Examen de programmation logique et fonctionnelle

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

Sujet de la session 1, mai 2023. Durée 2h.

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

Calculatrices et appareils électroniques communicants interdits.

Logique propositionnelle (6 points)

Q1 (1 point)

Combien de modèles et de contre-modèles admet la formule suivante ?

$a \vee b \vee c \vee d \vee e$

Q2 (2 points)

La formule suivante est-elle une tautologie ? Est-elle cohérente ? Justifiez brièvement vos réponses.

$(a \rightarrow b) \rightarrow (b \rightarrow a)$

Q3 (1 point)

L'affirmation suivante est elle correcte ? Justifiez brièvement votre réponse.

$\neg a \models a$

Q4 (2 points)

On représente un entier A compris entre 0 et 2 par trois variables a_0, a_1, a_2 avec la convention suivante :

A	a_0	a_1	a_2
0	1	0	0
1	0	1	0
2	0	0	1

On représente un entier B compris entre 0 et 2 par les variables b_0, b_1, b_2 avec la même convention. La formule $\Sigma_A = (a_0 \wedge \neg a_1 \wedge \neg a_2) \vee (\neg a_0 \wedge a_1 \wedge \neg a_2) \vee (\neg a_0 \wedge \neg a_1 \wedge a_2)$ modélise l'intégrité de la représentation de l'entier A, dans le sens où elle est satisfaite uniquement par les assignations de a_0, a_1, a_2 qui représentent les valeurs $A = 0$, $A = 1$ et $A = 2$.

La formule $\Sigma_B = (b_0 \wedge \neg b_1 \wedge \neg b_2) \vee (\neg b_0 \wedge b_1 \wedge \neg b_2) \vee (\neg b_0 \wedge \neg b_1 \wedge b_2)$ modélise l'intégrité de la représentation de l'entier B.

Vous devez donnez une formule Ω telle que $\Sigma_A \wedge \Sigma_B \wedge \Omega$ modélise la propriété $A < B$. Sur la fiche de réponse, donnez uniquement Ω. Ne recopiez pas les formules Σ_A et Σ_B.

Logique du premier ordre (6 points)

Q5 (1 point)

Soit la formule Σ suivante :

$[\exists X \exists Y p(X, Y)] \rightarrow [\forall X \forall Y p(X, Y)]$

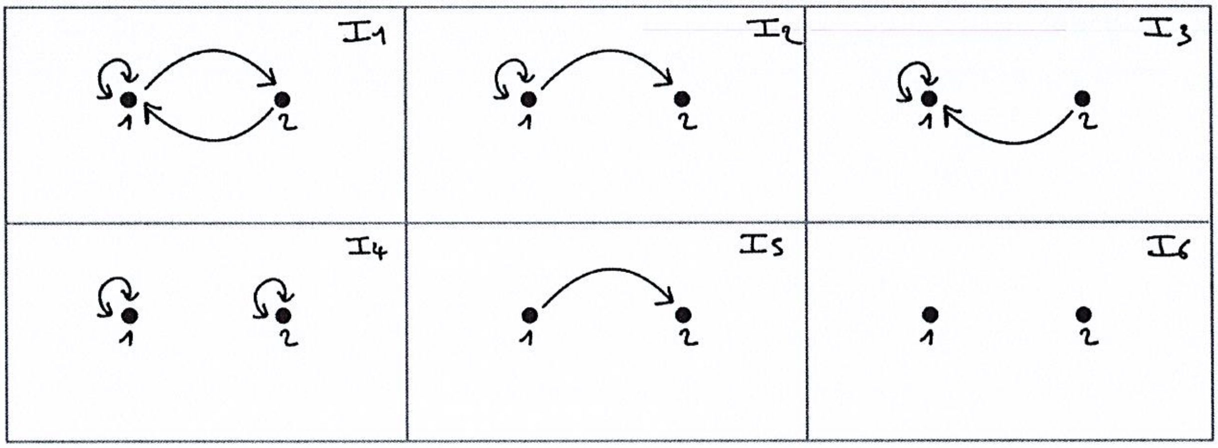
Donnez, en représentation sagittale, deux modèles de Σ avec le domaine d'interprétation $\{1, 2\}$.

