

**EXAMEN DE LOGIQUE COMPUTATIONNELLE & PROLOG**

12 janvier 2011 – DURÉE 3h00

*La consultation des documents et l'échange des documents et des calculatrices est interdit.*

*L'utilisation des 3 feuilles manuscrites recto-verso, format A4 est autorisée*

- Ne pas détacher les feuilles.
- Utiliser l'espace blanc pour vos réponses et le verso pour brouillon.
- Pensez à indiquer votre nom sur chaque feuille

NOM : .....

DÉTAIL	NOTE LOG. CO.	NOTE PROLOG																								
	21	21																								
Exercice 1.	<table border="1"> <tr><td><b>1</b></td><td><b>2</b></td></tr> <tr><td>2</td><td>2</td></tr> </table>	<b>1</b>	<b>2</b>	2	2	4																				
<b>1</b>	<b>2</b>																									
2	2																									
Exercice 2.	<table border="1"> <tr><td><b>1</b></td><td><b>2a</b></td><td><b>2b</b></td><td><b>2c</b></td><td><b>2d</b></td><td><b>2e</b></td></tr> <tr><td>1</td><td>0.5</td><td>0.5</td><td>0.5</td><td>0.5</td><td>0.5</td></tr> <tr><td><b>2f</b></td><td><b>2g</b></td><td><b>2h</b></td><td><b>2i</b></td><td><b>2j</b></td><td></td></tr> <tr><td>0.5</td><td>0.5</td><td>0.5</td><td>0.5</td><td>0.5</td><td></td></tr> </table>	<b>1</b>	<b>2a</b>	<b>2b</b>	<b>2c</b>	<b>2d</b>	<b>2e</b>	1	0.5	0.5	0.5	0.5	0.5	<b>2f</b>	<b>2g</b>	<b>2h</b>	<b>2i</b>	<b>2j</b>		0.5	0.5	0.5	0.5	0.5		6
<b>1</b>	<b>2a</b>	<b>2b</b>	<b>2c</b>	<b>2d</b>	<b>2e</b>																					
1	0.5	0.5	0.5	0.5	0.5																					
<b>2f</b>	<b>2g</b>	<b>2h</b>	<b>2i</b>	<b>2j</b>																						
0.5	0.5	0.5	0.5	0.5																						
Exercice 3.		3																								
Exercice 4.		1																								
Exercice 5.	<table border="1"> <tr><td><b>2</b></td></tr> <tr><td>2</td></tr> </table>	<b>2</b>	2	<table border="1"> <tr><td><b>1a</b></td><td><b>1b</b></td></tr> <tr><td>1</td><td>2</td></tr> </table>	<b>1a</b>	<b>1b</b>	1	2	3																	
<b>2</b>																										
2																										
<b>1a</b>	<b>1b</b>																									
1	2																									
Exercice 6.		1																								
Exercice 7.	<table border="1"> <tr><td><b>1</b></td><td><b>2</b></td><td><b>3</b></td></tr> <tr><td>1</td><td>1</td><td>3</td></tr> </table>	<b>1</b>	<b>2</b>	<b>3</b>	1	1	3	<table border="1"> <tr><td><b>4</b></td></tr> <tr><td>2</td></tr> </table>	<b>4</b>	2	2															
<b>1</b>	<b>2</b>	<b>3</b>																								
1	1	3																								
<b>4</b>																										
2																										
Exercice 8.		5																								
Exercice 9.		<table border="1"> <tr><td><b>1</b></td><td><b>2</b></td><td><b>3</b></td><td><b>4a</b></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td><b>4b</b></td><td><b>4c</b></td><td><b>4d</b></td><td><b>4e</b></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4a</b>	1	1	1	1	<b>4b</b>	<b>4c</b>	<b>4d</b>	<b>4e</b>	1	1	1	1	8							
<b>1</b>	<b>2</b>	<b>3</b>	<b>4a</b>																							
1	1	1	1																							
<b>4b</b>	<b>4c</b>	<b>4d</b>	<b>4e</b>																							
1	1	1	1																							
Exercice 10.		2																								

### Exercice 1 – Calcul propositionnel

Soit  $A, B, C$  et  $D$  quatre variables booléennes.

