Master MIAGE 2e année
UE MGE903 Systèmes experts (Ch. Retoré)
Examen du lundi 10 décembre 2007
Durée 1h30
Documents autorisés
Ce sujet comporte deux pages

A. Calcul propositionnel

Exercice I *(Modélisation, séquents)*

1. S'il pleut, je ne viens pas à vélo.
2. Si mon réveil sonne en retard, et que je ne viens pas à vélo, alors je suis en retard.
3. Mon réveil sonne en retard.
4. Je suis à l'heure.
5. Donc il ne pleut pas.

Ia. Modéliser dans le calcul propositionnel chacun de ces énoncés, en donnant au préalable la liste des propositions utilisées et leur signification.

Ib. Quel séquent exprime le fait que la dernière phrase soit conséquence des précédentes.

Ic. Montrer que la dernière phrase est effectivement conséquence des précédentes.

B. Cacul des prédicats, datalog, prolog

Tous les exercices de cette partie concernent la situation suivante.

Une lotissement comporte m maisons, chacune habitée par un certain nombre de personnes. Elles souhaitent organiser un dîner qui les réunissent toutes autour de n tables.

Les principes suivants sont supposés vrais:

1. Toute personne du lotissement habite une maison et une seule.
2. Pour apprécier une personne il faut la connaître.
3. Les occupants d'une même maison se connaissent et s'apprécient.

et on souhaite que le dîner proposé satisfasse:

4. Toute personne dîne à une et une seule table.
5. Dans un dîner réussi, chacun apprécie les personnes à sa table ou alors il ne les connaît pas.

On souhaite décrire cette situation en n'utilisant que des prédicats portant sur les personnes, et on interprétera les formules dans un domaine qui ne contient que des personnes. On utilisera donc le langage suivant:

- Des prédicats à une place $maison1$,$maison2$, ...: par exemple $maison1(anne)$ signifie que $anne$ habite la maison 1.
- Des prédicats à une place $table1$, $table2$, ...: par exemple $table3(fairouz)$ signifie que fairouz dîne à la table 3.
- Un prédicat à deux places: *connaît*. Par exempel *connaît*($anne$,$pablo$) signifie que $anne$ connaît $pablo$.
- Un prédicat à deux places: *apprécie*. Par exemple *apprécie*($anne$,$pablo$) signifie que $anne$ apprécie $pablo$.

Exercice II *(Modélisation dans le calcul des prédicats)*

On suppose pour cet exercice qu'il y a **trois maisons** et que le dîner comporte **deux tables**.

IIa. Ecrire dans le calcul des prédicats les principes 2, 3 et 5 dans le calcul des prédicats avec le langage de description donné.

IIb. Donner un modèle (une interprétation) qui satisfasse toutes les principes sauf le dernier.

IIc. Donner un modèle (une interprétation) qui satisfasse tous les principes.

IId. (difficile) Avec le langage choisi, pourquoi ne peut-on pas exprimer par une formule le principe1 lorsqu'on ne connaît pas le nombre de maisons?

Exercice III *(Datalog)*

On vérifiera que les clauses proposées soient correctes.

IIIa. Définir en Datalog une relation *connaissance_commune*(X,Y) qui donne les couples (X,Y) de personnes telles que X et Y connaissent une même personne.

IIIb. Définir en Datalog une relation *connaît_indirectement*(X,Y) qui donne les couples (X,Y) de personnes telles qu'il existe une suite de personnes dont la première est X et la dernière Y, $X = X_0,X_1,...X_{n-1},X_n = Y$, telles que chaque X_i connaît le suivant X_{i+1} pour $i = 1,...,n-1$.

IIIc. Définir en Datalog une relation qui donne les couples (X,Y) de personnes telles qu'il existe une suite de personnes dont la première est X et la dernière Y, $X = X_0,X_1,...X_{n-1},X_n = Y$, telles que chaque X_i connaît le suivant X_{i+1} pour $i = 1,...,n-1$ mais ne l'apprécie pas.

Exercice IV *(Prolog)*

On souhaite écrire un programme qui vérifie automatiquement si une organisation d'un dîner satisfait le principe 5. Afin de faciliter l'écriture du programme, on utilise des listes pour définir les tablées. On suppose définis les prédicats *connaît* et *apprécie*, ainsi que *connaît_pas* (qu'on pourrait facilement le définir en PROLOG à partir de *connaît*): par exemple *connaît_pas*(X,Y) signifie que X ne connaît pas Y). On dit qu'une personne X en supporte une autre Y si X apprécie Y ou si X ne connaît pas Y.

IVa. Définir un programme $supporte(X,Y)$ qui est vrai si X supporte Y.

IVb. Définir un programme $supporte\_liste(X,L)$ qui est vrai, si pour tout élément Y de L, X supporte Y.

IVc. Définir un programme $liste\_supporte\_liste(L1,L2)$ est vrai si chaque élément de $L1$ supporte chaque élément de $L2$.

IVd. Définir un programme $correct(L)$, qui vérifie si la liste L correspond à une table satisfaisant 5.