Q6 (3 points)

Soient les formules suivantes :

- $\Sigma_1 : \exists X \forall Y\, [p(X, Y) \wedge p(Y, X)]$
- $\Sigma_2 : \exists X \forall Y\, [p(X, Y) \vee p(Y, X)]$
- $\Sigma_3 : \forall X \exists Y\, [p(X, Y) \wedge p(Y, X)]$
- $\Sigma_4 : \forall X \exists Y\, [p(X, Y) \vee p(Y, X)]$

Soient les interprétations suivantes du symbole p sur le domaine $\{1, 2\}$:

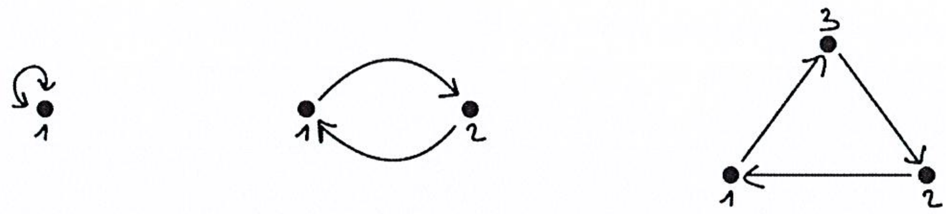


Lesquelles des interprétations I_1 à I_6 satisfont chacune des 4 formules Σ_1 à Σ_4 ? (Sur la fiche de réponse, entourez les modèles de chaque formule.)

Q7 (2 points)

On représente un graphe orienté par une interprétation d'un symbole relationnel g. Chaque élément du domaine d'interprétation représente un sommet du graphe. Pour tout sommet X et tout sommet Y, $g(X, Y)$ est vrai si et seulement si un arc du graphe relie X à Y. En d'autres termes, le graphe est la représentation sagittale de l'interprétation de g.

Un circuit de longueur k est une séquence de k arcs consécutifs qui commence et se termine à un même sommet. Voici quelques exemples :



un circuit de longueur 1 — un circuit de longueur 2 — un circuit de longueur 3

Donnez une formule close Ω qui modélise la propriété suivante :

> Le graphe représenté par l'interprétation de g ne comporte aucun cycle de longueur 1, et aucun cycle de longueur 2.

PROLOG (8 points)

Q8 (2 points)

Dans une base de faits Prolog, on utilise le prédicat `hab/2` pour indiquer qu'une personne est habilitée à utiliser une machine. Les personne sont représentées par des symboles fonctionnels, comme `anne`, `bart`, etc. Les machines sont représentées par des entiers comme 1, 2, 3, etc.

Voici un exemple de base de faits.

```
hab(anne,1). hab(anne,2).
hab(bob,1). hab(bob,2).
hab(carl,2). hab(carl,3).
hab(dora,1). hab(dora,3). hab(dora,4).
```

Anne et Bob sont habilités à utiliser les machines 1 et 2, Carl peut utiliser les machines 2 et 3, et Dora peut travailler avec les machines 1, 3 et 4.

On définit par ailleurs le prédicat `alldiff/1` tel que si `L` est une liste, alors `alldiff(L)` est vrai si et seulement si les éléments de `L` sont tous différents.

```
alldiff([]).
alldiff([T|R]) :- not(member(T,R)), alldiff(R).
```

Par exemple, le but `alldiff([1,2,1,3])` échoue car la valeur 1 apparait deux fois dans la liste d'entrée, alors que `alldiff([2,4,1,3])` réussit. Pour exprimer, par exemple, que trois variables `A`, `B` et `C` ont des valeurs différentes, on écrit `affdiff([A, B, C])`.

Q8.1 (1 point)

Donnez les buts Prolog permettant d'obtenir les informations suivantes :

1. Quelles sont les machines pouvant être utilisées par Bart ?
2. Quelle machines Dora peut elle utiliser ?
3. Carl est-il habilité à utiliser la machine 3 ?

Sur la fiche de réponse, vous pouvez mettre ces trois buts sur une même ligne, en les séparant par des points.