1. Donner tous les modèles de

$$(A \vee B \vee C) \wedge (A \vee B \vee D) \wedge (A \vee C \vee D) \wedge (B \vee C \vee D).$$

2. On note  $\mathbb{B}$  l'ensemble  $\{0, 1\}$  des booléens. Combien existe-t-il de fonctions  $\mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$  ?

Montrer qu'elles peuvent toutes être obtenues par des formules contenant uniquement les connecteurs  $\neg, \wedge$  et  $\vee$ .

### Exercice 2 – Calcul des prédicats

Soit la situation suivante :

Il y a quatre élèves,  $A, B, C, D$ .

$A$  et  $D$  sont des prépas.  $B$  et  $C$  sont des ingénieurs.

$A$  et  $B$  sont des premières années.  $C$  et  $D$  sont des deuxièmes années.

$A$  découvre l'informatique.  $B, C$  et  $D$  font un stage.

1. Décrivez le modèle  $M$  correspondant à cette situation. (1 pt LOG)

2. Formalisez les phrases ci-dessous dans le langage des prédicats. Pour chacune des formules correspondantes, dites si elle est vraie ou fausse dans le modèle  $M$  en justifiant votre réponse. (2 pt LOG)

(a)  $A$  découvre l'informatique ou fait un stage

(b) Si  $B$  et  $C$  font un stage alors  $D$  ne découvre pas l'informatique

(c) Tous les élèves sont en première ou deuxième année

(d) Si un élève découvre l'informatique, un élève fait un stage

(e) Tous les élèves qui font un stage ne découvrent pas l'informatique

(f) Aucun prépa ne fait de stage

(g) Si tous les ingénieurs font un stage, tous les prépas découvrent l'informatique

(h) Certains ingénieurs ne découvrent pas l'informatique

(i) Si un prépa découvre l'informatique alors un prépa fait un stage

(j) Si tous les prépas sont en première année ou font un stage, alors tous les ingénieurs sont en seconde année ou découvrent l'informatique.

### Exercice 3 – Listes et récursivité

Écrire un prédicat `Prolog repartition(L,LT,Lres)` permettant de répartir les éléments d'une liste  $L$  en sous listes de tailles respectives  $LT$ . (3 pt PRO)

Exemple :

?- `repartition([a,b,c,d,e,f,g,h,i],[2,4,3],Lres)`.

$Lres = [[a, b], [c, d, e, f], [g, h, i]]$

### Exercice 4 – Forme normale conjonctive

Mettre sous forme normale conjonctive la fbf suivante

$$\forall X \exists Y \forall Z \neg(p(X, Z) \rightarrow q(X, Y))$$

**SOL.**  $p(X, Z) \wedge q(X, s_Y(X))$  où  $s_Y(X)$  fonction de Skolem pour la variable  $Y$  associée à un quantificateur existentiel qui est sous le quantificateur universel pour la variable  $X$ .

### Exercice 5 – Coupure

Soit le programme suivant :

```
div(0,Y,0) :- !.
div(X,Y,s(Z)) :- Y\==0,minus(X,Y,U), div(U,Y,Z),!.
div(X,0,Z) :- !,fail.
```

```
minus(X,0,X).
minus(s(X),s(Y),Z) :- minus(X,Y,Z).
```

1. Pour chacune des trois coupures du programme, répondre aux questions suivantes en justifiant vos réponses :

(a) Est-ce cette coupure “rouge” ou “verte” ?

**SOL.**

(b) Qu’est-ce qui change lors de l’exécution du programme si cette coupure est omise ?

**SOL.**

2. Dédurre de ce qui précède l’arbre SLD de la réponse à la question ?-  $\text{div}(X, s(0), s(0))$  et donner l’ensemble des réponses calculées par celui-ci.

