Université de Marne-la-Vallée. Mai 2000
Examen de Logique. Licence et IUP d'informatique
Durée 2 heures. Tous les documents sont autorisés.

Tous les exercices sont indépendants.

Exercice 1.— On définit en Caml le type suivant `'a arbre` qui permet de manipuler des arbres binaires dont les étiquettes des noeuds sont formés d'éléments de type `'a`.

```
type 'a arbre =
    Vide
  | Noeud of 'a * 'a arbre * 'a arbre ;;
```

Dans les questions qui suivent, a), b), c) et d) sont indépendantes.

a) (2 points)
On suppose que les étiquettes des noeuds sont des entiers tels que, pour tout noeud, les étiquettes des noeuds du sous-arbre gauche sont inférieures à l'étiquette du noeud, elle-même inférieure aux étiquettes des noeuds du sous-arbre droit. L'arbre représenté est ainsi un arbre binaire de recherche. Ecrire une fonction Caml `insere` qui, étant donné un arbre binaire de recherche a et un élément x renvoie l'arbre binaire de recherche obtenu en insérant x dans a. Donner le type Caml de cette fonction.

b) (2 points)
Écrire une fonction Caml `interne` qui renvoie la liste des étiquettes des noeuds d'un arbre en ordre interne ou infixe. Donner le type Caml de cette fonction.

c) (2 points)
Écrire une fonction Caml `first` (resp. `second`) qui prend en argument une liste de couples et renvoie la liste formée des premières (resp. secondes) composantes des couples. On utilisera la fonctionnelle `map` dont on rapellera le type. Donner le type Caml de vos fonctions.

d) (2 points)
Les arbres dits cartésiens sont des arbres dont les étiquettes sont des couples d'entiers tels que l'arbre obtenu en masquant les secondes composantes des couples est un arbre binaire de recherche, et l'arbre obtenu en masquant les premières composantes des couples est un tas où le minimum est à la racine. On dit qu'un arbre cartésien a est inférieur à un arbre cartésien b si toutes les premières composantes des couples étiquettes de noeuds de a sont inférieures à toutes les premières composantes des couples étiquettes de noeuds de b. Ecrire une fonction Caml `fusion` qui prend en argument deux arbres cartésiens a et b, en supposant a inférieur à b, et renvoie un arbre cartésien dont les étiquettes des noeuds sont la réunion des étiquettes des noeuds de a et b. Donner le type Caml de cette fonction.

e) (2 points)
Ecrire une fonction Caml `affiche` qui prend en argument un arbre dont les étiquettes des noeuds sont des couples et renvoie un couple de listes. La première sera la liste des premières composantes des étiquettes des noeuds l'arbre en ordre infixe ou interne. La deuxième sera la liste des secondes composantes des étiquettes des noeuds de l'arbre en ordre infixe ou interne. Donner le type Caml de cette fonction.

Exercice 2.— Si a, b, c désignent des atomes (ou variables) du calcul des propositions, indiquer si les formules suivantes sont des tautologies, des antilogies ou ni l'un ni l'autre.

a) (2 points)
$(a \to (b \vee c)) \to ((a \to b) \vee (a \to c))$

b) (2 points)
$(a \to (b \wedge c)) \to ((a \to b) \wedge (a \to c))$

Exercice 3.— Si a, b, c désignent des atomes (ou variables) du calcul des propositions, montrer que :

a) (1 point)
$\vdash (a \to b) \to (c \to a) \to c \to b;$

b) (2 points)
$\vdash ((a \wedge b) \to c) \leftrightarrow (a \to (b \to c)).$

Le système de déduction à utiliser est le système de Hilbert présenté en cours. On pourra utiliser tous les résultats du cours (notamment le théorème de déduction), à l'exclusion toutefois du théorème de complétude.

c) (1 point)
Définir une fonction Caml du type :

```
('a -> 'b) -> ('c -> 'a) ->'c ->'b
```

Exercice 4.— (2 points) Question très difficile.
On appelle littéral un atome propositionnel p ou sa négation $\neg p$. On appelle 2-clause une clause qui est formée de deux littéraux exactement. Par exemple $(p \vee q)$, $(\neg p \vee r)$, $(\neg p \vee \neg r)$ sont des 2-clauses. Enfin on appelle 2-formule une formule qui est une conjonction de 2-clauses. Par exemple $F = (p \vee q) \wedge (p \vee \neg r) \wedge (\neg q \vee r)$ est une 2-formule. Trouver un algorithme polynomial pour tester si une 2-formule est satisfiable. On peut utiliser des graphes pour modéliser le problème.

Exercice 5.— (points en plus)
QUESTION SUBSIDIAIRE (ne traitez cette question que s'il vous reste du temps)
Donner une formule logique MSO (logique monadique du second ordre étudiée en cours) F qui est close et telle que le langage de mots finis défini par F est l'ensemble des mots sur un alphabet à deux lettres $\{a, b\}$ qui ont un nombre de a entre deux b qui est un multiple de 3.