Q8.2 (1 point)

Spécifiez le prédicat `affecte/4` tel que le but `affecte(Ma, Mb, Mc, Md)` recherche une manière d'attribuer une machine à Anne, Bart, Carl et Dora en respectant les deux conditions suivantes :

- Chacune des 4 personnes doit être habilitée à utiliser la machine qui lui est assignée.
- Il ne peut pas y avoir plus d'une personne travaillant sur une même machine.

Pour chaque solution, l'évaluation du but doit faire remonter dans la variable `Ma` le numéro de la machine attribuée à Anne, dans la variable `Mb` le numéro de la machine attribuée à Bart, dans la variable `Mc` le numéro de la machine attribuée à Carl, et dans la variable `Md` le numéro de la machine attribuée à Dora.

Vous devez utiliser le prédicat `alldiff/1`.

Q9 (2 points)

Définissez le prédicat `sqs/2` tel que si `L` est une liste d'entiers, alors le but `sqs(L,S)` fait remonter dans la variable `S` la somme des carrés des éléments de `L`.

Par exemple, l'évaluation du but `sqs([1,2,3],S)` doit produire le résultat `S=14`.

Q10 (2 points)

Le prédicat `addFin/3` a été définit de la manière suivante dans les supports d'enseignement.

```
addFin([],F,[F]).
addFin([T|Q],F,[T|R]) :- addFin(Q,F,R).
```

Ce prédicat permet notamment de décomposer une liste connue non vide, placée en entrée dans le troisième slot, en deux parties : son dernier élément et la liste de tous les autres éléments.

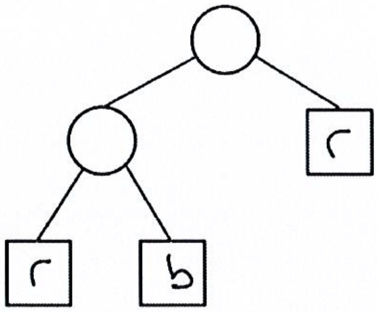
Par exemple, le but `addFin(P,F,[1,2,3])` produit le résultat `F=3`, `P=[1,2]`.

En utilisant le prédicat `addFin/3`, vous devez définir le prédicat `miroir/2` tel que si `L` est une liste, alors le but `miroir(L,M)` fait remonter dans `M` la liste miroir de `L`, c'est à dire la liste comportant les même éléments, mais en ordre inverse (le dernier en premier, l'avant-dernier en deuxième, etc.)

Par exemple, l'évaluation du but `miroir([1,2,3],M)` a pour résultat `M=[3,2,1]`.

Q11 (2 points)

On s'intéresse aux arbres à feuilles colorées qui, dans cet exercice, sont des arbres binaires dont les noeuds ne sont pas étiquetés et pouvant avoir deux types de feuilles : les rouges et les bleues. Voici un exemple d'arbre à feuilles colorées et sa représentation.



n(n(r,b),r)

Les feuilles rouges sont représentées par le symbole fonctionnel `r/0`, les feuilles bleues par le symbole fonctionnel `b/0`, et les noeuds sont représentés par le symbole fonctionnel `n/2`.

Vous devez définir, avec au plus trois clauses, un prédicat `nchemr/2` permettant de déterminer le nombre de feuilles rouges dans un arbre à feuilles colorées.

Par exemple, l'évaluation du but `nchemr(n(n(r,b),r),R)` doit avoir pour résultat `R=2`.

Fiche réponses **Numéro anonymat :**

`Nombre de modèles :` `de contre-modèles :`

Q1 (/ 1)

Q2 (/ 2)

Q3 (/ 1)

Q4 (/ 2)

Q5 (/ 1)

`Entourez les réponses correctes`

Modèles de Σ_1 : I_1 I_2 I_3 I_4 I_5 I_6

Modèles de Σ_2 : I_1 I_2 I_3 I_4 I_5 I_6

Modèles de Σ_3 : I_1 I_2 I_3 I_4 I_5 I_6

Modèles de Σ_4 : I_1 I_2 I_3 I_4 I_5 I_6

Q6 (/ 3)

Q7 (/ 2)

Q8 (/ 2)

Q9 (/ 2)

Q10 (/ 2)

Q11 (/ 2)