

Rappels : Aucun document n'est autorisé. Le barème n'est donné qu'à titre indicatif. Une annexe contenant des rappels de cours est donnée en fin de sujet. **Ce sujet comporte 5 pages.**

Les 2 parties A et B doivent être rédigées sur des copies séparées.

Attention : la partie B est plus longue que la partie A.

Partie A

Exercice 1 *Jeu à 2 joueurs (6 points)*

On considère l'arbre de recherche (noté ADR) donné dans le tableau suivant :

Positions	A	B	C	D	E	F	G
Filles	B ; C ; D	E ; F ; 12	8 ; G ; 5 ; 15	14 ; H ; J	7 ; -2	K ; L	10 ; 3
Positions	H	J	K	L	M	N	P
Filles	9 ; M ; 16	17 ; N ; 20	10 ; 9 ; 12	4 ; 3	6 ; P ; 12	2 ; 5	7 ; 0

La racine de cet arbre est la position A. Les positions filles de A (atteignables en 1 coup) sont les positions B, C et D, et ainsi de suite. Les positions terminales (feuilles) de ADR n'ont pas de nom, pour elles, on ne donne que le résultat de la fonction d'évaluation. Les positions filles d'une position sont examinées dans l'ordre donné (B avant C, etc.). Le camp qui doit jouer dans la position A est le camp AMI (qui cherche maximiser la fonction d'évaluation).

1. Donner une évaluation de la racine A de ADR en utilisant l'algorithme alpha-bêta. Les positions examinées doivent être détaillées, dans l'ordre dans lequel elles sont examinées. Pour chaque branche coupée, le type de coupe (type α ou type β) doit être indiqué et justifié.
2. Par rapport à l'application de l'algorithme minimax, quel est le gain, en nombre de positions terminales examinées, de l'application de l'algorithme alpha-bêta sur ADR ?
3. L'efficacité de l'algorithme alpha-bêta dépend de l'ordre d'examen des feuilles de l'arbre sur lequel on l'applique. Expliquez, en justifiant votre réponse, dans quel(s) cas l'algorithme alpha-bêta peut être le plus efficace possible dans l'examen de ADR.

Exercice 2 *Base de règles (5 points)*

On considère un club écossais très privé, dont tout membre doit vérifier les règles sévères ci-dessous :

- R_1 s'il joue de la cornemuse, alors il porte un kilt
- R_2 s'il est roux, alors il porte un kilt
- R_3 s'il joue de la cornemuse, alors il est fan de rugby
- R_4 s'il porte un kilt, alors il a un accent épouvantable
- R_5 s'il connaît par cœur les poèmes de Robert Burns, alors il sait préparer le haggis¹
- R_6 si son nom est de la forme MacQuelqueChose et s'il connaît par cœur les poèmes de Robert Burns, alors il a déjà vu le monstre du Loch Ness
- R_7 s'il est roux et si son nom est de la forme MacQuelqueChose, alors il joue de la cornemuse
- R_8 s'il a un accent épouvantable et s'il boit du whisky, alors il a déjà vu le monstre du Loch Ness
- R_9 s'il joue de la cornemuse et s'il habite un manoir hanté, alors il connaît par cœur les poèmes de Robert Burns
- R_{10} s'il sait préparer le haggis et s'il est fan de rugby, alors il boit du whisky

1. Quel ordre de langage peut-on considérer pour représenter ces règles ?

1. Panse de brebis farcie, plat traditionnel écossais

2. On considère le membre James MacPherson qui est roux et sait préparer le haggis.
En choisissant le mode de chaînage le plus approprié et en justifiant formellement les réponses,
 - (a) prouver qu'il a déjà vu le monstre du Loch Ness.
 - (b) indiquer combien de preuves peuvent être établies.
 - (c) indiquer le nombre minimal d'inférences à réaliser.
 On considère que les règles sont choisies dans l'ordre de la base de règles.
3. [Question indépendante] Quelle est la différence principale entre les moteurs d'inférence de Prolog et de clips ?

Partie B

Exercice 3 Problème : choix d'une pizza (8 points)

Trois amis, Laure, Marie et Nicolas veulent commander une pizza à se partager. La pizzeria propose des pizzas utilisant 4 sortes de garnitures : des champignons, des poivrons, du jambon ou de l'œuf. Marie pose un veto sur toute pizza avec des champignons. Nicolas ne veut pas d'œuf et veut qu'il y ait du jambon dans la pizza. Laure, elle, n'apprécie le jambon que s'il est accompagné de champignons, elle s'oppose donc aux pizzas contenant du jambon sans contenir de champignons.

