L3 - Programmation logique et fonctionnelle

Examen - session de septembre 2008

Modalités

Tous les documents sur papier sont autorisés. Les calculatrices sont autorisées. Durée 2h00.

1 Lambda-calcul (3 points)

Réduisez les termes de lambda-calcul suivants, en identifiant à chaque étape la tête, le corps et l'argument du redex.

1. $(\lambda x.xx)(\lambda x.\lambda y.yx)$
2. $(\lambda x.xx)(\lambda x.(\lambda y.y)x)$
3. $((\lambda x.x)x)(\lambda x.\lambda y.yx)$

2 Logique

Justifiez de manière concise chacune de vos réponses.

2.1 Logique propositionnelle (3 points)

1. Quelle est la valeur de vérité de la formule $(\neg a \rightarrow b) \wedge (\neg b \rightarrow \neg c) \wedge (c \rightarrow a)$ lorsque $a =$ `vrai`, $b =$ `vrai` et $c =$ `faux` ?
2. Modélisez la phrase « Le raccordement à l'eau et à l'électricité sont deux conditions suffisantes (i.e., chacune d'elle est suffisante) pour que la maison soit habitable. » en logique propositionnelle avec les variables `eau`, `électricité` et `habitable`.
3. La formule propositionnelle $(a \vee \neg b) \wedge (b \vee \neg c) \wedge (c \vee \neg d) \wedge (d \vee \neg e) \wedge (\neg a) \wedge (e)$ est elle valide ? est elle satisfaisable ?

2.2 Logique du premier ordre (4 points)

1. On considère la formule $G = \exists Y \forall X ((p(X) \wedge q(Y)) \rightarrow (p(X) \vee q(Y)))$ et l'interprétation I sur le domaine $\{1, 2, 3, 4\}$ telle que $p(1) =$ `vrai`, $p(2) =$ `vrai`, $p(3) =$ `faux`, $p(4) =$ `faux`, $q(1) =$ `vrai`, $q(2) =$ `faux`, $q(3) =$ `vrai`, $q(4) =$ `faux`.
   – Quelle est la valeur de vérité de G pour cette interprétation I ?

– G est elle valide ?

2. Modélisez en logique du premier ordre la phrase « Tout homme n'ayant pas de voiture possède un chien ou un chat. » en utilisant les prédicats `homme/1`, `possede/2`, `chien/1`, `chat/1` et `voiture/1`.

3 Prolog et Caml

Il sera tenu compte de la *simplicité* et de la *concision* de vos programmes.

3.1 Listes (4 points)

1. Définissez une fonction caml `nn` qui, appliquée à une liste q d'entiers, retourne la plus petite valeur non nulle de q.
2. Spécifiez en prolog un prédicat `nn/2` tel que `nn(L,X)` soit satisfait si et seulement si `X` est le plus petit élément non nul de `L`.

3.2 Arbres (6 points)

Introduction et définitions

On considère le type `arbre` défini de la manière suivante : un `arbre` est
– soit une feuille, n'ayant aucun attribut,
– soit un noeud pour attributs :
  – un entier appelé *étiquette*,
  – un `arbre` appelé *fils gauche*,
  – un `arbre` appelé *fils droit*.

On appelle *chemin* d'un arbre t toute suite de noeuds reliant la racine de t à l'une de ses feuilles. On dit qu'un chemin est *homogène* de valeur x si et seulement si les valeurs des étiquettes de tous ses noeuds sont égales à x. On appelle *longueur* d'un chemin le nombre de noeuds de ce chemin.

Partie Caml

Écrivez une fonction `pch` qui, appliquée à un `arbre` t, un entier x, et un entier n, retourne un Booléen valant `true` si et seulement si t comporte un chemin homogène de longueur n et de valeur x.

Partie Prolog

Spécifiez un prédicat `pch/3` tel que `pch(T,X,N)` soit satisfait si et seulement si `T` possède au moins un chemin homogène de longueur `N` et de valeur `X`.