**SOL.**

### Exercice 6 – Unification

Trouver, s’il existe, l’unificateur le plus général pour les deux fbf suivantes :

$$p1 : g(X, h(h(Z)), f(Z, X)) \text{ et } p2 : g(h(Y), h(X), f(a, h(Z)))$$

**SOL.** En appliquant successivement les substitutions  $\sigma_1 = (X/h(Y))$ ,  $\sigma_2 = (Z/Y)$  et  $\sigma_3 = (Y/a)$  les deux formules se reduisent à la formule  $g(h(a), h(h(a)), f(a, h(a)))$ .

### Exercice 7 – Modèle minimal de Herbrand

Soit le programme défini  $E$  suivant

```
p(X,a) :- q(X).
p(X,Y) :- q(X),r(Y).
q(a).
q(b).
r(b).
r(c).
```

1. Donner l’univers de Herbrand de ce programme.

**SOL.**  $\mathcal{U}_E = \{a, b, c\}$

2. Donner la base de Herbrand de ce programme.

**SOL.**  $\mathcal{B}_E = \{p(a, a), p(a, b), p(a, c), p(b, a), p(b, b), p(b, c), p(c, a), p(c, ab), p(c, c), q(a), q(b), q(c), r(a), r(b), r(c)\}$

3. Calculer le modèle minimal de Herbrand.

**SOL.** On applique l’opérateur de la conséquence immédiate. On a

$$\begin{aligned} \mathcal{T}_E \uparrow 0 &= \emptyset \\ \mathcal{T}_E \uparrow 1 &= \mathcal{T}_E(\mathcal{T}_E \uparrow 0) = \{q(a), q(b), r(b), r(c)\} \\ \mathcal{T}_E \uparrow 2 &= \mathcal{T}_E \uparrow (\mathcal{T}_E \uparrow 1) = \{q(a), q(b), r(b), r(c), p(a, b), p(b, b), p(b, c)\} \\ \mathcal{T}_E \uparrow 3 &= \mathcal{T}_E \uparrow (\mathcal{T}_E \uparrow 2) = \{q(a), q(b), r(b), r(c), p(a, a), p(b, a), p(a, b), p(a, c), p(b, b), p(b, c)\} = \mathcal{T}_E \uparrow 2 \end{aligned}$$

Le modèle minimal de Herbrand est le point fixe de l’opérateur  $\mathcal{T}_E \uparrow 2$ , c-a-d.

$$\text{modèle minimal } \mathcal{I}_E = \{q(a), q(b), r(b), r(c), p(a, a), p(b, a), p(a, b), p(a, c), p(b, b), p(b, c)\}$$

4. Quelle est la raison qui permet d'utiliser le modèle minimal de Herbrand afin d'évaluer les réponses de Prolog du programme précédent à une question.

Application.- Donner les réponses de Prolog à la question ?-  $p(b, X)$ .

**SOL.** Le modèle minimal de Herbrand contient toutes les réponses que le programme peut fournir à une question.

Application.- En examinant  $\mathcal{I}_E$  on trouve comme réponses à la question ?-  $p(b, X)$  :

$Y = a$ ;

$Y = b$ ;

$Y = c$ .

### Exercice 8 – Contraintes

On appelle carré latin d'ordre  $n$  une matrice  $n * n$  telle que chacun des nombres de 1 à  $n$  est présent exactement une fois sur chaque ligne et chaque colonne.

Écrire un programme qui trouve les carrés latins d'ordre 3.

