

Programmation logique et fonctionnelle

Examen terminal - première session

Modalités

Les documents de cours, TD et TP, ainsi que les notes personnelles, sont autorisés. Les calculatrices sont autorisées. Il sera tenu compte de la clarté et de la concision de vos réponses.

1 Logique (10 points)

1.1 Logique propositionnelle (5 points)

1.1.1 Validité et satisfaisabilité

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

$$((a \wedge \neg b) \vee (b \wedge \neg c) \vee (b \wedge \neg d)) \rightarrow (a \vee b \vee c)$$

1.1.2 Modélisation

On considère une urne, une boule rouge, une boule bleue, et une boule verte. Par convention, la variable b vaut **vrai** si et seulement si la boule bleue est dans l'urne, la variable r vaut **vrai** si et seulement si la boule rouge est dans l'urne, et la variable v vaut **vrai** si et seulement si la boule verte est dans l'urne.

1. Proposez une formule modélisant la propriété « certaines boules ne sont pas dans l'urne » ;
2. proposez une formule modélisant la propriété « s'il y a au moins une boule dans l'urne, alors la boule rouge est dans l'urne ».

1.2 Logique des prédicats (5 points)

1.2.1 Syntaxe

La formule $\forall X((X \wedge p(X)) \rightarrow q(X))$ est elle syntaxiquement correcte ? Si votre réponse est négative, expliquez pourquoi.

1.2.2 Validité et satisfaisabilité

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

$$(\forall X(p(X) \wedge \neg q(X))) \rightarrow (\exists X q(X))$$

1.2.3 Modélisation

A l'aide des prédicats suivants :

- **homme**/1 tel que **homme**(X) est **vrai** si et seulement si X est un homme ;
- **chien**/1 tel que **chien**(X) est **vrai** si et seulement si X est un chien ;
- **maitre**/2 tel que **maitre**(X,Y) est **vrai** si et seulement si X est le maître de Y ;
- **mordre**/2 tel que **mordre**(X,Y) est **vrai** si et seulement si X mord Y ;

modélisez la phrase : « Aucun homme ne peut être mordu par un chien ayant un maître. »

2 Prolog (10 points)

2.1 Faits et clauses (3 points)

Soit les faits et clauses Prolog suivants :

```
succ(zero, un).
succ(un, deux).
succ(deux, trois).
succ(trois, quatre).
succ( quatre, cinq).
myst(A,B) :- succ(A,Z), succ(Z,B).
```

Donnez tous les résultats des évaluations des buts suivants :

1. `myst(X,X)`.
2. `myst(X, quatre)`.
3. `myst(X,Y)`.

2.2 Listes (5 points)

2.2.1 Ajout d'une valeur aux éléments d'une liste

Complétez la spécification suivante pour que le prédicat `add(L,X,R)` permette la production d'une liste `R` obtenue en ajoutant `X` à chaque élément d'une liste `L`. Par exemple le but `add([1,1,3],2,R)` devra avoir pour résultat `R = [3,3,5]`.

```
add([], , ).
add([T|Q], X, [H|R]) :- H is , add( , , ).
```

2.2.2 Réduction d'une liste

Complétez la spécification suivante pour que le prédicat `red(L,R)` permette de produire la liste `R` obtenue en supprimant toutes les valeurs 0 du début de `L` jusqu'à la première valeur non nulle. Par exemple, le but `red([0,0,1,0,2],R)` doit avoir pour résultat `R = [1,0,2]`.

```
red([T|Q], ) :- T\=0.
red([0|Q], R) :- red( , ).
```

2.3 Négation et cut (2 points)

Ce prédicat destiné à calculer une valeur absolue ne fonctionne pas correctement. Corrigez sa spécification en ajoutant un coupe choix à l'emplacement approprié. Justifiez brièvement votre réponse.

```
vabs(X,X) :- X >= 0.
vabs(X,Y) :- Y is -X.
```