1. **SAT.** Peut-on composer une pizza qui satisfasse tout le monde ?
 - (a) Répondre à cette question en traduisant le problème en SAT, en utilisant les variables propositionnelles suivantes :
 - c, p, j, o représentent les ingrédients présents dans la pizza (respectivement champignons, poivrons, jambon et œuf). c est vrai si la pizza contient des champignons et faux sinon.
 - v_L, v_M, v_N représentent les veto de respectivement Laure, Marie et Nicolas. v_L est vrai si Laure met un veto sur la pizza proposée et faux si la pizza lui convient.
 - (b) On relâche le problème en demandant une pizza qui satisfasse au moins 2 personnes. Exprimer cette condition avec une contrainte de cardinalité, puis la traduire en SAT avec l'encodage de votre choix. Il n'est pas demandé de résoudre ce nouveau problème SAT.
2. **ASP.**

La pizzeria propose 3 recettes au choix : champignon-jambon (dite "reine"), poivron-œuf (dite "végétarienne") ou jambon-œuf (dite "fermière"). Peut-on commander 2 pizzas parmi celles-ci de sorte que chacun apprécie au moins une des pizzas commandées ?

On donne le début de programme ASP suivant, où c, j, p, o correspondent comme précédemment aux ingrédients :

```
ing(c;j;p;o).
pizza(reine;vege;ferm).
client(laure; marie; nicolas).
num(1..2).
recette(reine,c). recette(reine,j).
recette(vege,p). recette(vege,o).
recette(ferm,j). recette(ferm,o).
```

Note : si vous utilisez des domaines, il est exigé de donner leur déclaration (par exemple `#domain pizza(Piz;Piz1). #domain client(Cl1).`).

- (a) On veut générer le prédicat `choix(N,P)` pour indiquer les pizzas choisies, où $N=1$ ou 2 et P est une pizza. Par exemple `choix(1,ferm),choix(2,reine)` indique qu'on choisit une pizza reine et une pizza fermière, tandis que `choix(1,vege),choix(2,vege)` indiquerait que l'on choisit deux pizzas végétariennes. Donner en ASP le morceau de programme permettant de générer ce choix.
- (b) On utilise le prédicat `veto(C,P)` pour indiquer que le client C pose un veto sur la pizza P . Ecrire les règles ASP permettant de définir les vetos de Laure, Marie et Nicolas.

- (c) Indiquer comment définir un prédicat `satisfait(C)` indiquant que le choix de pizza fait satisfait le client *C* (*i.e.* le choix contient au moins une pizza sur laquelle il n'a pas posé de veto).
- (d) Donner une contrainte d'intégrité permet de filtrer les answer sets afin que le programme donne la réponse à la question (choisir 2 pizzas de façon à ce que tout le monde soit satisfait).

Exercice 4 *ASP (2 points)*

Déterminer les answer sets des programmes suivants :

$$\Pi_A \begin{cases} a \leftarrow b, \text{not } c. \\ c \leftarrow b, \text{not } a. \\ b \leftarrow d, \text{not } e. \\ d. \end{cases} \quad \Pi_B \begin{cases} a \leftarrow \text{not } b. \\ b \leftarrow \text{not } c. \\ c \leftarrow \text{not } a. \end{cases}$$

Exercice 5 *SAT - CDCL (3 points)*

On considère la théorie suivante :

$$\begin{array}{lll} \phi_1 = x_1 \vee x_5 & \phi_5 = \neg x_2 \vee \neg x_9 \vee \neg x_{12} & \phi_9 = x_8 \vee \neg x_9 \\ \phi_2 = x_1 \vee \neg x_2 \vee \neg x_4 \vee \neg x_7 & \phi_6 = x_3 \vee x_7 & \phi_{10} = \neg x_8 \vee x_{10} \vee \neg x_{11} \\ \phi_3 = \neg x_1 \vee x_3 \vee x_6 & \phi_7 = x_4 \vee x_5 & \phi_{11} = \neg x_9 \vee \neg x_{10} \\ \phi_4 = x_2 \vee \neg x_5 & \phi_8 = x_4 \vee x_6 \vee \neg x_7 \vee x_9 & \phi_{12} = x_{10} \vee x_{11} \vee x_{12} \end{array}$$

L'heuristique de choix de variables est initialement uniquement basée sur l'ordre de variables : x_1 est choisi en premier, puis x_2 , etc... On teste toujours en premier la valeur faux.

1. Dérouler les choix et propagations unitaires suivant l'heuristique jusqu'à arriver à un conflit. Indiquer à chaque niveau de décision la variable choisie et les variables dérivées par propagation unitaire (en indiquant pour chacune la clause dont elle est dérivée).
2. Analyser le conflit pour déterminer le premier point d'implication unique (FUIP). Vous pouvez utiliser au choix la méthode de résolution ou le graphe d'implication.
3. Indiquer la clause apprise et comment effectuer le backjump (retour arrière non synchrone) qui s'ensuit, puis dérouler le reste de l'algorithme (sans changer l'heuristique) jusqu'à la prochaine découverte d'un modèle ou d'un conflit. Peut-on conclure sur la satisfiabilité de cet ensemble de clauses à ce stade ?