



SUPINFO
International University

Sign of Success

4AIT – TP NOTE

Programmation Logique

Un monde contrôlé

Programmation
Espace d'états
Heuristique
Solveur

SOMMAIRE

1	PREAMBULE : LES CONSIGNES GENERALES.....	3
2	RESOLUTION (4 POINTS)	3
3	PROGRAMME PROLOG (6 POINTS).....	3
4	PROBLEME PROLOG (10 POINTS)	4

1 PREAMBULE : LES CONSIGNES GENERALES

Votre rendu se fera sous la forme d'un **dossier compressé (.zip)** nommé [4AIT]-IDOpenCampus-NomDuCampus-Nom-Prénom. Votre dossier peut contenir **uniquement** des documents sous la forme **.pdf, .doc** et **.pl**.

Pour cet examen, **vous pouvez utiliser les supports de cours (.ppt, LABS)**. L'utilisation d'**internet est interdite**. L'outil autorisé est **swi-prolog** **pas d'autres outils**. Vous pouvez utiliser papiers/stylos (non donnés par l'école). Si votre surveillant(e) constate une tricherie, votre épreuve sera annulée et votre relevé de notes portera la mention de « **cheater** » pour cet examen.

2 RESOLUTION (4 POINTS)

Question 1.1 (2 points): Soit l'ensemble : $\{a \vee b, a \vee \neg c, \neg b \vee c, \neg a\}$
Prouver que les clauses sont satisfiables ou insatisfiables.

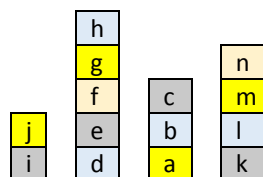
Question 1.2 (2 points): Vous disposez des propositions suivantes :

- $(a \wedge b) \Rightarrow c$
- $d \Rightarrow a$
- $d \Rightarrow b \vdash d \Rightarrow c$

Le système est-il valide ? Répondre à cette question en utilisant la forme Clausale.

3 PROGRAMME PROLOG (6 POINTS)

Vous disposez de 4 piles de cubes :



Question 2.1 (1 point): Ecrire la base de faits de ces 4 piles avec 3 prédicats Prolog, **SommetPile** d'arité 1, **BasPile** d'arité 1, **Sur** d'arité 2.

Question 2.2 (1.5 points): Ecrire 1 prédicat Prolog nommé **CompteCubeDessus** d'arité 2 qui indique le nombre de cube posé sur un cube. Par exemple **compteCubeDessus(a,R) \rightarrow R = 2**, **compteCubeDessus(m,R) \rightarrow R = 1**.

Les prédicats Prolog **SommetPile** et **BasPile** sont maintenant exclus de la base de faits.

Question 2.3 (1 point): Ecrivez 2 nouveaux prédicats nommés **NouveauSommetPile** et **NouveauBasPile** d'arité 1 en fonction uniquement du prédicat **Sur** d'arité 2 de la base de faits.

Question 2.4 (1.5 points) : Ecrire les prédicats Prolog nommés **Sommet** et **Bas** d'arité 0 qui donnent les listes de tous les cubes de bas de Pile et tous les cubes de sommet de Pile, en utilisant les prédicats de la **question 2.3**.

Question 2.5 (1 point) : Ecrire 1 prédicat Prolog nommé **CompteCubeDessous** d'arité 2 qui indique le nombre de cube dessous un cube en utilisant entre autres les prédicats de la **question 2.3**. Par exemple **compteCubeDessous(a,R) → R = 0**, **compteCubeDessous(m,R) → R = 2**.

4 PROBLEME PROLOG (10 POINTS)

Un drone peut voler si son lanceur lui donne un code de démarrage. Le code est donné par un utilisateur et par un automate. Le code est soumis à des contraintes. Vous devez créer un programme Prolog pour fournir des codes valides à 5 chiffres. Les codes se présentent sous la forme d'une séquence de 5 valeurs.



Figure1 : Exemple d'un code à 5 chiffres.

Voici les caractéristiques imposées sur la séquence :

1. 5 chiffres.
2. Les 3 premiers chiffres ont une valeur entre 1 et 5, soit [1..5].
3. Le 4^{ème} chiffre a une valeur entre 1 et 6, soit [1..6].
4. Le 5^{ème} chiffre a une valeur entre 1 et 7, soit [1..7].
5. La somme des chiffres de la séquence est 16.

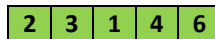


Figure2 : Exemple d'un code valide.

Question 2.1 (3 points) : Ecrire 1 prédicat Prolog nommé **Vérifier** d'arité 1, qui demande à un utilisateur de rentrer les deux premiers chiffres compris entre 1 et 5 et qui produit les 3 autres chiffres en respectant les contraintes nommées.

Exemple :

1 ?- verifier(L).

Le premier chiffre

|: 2.

Le deuxième chiffre

|: 3.

L = [2, 3, 1, 3, 7]

Votre programme évolue :

1. La somme des 5 chiffres est toujours 16.
2. Les 2 premiers chiffres donnés par l'utilisateur sont obligatoirement différents (les valeurs sont toujours comprises entre 1 et 5).
3. Les 3 chiffres donnés par votre programme sont différents eux aussi entre eux et différents de ceux de l'utilisateur.
4. Les 3 autres chiffres donnés par votre programme suivent toujours les contraintes initiales :
 - a. Le choix du premier chiffre de la machine se fait entre les valeurs : 1 et 5,
 - b. Le choix du deuxième chiffre de la machine se fait entre les valeurs : 1 et 6,
 - c. Le choix du troisième chiffre de la machine se fait entre les valeurs : 1 et 7.

Question 2.2 (3 points) : Ecrire 1 prédicat Prolog nommé **Combinaisons** d'arité 1, qui donne toutes les combinaisons en une seule demande et qui redemande à l'utilisateur de rentrer ses chiffres si ils ne sont pas différents.

Question 2.3 (1 point) : Donnez toutes les combinaisons avec votre prédicat **Combinaisons** d'arité 1, pour les couples de chiffres de l'utilisateur (1,2), (2,3), (3,4), et (4,5).

Votre programme évolue, nous gardons toutes les contraintes sauf que maintenant votre programme ne se limite pas uniquement à la valeur 16. La valeur de la somme sera passée en argument.

Question 2.4 (3 points): Ecrire 1 prédicat Prolog nommé **Combinaisons2** d'arité 2, qui donne toutes les combinaisons en une seule demande et qui redemande à l'utilisateur de rentrer ses chiffres si ils ne sont pas différents et qui permet de donner une valeur pour la somme.

Exemple :

1 ?- combinaisons2(L, 23).

Le premier chiffre

|: 4.

Le deuxième chiffre

|: 5.

[4, 5, 1, 6, 7]