**SOL.**

```
carre_latin :-
    L=[X11,X12,X13,X21,X22,X23,X31,X32,X33],
    L ins 1..3,
    all_different([X11,X12,X13]),
    all_different([X21,X22,X23]),
    all_different([X31,X32,X33]),
    all_different([X11,X21,X31]),
    all_different([X12,X22,X32]),
    all_different([X13,X23,X33]),
    label(L),
    afficher(L).
```

```
afficher(L) :- afficher_aux(L,1).
afficher(_L) :- nl.
afficher_aux([],10) :- nl,!,fail.
afficher_aux([X|L],N) :-
    write(X),
    write(' '),
    ((N mod 3:=0,nl);true),
    !,
    N1 is N+1,
    afficher_aux(L,N1).
```

### Exercice 9 – Bases de données et planification

Soit un extrait des enregistrements de la base de données d'une bibliothèque

```
% livre(titre, nomAuteur, prenomAuteur, editeur).
livre(guideClubs, toto, jean, editions_du_village).
livre(guideClubs, koko, marcel, editions_du_village).
livre(guideRestos, jojo, simon, edition_de_la_plume).
livre(guideCD, toto, jean, editions_du_village).
livre(guideDVD, momo, maurice, les_trois_lanternes).
livre(guideProlog, lolo, georges, edition_du_fanion).
```

Un livre peut apparaître plus d'une fois s'il est plusieurs auteurs. Par exemple le 1er et le 2e enregistrement de cette base.

1. Telle quelle cette base permet-elle de faire des recherches (d'auteurs, de titres, etc.)? Justifier votre réponse et, dans le cas où la réponse est négative, ajouter le(s) champ(s) nécessaire(s) pour qu'elle devienne opérationnelle.

**SOL.** Non, car par enregistrement il n'y a pas une clé unique. On pourrait utiliser comme clé unique le numéro ISBN du livre que l'on placera en première position dans chaque enregistrement, à savoir

```
% livre(num\eroISBN, titre, nomAuteur, prenomAuteur, editeur).
```

2. Créer une bdd virtuelle (comme on avait vu en TP) pour cette bdd.

**SOL.**

```
bddVirt(livre,Isbn,isbn,Isbn) :- livre(Isbn,_,_,_,_).  
bddVirt(livre,Isbn,titre,Titre) :- livre(Isbn,Titre,_,_,_).  
bddVirt(livre,Isbn,nomAuteur,Nom) :- livre(Isbn,_,Nom,_,_).  
bddVirt(livre,Isbn,prenomAuteur,Prenom) :- livre(Isbn,_,_,Prenom,_).  
bddVirt(livre,Isbn,editeur,Editeur) :- livre(Isbn,_,_,_,Editeur).
```

3. Utiliser la bdd virtuelle pour qu'à la question ?- auteur(NomAuteur) obtenir tous les titres et les éditeurs d'un auteur.

Exemple : ?- auteur(toto). donne comme réponse :

```
guideClubs, editions_du_village  
guideCD, editions_du_village
```

N.B.- On suppose que les opérations `afficheEltListe [Elt,Criteres]` et `afficheTout X` ainsi que le programme pour le prédicat `for(X,Y)`, qui sont déjà vus en TP, existent et donc on pourra les utiliser sans les re-écrire.

**SOL.**

```
auteur(NomA) :- afficheEltListe [(Titre, Ed),  
    (bddVirt(livre,Isbn,nomAuteur,NomA),  
    bddVirt(livre,Isbn,titre,Titre),  
    bddVirt(livre,Isbn,editeur,Ed))].
```

4. Supposons que les livres sont placés dans les rayonnage de la bibliothèque par ordre alphabétique du nom du premier auteur, à des cases numérotées 1,2, ... . Supposons aussi qu'il reste des cases libres numérotées. Par exemple supposons qu'il y a 5300 livres et il y a 6000 cases numérotées ;

Les nouveaux livres sont placés aux rayonnages à la fin de chaque mois et on suppose que pendant cette opération tous les livres de la bibliothèque sont dans leurs cases respectives. Pour placer un livre dans une case, en respectant l'ordre alphabétique, on suit une procédure, certes non optimale mais ô combien reposante pour l'esprit : pour libérer la case qui correspond au livre, selon l'ordre alphabétique du nom de l'auteur, on décale d'une case vers la droite tous les livres, à partir de cette case, ce qui permet de libérer la case en question et placer ainsi le nouveau livre dans l'emplacement approprié.

