

Médian IA02 : modélisation de problèmes en logique – Printemps 2023

Préliminaires

- **Durée** : 1h30 (+ éventuel 1/3 temps).
- **Seul document autorisé** : une feuille A4 recto verso **manuscrite** (aménagements possibles pour certains 1/3 temps qui en ont préalablement fait la demande).
- Les machines, en particulier les téléphones portables, sont **interdites** (sauf aménagements prévus).
- **Barème indicatif**, susceptible de changement sans préavis : 5 + 4,5 + 5 + 5 (+ 0,5 point pour la clarté et la présentation de la copie).
- L'élégance des solutions fournies fait partie intégrante des critères d'évaluation.
- Toute ambiguïté, erreur ou omission dans le sujet devra être résolue par le candidat et par lui seul.

Exercice 1 – Prolog, négation par l'échec et raisonnement non monotone

On considère le programme **Prolog** suivant.

```
vole(X) :- oiseau(X).
vole(X) :- chauve_souris(X).
vole(X) :- avion(X).

oiseau(X) :- canari(X).
mammifere(X) :- chauve_souris(X).
vivipare(X) :- mammifere(X).

canari(titi).
chauve_souris(bruce).
avion(dusty).
```

Ainsi que la requête :

```
?- vole(titi), vole(bruce).
```

Questions

1. Traduire en langage naturel (et pas en pseudo-mathématico-charabia) les différentes règles et faits.
2. Traduire en clauses de la logique du premier ordre les différentes règles, les faits, ainsi que la requête.
3. Montrer, en utilisant uniquement les principes de réfutation et de résolution, que la requête est vraie.
4. Qu'est-ce que la négation par l'échec ? Quelle différence avec la négation logique ? En quoi peut-on dire que l'utilisation de la négation par l'échec induit une conséquence logique non monotone ?
5. On souhaite ajouter l'information « Ce qui vole et qui n'est pas vivipare pond des œufs ». Ajouter la règle correspondante en **Prolog**. De **titi**, de **bruce**, et de **dusty**, qui pond des œufs ? Pourquoi ?

Exercice 2 - Dimacs

Considérons le fichier DIMACS suivant.

```
p cnf 4 7
-1 -2 0
-2 -3 0
-3 -4 0
1 2 3 4 0
-4 -1 0
-1 -3 0
-2 -4 0
```

1. Que veut dire la première ligne ?
2. Écrire la formule propositionnelle associée à ce fichier. Utiliser des noms de variables de type X_i et les mettre dans l'ordre.
3. Combien de modèles cette formule admet-elle ? Le montrer en utilisant l'algorithme de Quine.
4. Le résultat était-il attendu ? Que pouvait-on reconnaître ?

Exercice 3 - « Pas taper » le chat et logique du premier ordre

Soit les indices suivants :

1. Tous ceux qui aiment tous les animaux sont aimés par quelqu'un.
2. Quiconque frappe un animal n'est aimé par personne.
3. Johann aime tous les animaux.
4. C'est soit Johann, soit Gargamel qui a frappé le chat qui s'appelle Azraël.
5. Les chats sont des animaux.

Questions

1. Modéliser chacun des indices en logique du premier ordre.
2. Mettre cette modélisation sous forme normale conjonctive.
3. Déduire, en utilisant les principes de résolution et de réfutation, que Gargamel a frappé le chat Azraël.

Problème - Démineur en logique propositionnelle

On souhaite modéliser le jeu du démineur en logique propositionnelle.



D'après Wikipédia :

Le Démineur (Minesweeper) est un jeu vidéo de réflexion dont le but est de localiser des mines cachées dans une grille représentant un champ de mines virtuel, avec pour seule indication le nombre de mines dans les zones adjacentes.

On souhaite modéliser le problème en se concentrant sur le coin de la grille $(0,0)$, adjacente à seulement 3 cases.

On utilisera seulement des variables de type $X_{i,j}$ signifiant qu'il y a une mine sur la case (i,j) . On évitera également de partir d'une forme normale disjonctive pour répondre aux questions.

Questions

1. Quelles sont les cases adjacentes à la case $(0,0)$?
2. Coder la contrainte « la case $(0,0)$ n'est adjacente à aucune mine ». Transformer cette formule en un ensemble de clauses.
3. Coder la contrainte « la case $(0,0)$ est adjacente à une seule mine ». Transformer cette formule en un ensemble de clauses.
4. Coder la contrainte « la case $(0,0)$ est adjacente à exactement 2 mines ». Transformer cette formule en un ensemble de clauses.
5. Coder la contrainte « la case $(0,0)$ est adjacente à exactement 3 mines ». Transformer cette formule en un ensemble de clauses.
6. En déduire la formule générale pour n'importe quelle case (i,j) « la case (i,j) adjacente à n cases, est adjacente à k mines ». Transformer cette formule en un ensemble de clauses.