

TD Datalog

Exercice 1 : (il)logique ?

Question 1.1 : Traduire en formule logique : “*quand je suis en forme, je fais de bons TPs, donc quand je fais de mauvais TPs, c’est que je ne suis pas en forme.*”. Ce raisonnement est-il valide ?

CORRECTION:

la formule est $(P \Rightarrow Q) \Rightarrow (\text{not } Q \Rightarrow \text{not } P)$. C’est vrai (contraposée!)

Question 1.2 : Je dis : “*Si j’avais envoyé des textos sur le canapé, alors mon portable serait resté sur le canapé. Je n’ai pas envoyé de textos sur le canapé. Donc mon portable n’est pas resté sur le canapé!*” Traduisez ceci en formule logique. Est-ce que le raisonnement est valable ?

CORRECTION:

la formule est $((P \Rightarrow Q) \text{ and } \text{not } P) \Rightarrow \text{not } Q$. C’est faux ! Si P est faux et Q vrai, on le voit.

Question 1.3 : Je dis “*Si je ne réussis pas bien ce TP, je ne validerai pas l’UE, et si je ne valide pas l’UE, je vais déprimer, donc si je ne réussis pas bien ce TP, je vais déprimer.*” Traduisez ceci en formule logique. Est-ce vrai ?

CORRECTION:

la formule est $\left((P \Rightarrow Q) \text{ and } (Q \Rightarrow R) \right) \Rightarrow (P \Rightarrow R)$. C’est vrai !

Exercice 2 : L’amphi

Claus, Timoléon et Paul sont des étudiants. Timoléon a un 15, Claus un 9. Une bonne note est une note supérieure à 14. Tous les étudiants qui ont des bonnes notes, venaient en cours.

Question 2.1 : Formuler en logique la dernière phrase.

Question 2.2 : Décrire ces connaissances en Datalog.

```
% Question 1
etudiant(claus).
etudiant(timoleon).
```

```

etudiant(paul).

note(timoleon,15).
note(claus,9).

bonnenote(X) :- note(X,Y), Y >= 14.

veniramphi(X) :- bonnenote(X) .

```

Question 2.3 : Qui venait en cours ?

CORRECTION:

Trouver la requete, et expliquez comment elle est evaluee. Timoleon.
il faut faire une requete disjonctive!
expliquer, avec la double implication, pourquoi il n'y a pas de probleme en Datalog (qui memorise les faits deja trouves, et n'essaye pas de le retrouver)

Question 2.4 : Qui ne venait pas en cours ?

```

% qui ne venait pas en cours? moins facile de poser la bonne requete.

% 3 requetes. quelle est la bonne? expliquer les fautes.

% expliquer la CLOSED WORLD ASSUMPTION: tout ce qui n'est pas
%   inclut dans notre monde est suppose faux.
%   -> pas de note -> pas de BONNE note -> pas venu en amphi
%   -> note <14   -> pas de ''      -> ''

% expliquer les variables

%not(veniramphi(X))

%not(veniramphi(X)),etudiant(X)

%etudiant(X),not(veniramphi(X))

```

Exercice 3 : Les princesses (1/4)

Cet exercice repose sur *Le livre qui rend fou* de Raymon Smullyan.

Il y a très longtemps, et dans une contrée très lointaine, un roi légèrement sadique posait des problèmes logiques à ses prisonniers. Il les amenait en face de deux portes, sur lesquelles des affiches étaient collées. Derrière chacune des deux portes, il y a ou bien un tigre, ou

bien une princesse. Les prisonniers doivent choisir une des deux portes et l'ouvrir. S'ils tombent sur un tigre, ils sont dévorés par l'animal affamé. S'ils tombent sur une princesse, alors ils peuvent s'enfuir avec elle. Toutes les combinaisons sont possibles (y compris deux tigres derrière les deux portes). Tout le problème consiste à ouvrir la bonne porte. Une difficulté supplémentaire, c'est que les affiches ne disent pas forcément la vérité! Le roi dit au premier prisonnier : "soit les deux affiches disent la vérité, soit elles mentent toutes les deux". Voici ce que le prisonnier pouvait lire :

1 : Il y a un tigre dans cette cellule,
ou il y a une princesse dans l'autre

2 : Il y a une princesse dans l'autre
cellule

Question 3.1 : Discutez en mini-groupes pendant 5 minutes. Comment faut-il décider? Faites un vote sur les différentes propositions.

Question 3.2 : Ecrivez une formule logique qui exprime l'idée que chaque cellule contient soit un tigre, soit une princesse.

CORRECTION:

$(T(1) \text{ or } P(1)) \text{ and } (T(2) \text{ or } P(2))$

Question 3.3 : Traduisez les deux affiches en formules logiques

CORRECTION:

$(T(1) \text{ or } P(2)) \text{ and } P(1)$

montrer que cela implique qu'il y a des princesses partout.

Question 3.4 : Ecrivez ensuite la **négation** de ces deux formules

CORRECTION:

$\text{not}(T(1) \text{ or } P(2)) \text{ and } \text{not}(P(1))$

montrer que ce n'est pas possible.

On voudrait utiliser **DataLog** pour résoudre le problème. Pour cela il faut écrire un prédicat **contenu(X)** qui est vrai si *X* est l'un des deux constantes **tigre** ou **princesse**. Si l'on soumet la requête :

```
DES> contenu(X), contenu(Y).
```

On doit récupérer :

```
answer(princesse, princesse),  
answer(princesse, tigre),  
answer(tigre, princesse),  
answer(tigre, tigre)
```

Question 3.5 : Ecrivez le prédicat **contenu**.

Ensuite, on veut écrire deux prédicats **affiche_1(X,Y)** et **affiche_2(X,Y)** qui traduisent les deux affiches. Ce qu'on veut est :

```
DES> affiche_1(X,Y).
{
  affiche_1(princesse , princesse) ,
  affiche_1(tigre , princesse) ,
  affiche_1(tigre , tigre)
}
DES> affiche_2(X,Y).
{
  affiche_2(princesse , princesse) ,
  affiche_2(princesse , tigre)
}
```

On voit par exemple que la combinaison (princesse,tigre) est interdite par la première affiche.

Question 3.6 : Définissez les deux prédicats `affiche_1` et `affiche_2`.

Question 3.7 : Définissez un prédicat `solutions(X,Y)` qui liste toutes les combinaisons qui satisfont les deux affiches à la fois. Quelles sont les solutions ?

Question 3.8 : On se rappelle que les deux affiches peuvent soit dire toutes les deux la vérité, soit mentir toutes les deux. Définissez un prédicat `alt_solutions(X,Y)` qui liste toutes les combinaisons admissibles en supposant que les deux affiches mentent. Quelles sont les solutions ?

Corrigé :

```
contenu(tigre).
contenu(princesse).

affiche_2(princesse ,Y) :- contenu(Y).
affiche_1(tigre ,Y)      :- contenu(Y).
affiche_1(X,princesse) :- contenu(X).

solutions(X,Y) :- affiche_1(X,Y), affiche_2(X,Y).

alt_solutions(X,Y) :- contenu(X), contenu(Y), \
                        not(affiche_1(X,Y)), not(affiche_2(X,Y)).
```