Examen de programmation logique et fonctionnelle

Licence d'informatique, Université de Bourgogne, UFR Sciences et Techniques.

Sujet de la session 1, mai 2022. Durée 2h.

Documents autorisés : 4 pages A4 recto-verso manuscrites ou imprimées, avec le contenu de votre choix.

Calculatrices et appareil électroniques communicants interdits.

Logique propositionnelle

A1 (1point)

Donnez tous les modèles de la formule suivante :

$(a \wedge b \wedge c \wedge d) \vee (\neg a \wedge \neg b \wedge \neg c \wedge \neg e)$

A2 (2 points)

Parmi les formules suivantes, y-a-t-il des tautologies ? Et si oui lesquelles ? Y-a-t-il des formules incohérentes ? Et si oui lesquelles ?

Sur la fiche de réponse, donnez les numéros des tautologies (si applicable) et des formules incohérentes (si applicable).

1. $(a \to b) \to \neg b$
2. $(a \to b) \to b$
3. $a \to (a \to b)$
4. $b \to (a \to b)$

A3 (1,5 point)

Soient les deux formules suivantes :

- $\Sigma = (a \vee b) \to c$
- $\Omega = (a \wedge b) \to c$

Donnez les tables de vérité de ces deux formules. D'après ces tables, lesquelles de ces trois affirmations sont correctes :

1. $\Sigma \models \Omega$
2. $\Omega \models \Sigma$
3. $\Omega \equiv \Sigma$

A4 (1,5 point)

On considère trois variables x_2, x_1, x_0 représentant en base 2 un entier X compris entre 0 et 7. Par exemple, la valuation $\{(x_2, 1), (x_1, 1), (x_0, 0)\}$ représente la valeur $X = 6$.

Donnez une formule Σ utilisant les variables x_2, x_1, x_0 qui modélise la propriété $3 \leq X \leq 5$.

Logique du premier ordre

B1 (1 point)

Soit la formule $\Sigma = \forall X(p(X, X) \rightarrow q(X))$.

Donnez un modèle et un contre-modèle de Σ ayant tous les deux pour domaine d'interprétation l'ensemble $\{1, 2\}$.

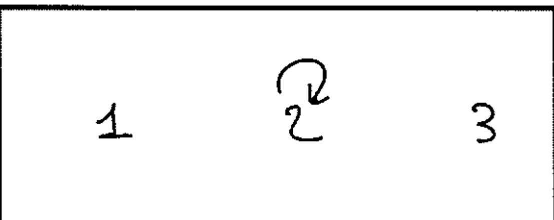
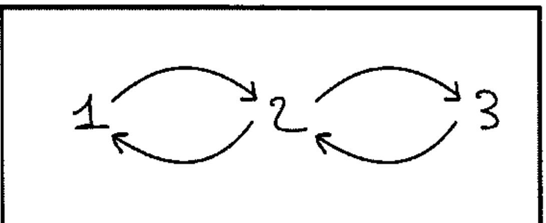
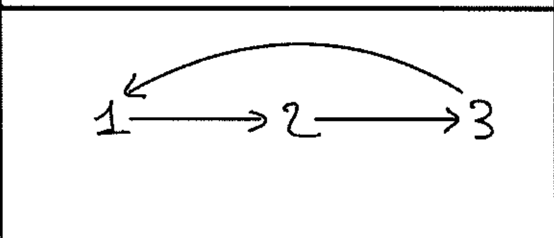
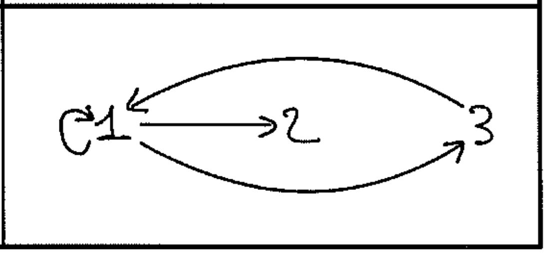
Sur la fiche de réponse, pour donner une interprétation de p, reliez par une flèche toutes valeurs α et β telles que $p(\alpha, \beta) = \text{vrai}$. Pour donner une interprétation de q, entourez toute valeur α telle que $q(\alpha) = \text{vrai}$.

B2 (2 points)

Soient les formules suivantes :

- $\Sigma = \exists X \exists Y(p(X, Y) \wedge p(Y, X))$
- $\Omega = \exists X \forall Y(p(X, Y) \wedge p(Y, X))$

Soient les interprétations suivantes, qui ont toutes pour domaine $\{1, 2, 3\}$. Une flèche reliant une valeur α et une valeur β indique que $p(\alpha, \beta) = \text{vrai}$.

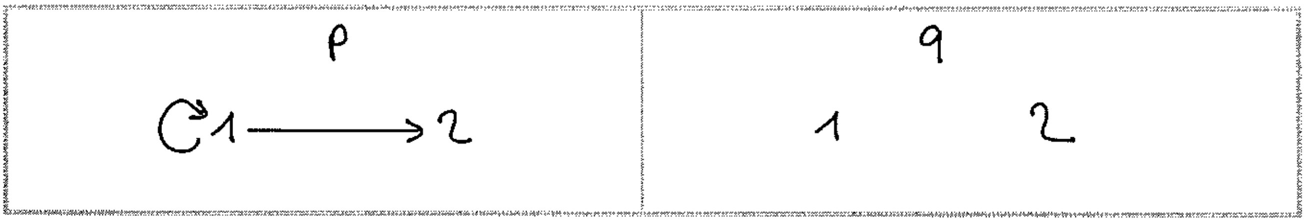
I_1		I_2	
I_3		I_4	

Parmi ces interprétations, lesquelles sont des modèles de Σ ? Lesquelles sont des modèles de Ω?

