



UE InfF4 : Examen de Prolog

Durée : 1 h 30

Documents autorisés, appareils électroniques interdits

Exercice 1 : Qui aime qui (5 points)

On considère le programme Prolog suivant :

```
femme(marie).
femme(julie).

aime(jean,_).
aime(jean,marie).
aime(arnaud,vin).
aime(julie,jean).
aime(jean,X) :- aime(X,vin).
aime(jean,X):- femme(X), aime(X,jean).
aime(marie,X):- not(aime(X,vin)).
```

- a) Quelles sont toutes les réponses de Prolog à la requête `aime(X,vin)` ?
- d) Quelle est la réponse de Prolog à la requête `aime(marie,jean)` ?
- c) Quelle est la réponse de Prolog à la requête `aime(marie,julie)` ?
- d) Quelles sont toutes les réponses de Prolog à la requête `aime(marie,X)` ?
- e) Quelles sont toutes les réponses de Prolog à la requête `aime(jean,X)` ?

Exercice 2 : Trois carrés pour un (4 points)

Ecrire un programme Prolog qui résout le problème suivant :

- *C'est un nombre de cinq chiffres.*
- *C'est un carré parfait.*
- *Le nombre formé par ses 2 premiers chiffres est un carré parfait.*
- *Son chiffre central est aussi un carré parfait.*
- *Le nombre formé par ses 2 derniers chiffres est un carré parfait.*
- *Son chiffre central est égal à la somme de son premier et de son dernier chiffre.*

Quel est ce nombre ?

Ecrire le prédicat `troiscarrés/1` permettant de trouver le nombre à cinq chiffres recherché.

```
?- troiscarrés(N).
N = 36481
```

Exercice 3 : Listes de lettres (6 points)

On considère une liste de lettres minuscules codée par une liste de doublets : le premier élément du doublet est un nombre d'occurrences, le second la lettre. Par exemple, la liste `[a,a,a,a,b,c,c,a,a,d,e,e,e,e]` est codée par la liste `[[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]]`.

a) Ecrire le prédicat **decoder/2** qui à partir d'une liste codée produit la liste décodée. Exemple :

```
?- decoder([[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]], L).
L = [a,a,a,a,b,c,c,a,a,d,e,e,e,e]
```

b) Ecrire le prédicat **decoderpaquets/2** qui à partir d'une liste codée produit une liste formée de paquets de lettres identiques. Exemple :

```
?- decoderpaquets([[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]],L).
L = [[a,a,a,a],[b],[c,c],[a,a],[d],[e,e,e,e]]
```

c) Ecrire le prédicat inverse **codarpaquets/2** qui à partir d'une liste formée de paquets de lettres identiques, construit la liste de doublets correspondante.

Exemple :

```
?- codarpaquets([[a,a,a,a],[b],[c,c],[a,a],[d],[e,e,e,e]],L).
L = [[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]]
```

Exercice 4 : Grammaire (5 points)

a) Ecrire une **grammaire DCG** en Prolog pour reconnaître des énoncés de la forme `[chiffre, opérateur, chiffre]`. Par exemple : `[un, plus, trois]` ou `[quatre, moins, un]`. Les chiffres vont de **un** à **cinq** et les opérateurs sont **plus** et **moins**. La règle de départ s'appelle **exp**.

Exemple :

```
?- exp([trois, plus, deux],[ ]).
true
?- exp([cinq, moins],[ ]).
false
```

b) Ajouter à la grammaire le code Prolog permettant d'évaluer l'expression. On ajoute un argument à la règle `exp` (la valeur calculée) :

```
?- exp(V,[deux, plus, cinq],[ ]).
V = 7.
?- exp(V,[trois, moins, cinq],[ ]).
V = -2.
```