à partir de la liste de `n` ingrédients `L1` unifie `L2` avec la liste des `n` hamburgers utilisant les ingrédients de `L1`.

```
?- hamburger([poulet, boeuf, poisson], L).
L = [hamb(pain, poulet, pain), hamb(pain, boeuf, pain),
     hamb(pain, poisson, pain)]
```

2. Écrire le prédicat `begaie(+L1, -L2)` qui, étant donné une liste `L1`, unifie à `L2` la liste `L1` dont tous les éléments sont répétés 2 fois de suite.

```
?- begaie([a, b, c, d, e, f], L).
L = [a, a, b, b, c, c, d, d, e, e, f, f]
```

3. Écrire le prédicat `repete(+X, +N, -L)` qui unifie la liste `L` avec la liste composée de l'élément `X` répété `N` fois

```
?- repete(a, 4, L).
L = [a, a, a, a]
```

4. Écrire le prédicat `concat(+L1, +L2, -L3)` qui unifie à `L3` la concaténation de `L1` et `L2`.

```
?- concat([a,b,c], [d,e,f], L).
L = [a, b, c, d, e, f]
```

5. Écrire le prédicat `begaie_n(+L1, +N, -L2)` qui unifie avec `L2` les éléments de la liste `L1` répétés `N` fois

```
?- begaie_n([a, b, c], 4, L).
L = [a, a, a, a, b, b, b, b, c, c, c, c]
```

Exercice 2bis – STRIPS et peinture (FISA)

On se place dans une pièce que l'on aimerait repeindre. À l'état initial, on a une échelle et des pinceaux, mais il nous manque de la peinture. D'autre part, la lampe (qui est au sol) ne fonctionne pas. Bien entendu, on ne peut pas peindre sans lumière, et on n'a pas d'ampoule de rechange. À l'état final, on voudrait avoir peint et avoir de la lumière dans cette pièce.

Questions

1. Donner en STRIPS les fluants, les prédicats, l'état initial et l'état final de ce problème.
2. Décrire les actions `Peindre`, `Changer-Ampoule`, `Allumer`, `Obtenir-Peinture` et `Obtenir-ampoule`, toujours en STRIPS.
3. Donner un plan permettant de passer de l'état initial à l'état final.

D'après : Bruno Bouzi, *Introduction à la planification*.

Exercice 3 – Logique du premier ordre : le retour du chat Bichou

On considère le problème suivant.

1. Tous les chats du quartier mangent dans le plat de mon chien.
2. Aucun chat roux ne peut être autrement que rusé.
3. Le chat Bichou n'a jamais eu de panier.
4. Les compagnons d'errance de mon chien aiment tous les os à moelle.
5. Seuls les chats du quartier sont rusés.
6. Seuls ses compagnons d'errance mangent dans le plat de mon chien.
7. Les chats qui ne sont pas roux ont tous un panier.

Questions

On souhaite modéliser ce problème en logique du premier ordre. On utilisera les concepts `chat/1`, `du_quartier/1`, `roux/1`, `rusé/1`, `possède_panier/1`, `aime_les_os_à_moelle/1`, ainsi que les relations `mange_dans_le_plat_de/2`, `compagnon_errance/2` et les constantes `bichou` et `mon_chien`.

- 1. Modéliser chacune des phrases en logique du premier ordre.
- 2. Passer chaque formule sous forme normale (CNF), via par exemple un passage sous forme prénexe et sous forme de Skolem.
- 3. Montrer, en utilisant les principes de résolution et de réfutation, que Bichou aime les os à moelle.

Problème – Le jeu du Chomp

N.B. Les parties 2 et 3 sont indépendantes.

Préliminaires



/File:Chomp_game.png
Par Lord Belbury (//commons.wikimedia.org/wiki/User:Lord_Belbury) — Travail personnel, CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0).

D’après : [https://fr.wikipedia.org/wiki/Chomp_\(jeu\)](https://fr.wikipedia.org/wiki/Chomp_(jeu)) (https://fr.wikipedia.org/wiki/Chomp_(jeu))

Le jeu du Chomp est un jeu à deux joueurs, joué sur un rectangle composé de blocs carrés représentés comme une tablette de chocolat. Chaque joueur choisit un carré à tour de rôle, et le mange, ainsi que tous les carrés situés à sa droite ou plus bas. Le carré en haut à gauche est empoisonné et celui qui le mange perd la partie.

Exemple



Le joueur A commence. On notera les coordonnées (ligne, colonne) en partant de (1,1) pour le carré empoisonné, en haut à gauche. Il choisit le carré en (3, 5) et mange les deux carrés en bas à droite, en (3, 5) et (4, 5). Le joueur B choisit le carré en (4, 2) et mange alors 3 carrés, (4, 2), (4, 3) et (4, 4). Puis le joueur A choisit le carré en (1, 2) et mange 11 carrés. Le joueur B choisit le carré en (2, 1) et mange trois carrés de la seule colonne restante. Enfin le joueur A est obligé de manger le seul carré restant, celui en (1, 1) qui est empoisonné. Le joueur A perd donc la partie.

Sauf mention contraire, on se restreindra dans la suite de ce problème à une tablette de

chocolat de taille 2×2.



Partie 1 : Modélisation du jeu

- 1. Comment modéliser un état de jeu ? Quel est l’état de départ ?
- 2. Énumérer toutes les actions possibles au premier coup.
- 3. Donner une fonction de gain possible. Dans quel cas peut-on obtenir un match nul ?

Partie 2 : Fonction de hachage de Zobrist pour le jeu du Chomp

On souhaite utiliser la fonction de hachage de Zobrist pour stocker un état du jeu.

On considère pour cela la suite aléatoire de nombres [3, 8, 1, 5].

- 1. Donner la décomposition binaire de chacun de ces nombres. Combien de bits sont nécessaires ?
- 2. Combien existe-t-il d’états possibles différents dans le jeu du Chomp 2×2 ? Les énumérer sous forme de petits schémas.
- 3. Pour chacun de ces états de jeu, donner une valeur de hachage de Zobrist. Donner les détails de votre modélisation et de vos calculs.

Partie 3 : Résolution du jeu

- 1. Appliquer l’algorithme min-max en classant les coups (le carré croqué) les plus en bas et les plus à droite (c.-à-d. priorité au numéro de ligne le plus élevé puis au numéro de colonne le plus élevée). Dessiner l’arbre entier. Que peut-on en déduire ?
- 2. Appliquer l’algorithme alpha-beta en utilisant le même ordre de coups. Redessiner l’arbre en entier en donnant les valeurs successives des variables `value`, `alpha` et `beta`.
- 3. Proposer une fonction d’évaluation pour le jeu du Chomp en taille nxn. Justifier le choix fait.