B3 (1 point)

Soit la formule $\Sigma = \forall X \forall Y[((p(X,Y) \vee \neg q(X,Y)) \wedge ((\neg p(X,Y) \vee q(X,Y))]$

Compétez l'interprétation suivante de manière à ce qu'elle satisfasse Σ. Vous devez ajouter des flèches à l'interprétation de q sans modifier l'interprétation de p.



B4 (2 points)

Donnez une formule modélisant l'énoncé :

> Tout nombre premier est plus petit qu'au moins un autre nombre premier.

Vous devez utiliser les deux prédicats suivants :

- $\text{premier}/1$: $\text{premier}(X)$ est vrai si et seulement si X est un nombre premier.
- $\text{inf}/2$: $\text{inf}(X,Y)$ est vrai si est seulement si X est plus petit que Y.

PROLOG

C1 (2 points)

Complétez la définition du prédicat...

```
sc(P_in, L_in, R1_out, R2_out)
```

...qui scinde la liste d'entrée `L_in` en deux listes de sortie `R1_out` et `R2_out` de manière à ce que :

- `R1_out` contienne tous les éléments de `L_in` inférieurs ou égaux à la valeur d'entrée `P_in`.
- `R2_out` contienne tous les éléments de `L_in` plus grands que la valeur d'entrée `P_in`.

Par exemple, le but...

```
sc(5,[9,5,1,8,7,4],L1,L2).
```

...doit avoir pour résultat :

```
L1 = [5, 1, 4], L2 = [9, 8, 7]
```

(La liste `L1` contient tous les éléments de la liste d'entrée de valeur inférieure ou égale à 5 et la liste `L2` tous les autres éléments.) L'ordre des éléments des deux listes de sortie n'est pas critique.

```
sc(_, ....., ....., .....).
sc(P, [T|Q], ....., .....) :- T =< P, sc(P,Q,R1,R2).
sc(P, [T|Q], ....., .....) :- T > P,  sc(P,Q,R1,R2).
```

C2 (2 points)

Définissez le prédicat...

```
alldiff(L_in)
```

...qui détermine si les éléments de la liste d'entrée `L_in` sont tous différents.

Voici deux exemples de but et leurs résultats :

```
alldiff([2,1,3,1,4,5]). --> false
alldiff([2,1,3,4,5]). --> true
```

Vous pouvez utiliser le prédicat standard `member(X_in,L_in)` permettant de déterminer si `X_in` est un élément de la liste `L_in`.

Vous pouvez également utiliser la négation `not`.

C3 (2 points)

On peut représenter une liste **non vide** d'entiers par un arbre binaire dont les feuilles sont étiquetées par des entiers et les noeuds ne sont pas étiquetés. Un tel arbre sera appelé **arbre-liste** dans la suite de l'exercice.

Par exemple :

- La liste `[1]` sera représentée par `f(1)`.
- La liste `[1,2,3]` pourra être représentée par `n(f(1),n(f(2),f(3)))` ou par `n(n(f(1),f(2)),f(3))`.

Cette représentation a des avantages et des inconvénients par rapport à la représentation standard. Par exemple, la concaténation de deux listes `A` et `B` est immédiate, puisqu'elle s'écrit `n(A,B)`, mais la comparaison de deux listes est plus couteuse, parce qu'une même liste peut admettre plusieurs représentations. Autre inconvénient, la liste vide n'est pas représentable.

Définissez le prédicat...

```
end(L_in,D_out)
```

...qui fait remonter dans la variable `D_out` le dernier élément de l'arbre-liste `L_in`.

Par exemple, le but...

```
end(n(n(f(1),f(2)),f(3)),D).
```

...doit produire le résultat :

```
D=3
```

C4 (2 points)

On reprend la représentation des listes introduite dans l'exercice précédent.

Définissez le prédicat...

```
count(L_in,N_out,S_out)
```

...qui fait remonter dans la variable de sortie `N_out` le nombre d'éléments de la liste représentée par l'arbre-liste `L_in` et qui fait remonter dans `S_out` la somme de ces éléments.

Par exemple, le but...

```
count(n(f(1),n(f(2),f(3))),N,S).
```

...doit produire le résultat :

```
N=3, S=6
```

L3 - PLF - mai 2022 Numéro anonymat :

A1 (1)

A2 (2)

Tautologies :

Incohérentes :

A3 (1,5)

a	b	c	Σ	Ω
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Affirmations correctes :

B1 (1)

Modèle de Σ :

p	q
1 2	1 2

Contre-modèle de Σ :

p	q
1 2	1 2

B2 (2)

Modèles de Σ :

Modèles de Ω :

B3 (1)

p	q
$1 \longrightarrow 2$	1 2

A4 (1,5)

B4 (2)

C1(2)

```
sc(_,          ,          ,          ).
sc(P, [T|Q],          ,          ):- T =< P, sc(P,Q,R1,R2).
sc(P, [T|Q],          ,          ):- T > P, sc(P,Q,R1,R2).
```

C2(2)

C3(2)

C4(2)