Numéro d'anonymat :

Examen de Programmation logique et fonctionnelle

Année 2014/2015 - Deuxième session

Université de Bourgogne - UFR Sciences et Techniques - L3

Vous devez répondre dans les cadres prévus à cet effet. Téléphones portables, calculatrices, ordinateurs et tablettes interdits. Notes personnelles (manuscrites ou imprimées) et documents de cours, TD et TP autorisés.

Logique propositionnelle.

La formule suivante est elle satisfaisable ? Est elle valide ? Justifiez brièvement votre réponse.

$$((a \to b) \to (\neg b \to \neg a)) \land ((\neg b \to \neg a) \to (a \to b))$$

10%

Proposez une modélisation en logique propositionnelle de la propriété suivante :

$$a + b + 2c < 3$$

où a, b et c sont des variables à valeurs binaires (0 ou 1). La valeur 0 sera représentée en logique propositionnelle par faux et la valeur 1 par vrai. La formule produite doit être satisfaite par toutes les interprétations qui respectent la propriété à modéliser et seulement par ces interprétations.

10%

Logique des prédicats.

Soit la formule du premier ordre suivante :

$$[(\exists \mathrm{X}\ (p(X)) \wedge \exists(\mathrm{X}\ (\neg p(X))] \rightarrow [\forall \mathrm{X}\ \mathrm{q}(\mathrm{X})]$$

Donnez une une interprétation sur le domaine {1,2} qui satisfait cette formule et une autre, sur le même domaine, qui la falsifie.

10%

On considère les prédicats suivants : h(X,Y) est vrai si et seulement si X a réalisé le haut fait Y, e(X) est vrai si et seulement si X est une élite, d(X,Y) est vrai si et seulement si X et Y sont deux valeurs distinctes (différentes).

Proposez une formule du premier ordre modélisant la propriété suivante : « Quiconque a réalisé au moins deux hauts faits est une élite ».

10%

PROLOG.

On représente par des faits Prolog une base de données sur les donateurs d'une association caritative.

Le prédicat donateur répertorie les personnes ayant déjà fait un don sous la forme donateur(<num>,<nom>,<prénom>), où <num> est un entier faisant office d'identifiant unique pour les donateurs. Par exemple le fait donateur(112,picsou,balthazar) indique qu'il y a un donateur nommé Balthazar Picsou ayant le numéro 112.

Le prédicat don répertorie les dons réalisés sous la forme don(<num>, <année>, <montant>) où <num> est l'identifiant du donateur, <année> l'année du don et <montant> le montant donné. Par exemple le fait don(112,2013,1) indique que le donateur numéro 112 a fait un don de 1 Euro en 2013.

Vous devez spécifier un prédicat req tel que req(N,P,A,M) soit satisfait si et seulement si un donateur ayant pour nom N et pour prénom P a fait un don de montant M au cours de l'année A.

Quel but permet d'afficher les noms, prénoms et montants des donateurs ayant fait un don en 2015 ***et*** en 2014 (pas nécessairement du même montant chaque année) ?

10%

Donnez la spécification d'un prédicat pos/3 tel que pos(X,K,L) est satisfait si et seulement si X apparaît dans la liste L à une position inférieure ou égale à K.

Exemples de cas d'échec du prédicat :

```
pos(3,2,[2,7,3,5]). (La valeur 3 n'apparaît pas en position 1 ni 2)
pos(3,4,[1,1,2,5]). (Il n'y a pas de valeur 3.)
```

Exemples de cas de réussite du prédicat :

```
pos(3,4,[1,2,3,5,7,8]).
pos(3,7,[3,3,7,5,3]).
```

10%

Lambda-calcul.

Donnez toutes les étapes de l'évaluation du terme suivant :

$$(\lambda x.\lambda y.yx)tu$$

10%

Donnez toutes les étapes de l'évaluation du terme suivant :

$$(\lambda x.\lambda y.yx)yx$$

10%

CAML.

Définissez la fonction CAML `inck` telle que si k est un entier positif ou nul et w une liste d'entier alors `inck w k` retourne une liste obtenue à partir de w en ajoutant 1 aux k premiers éléments de w. Si k vaut 0, la liste retournée est w. Si w comporte plus de k éléments alors la liste retournée est celle obtenue en ajoutant 1 à tous les éléments de w.
Par exemple inck [1;2;3;4;5] 3 retourne la liste [2,3,4,4,5].

15%

On définit le type suivant pour représenter un arbre binaire ayant deux types de feuilles : les feuilles les feuilles de type A et les feuilles de type B.
`type arbre = A | B | N of arbre * arbre;;`
Complétez la définition de la fonction CAML `nba` telle que si t est un arbre alors `nba t` retourne le nombre de feuilles de type A dans t.

```
let rec nba t =

  match t with

  |A ->

  |B ->

  |

  ;;
```

5%