On voudrait automatiser cette procédure en écrivant un programme en Prolog qui planifie les différents décalages.

Présenter les éléments de base pour cette planification, à savoir

- (a) les objets et les états de la bibliothèque. En particulier on doit présenter les prédicats qui permettent la description d'un état, comme par exemple le prédicat `sur` ou `libre` que nous avons vu pour le monde des blocs.

**SOL.** Les objets sont les livres et les cases. L'état est, à chaque instant, décrit par une base de données qui contient les livres et pour chaque livre le numéro de la case dans laquelle se trouve.

- (b) un extrait significatif de la base de données qui décrit l'état de la bibliothèque. On suppose que nous avons 5300 livres dans les cases correspondantes et les cases numérotées de 5301 à 6000 sont libres. On prendra soin de limiter autant que faire se peut la redondance entre cette base de données et la base de données `livre`.

**SOL.** La base de données bibliothèque est la suivante :

```
% dans(NuCase, IsbnLivre).  
pour les premiers 5300 enregistrements et  
% libre(NuCase).  
pour le reste.
```

(c) les actions qui permettent l'évolution de la bibliothèque (c'est-à-dire son passage d'un état à un autre) ;

**SOL.** Il y a deux actions : `decal` – décalage à droite d'un livre et `place` – placement d'un livre dans une case.

(d) pour chaque action, les contraintes sous lesquelles cette action puisse avoir lieu. Pour la description de l'action on doit utiliser le prédicat `action` qu'on avait vu en TP où on a traité la planification avec Prolog, avec, bien évidemment, d'autres arguments.

**SOL.** `action(decal(IsbnLivre,NuCase,NuCase1), [dans(NuCase, IsbnLivre), libre(NuCase1)])`.  
`action(place(IsbnLivre,NuCase), [libre(NuCase)])`.

Notons que `NuCase1` est la case numéro `NuCase+1` que l'on peut calculer en utilisant le prédicat arithmétique `suc(X,X1) :- X1 is X+1`.

(e) pour chaque action, les faits nouveaux qui se rajoutent et les faits existants qu'on supprime. Pour leur description on doit utiliser (avec d'autres arguments) les prédicats `ajout` et `suppres` qu'on avait vu au même TP.

**SOL.** L'action `decal(IsbnLivre,NuCase)` vide la case `NuCase` et remplit la case `NuCase+1`. Donc

`ajout(dans(IsbnLivre, NuCase1),decal(IsbnLivre,NuCase, NuCase1))`.

`ajout(libre(IsbnLivre, NuCase),decal(IsbnLivre,NuCase, NuCase1))`.

et par conséquent

`supp(dans(IsbnLivre, NuCase),decal(IsbnLivre,NuCase, NuCase1))`.

`supp(libre(NuCase1),decal(IsbnLivre,NuCase, NuCase1))`.

L'action `place(IsbnLivre,NuCase)` remplit la case `NuCase`. On a donc

`ajout(dans(IsbnLivre, NuCase), place(IsbnLivre,NuCase))`.

et

`sup(libre(NuCase), place(IsbnLivre,NuCase))`.

### Exercice 10 – Logique de Hoare

On cherche à écrire un programme qui calcule le factoriel d'un naturel  $n$ . Nous utiliserons la logique de Hoare.

Nous avons la suite de deux de triplets du type  $\{Q\} P \{R\}$  suivante :

```
{n>0} r <-- 1 {R}
{R} while n > 0 [P] {r = n!}
```

Donner la post-condition  $R$  du premier triplet et aussi le code du programme  $[P]$  qui est sous la boucle de `while` du second triplet.

N.B.- Dans la deuxième ligne de la suite des triplets ci-dessus, la précondition  $\{R\}$  est la même que la post-condition  $\{R\}$  de la première ligne.

**SOL.** On a

```
{R} = {n>0, r = 1}
[P]=[r <-- r*n; n <-- n-1]
```