Examen de programmation logique et fonctionnelle

Licence informatique

Université Bourgogne Europe - UFR Sciences et Techniques - année 2024/2025

Documents autorisés : trois feuilles A4 recto-verso avec contenu libre.

Ordinateurs, calculatrices et appareils connectés interdits. Écrivez vos réponses directement sur le sujet, que vous glisserez dans la copie d'examen anonymisée.

Lambda-calcul

A1 (2 points)

Décomposez le terme suivant en abstractions et applications. Les abstractions doivent être encadrées en pointillés et les applications doivent être soulignées.

$$\lambda a.\ \lambda b.\ b\ a\ c\ \lambda a.\ a\ c$$

Entourez les variables libres et connectez les variables liées entre elles dans le terme suivant :

$$\lambda a.\ \lambda b.\ b\ a\ c\ \lambda a.\ a\ c$$

A2 (2 points)

Évaluez complètement les termes suivants :

$(\lambda a.b(ab))\lambda w.ww$

$(\lambda a.\lambda b.b(ab))b$

CAML

On souhaite représenter des listes d'entiers sous forme de variantes à l'aide d'un type `lvar`. Par exemple, la liste `[]` est représentée par `N` et la liste `[1;3;2]` est représentée par `L(1,L(3,L(2,N)))`. Cela revient à représenter chaque liste par un arbre dont chaque nœud est étiqueté par un entier et possède un seul fils. Cet arbre ne comporte qu'une seule feuille, désignée par `N`, qui représente la liste vide.

B1 (1 point)

Donnez la déclaration du type `lvar`.

B2 (1.5 point)

Définissez la fonction `list2lvar` telle que si `w` est une liste d'entier, alors `list2lvar` `w` retourne la représentation de `w` dans le type `lvar`.

B3 (1.5 points)

Définissez la fonction `lvar2list` telle que, si `v` est une valeur du type `lvar`, alors `lvar2list v` retourne la liste d'entiers correspondante (c'est-à-dire la liste représentée par `v`).

Logique propositionnelle

C1 (2 points)

Soient les deux formules suivantes :

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

Donnez tous les contre-modèles de Σ.

Donnez tous les contre-modèles de Ω.

C2 (2 points)

Répondez par oui ou par non aux questions suivantes, sans donner de justification (0 si au moins deux réponses fausses, sinon -1 point par réponse fausse) :

1. Σ est-elle conséquence logique de Ω ?
2. Ω est-elle conséquence logique de Σ ?
3. $\neg\Sigma$ est-elle conséquence logique de $\neg\Omega$?
4. $\neg\Omega$ est-elle conséquence logique de $\neg\Sigma$?

Logique du premier ordre

D1 (2.5 points)

Soit la formule $\Sigma = (\exists X p(X, X)) \rightarrow (\forall Y \forall Z\, p(Y, Z))$. Donnez (en représentations sagittales) tous les modèles de Σ ayant pour domaine d'interprétation l'ensemble $\{1, 2\}$.

D2 (1.5 point)

On suppose que $\mathrm{parent}(X, Y)$ est vrai si et seulement si Y est un enfant de X. Soit la formule $\Omega = \forall X \neg \mathrm{parent}(X, X)$.

- Donnez, en français, un énoncé modélisé par cette formule.

- Donnez une autre modélisation de cet énoncé qui n'utilise pas de quantificateur ∀.

PLOLOG

E1 (1.5 point)

Complétez la définition du prédicat `frag/4` tel que `fag(L,P,E,S)` est vrai si et seulement si la liste `L` se décompose en :

- une liste préfixe `P` possiblement vide,
- suivie d'un élément `E`,
- suivi d'une liste suffixe `S` possiblement vide.

Par exemple, L est la liste [1,2,3,4], alors les décompositions possibles sont :

```
P=[]        E=1    S=[2,3,4]
P=[1]       E=2    S=[3,4]
P=[1,2]     E=3    S=[4]
P=[1,2,3]   E=4    S=[]
```

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

E2 (1 point)

On suppose que le prédicat `frag/4` de la question précédente est correctement défini et on définit le prédicat suivant :

```
myst(L,[D|P]) :- frag(L,P,D,[]).
```

Quel est le résultat du but suivant :

```
myst([1,2,3,4,5],R).
```

E3 (1.5 points)

On appelle représentations de Peano les termes fonctionnels constitués uniquement des symboles fonctionnels `z/0` et `s/1`, par exemple `z`, `s(z)`, `s(s(z))`, `s(s(s(z)))`, ...

Définissez le prédicat `verif/1` tel que `verif(T)` est vrai si et seulement si `T` est une représentation de Peano.