Expression Logique et Fonctionnelle . . . Évidemment

TP4 : Les listes

1 Listes et récursivité

Le prédicat `a2b/2` a deux arguments et est valide si
- le premier est une liste de constantes `a` ;
- le deuxième est une liste de constantes `b` de même longueur.

Ainsi on aura les requêtes et réponses suivantes :

```
?- a2b([a,a,a],[b,b,b]).
yes
?- a2b([a,a,a,a],[b,b,b]).
no
?- a2b([a,t,a,a],[b,b,b,c]).
no
?- a2b([a,a,a,a],X).
X=[b,b,b,b]
```

Question 1 Définissez le prédicat `a2b/2`.

Question 2 À l'aide de la commande **trace**, analyser la résolution des requêtes closes suivantes
- `a2b([a,a,a,a],[b,b,b,b])` ;
- `a2b([a,a,a,a],[b,b,b])` ;
- `a2b([a,c,a,a],[b,b,5,4])`.

Testez vos propres requêtes !

Question 3 À l'aide de la commande **trace**, analysez la résolution de requêtes non close (i.e. avec variables) telles que
- `a2b([a,a,a,a],X)` ;
- `a2b(X,[b,b,b,b])`.

Question 4 Quel est le résultat de la requête `a2b(X,Y)` ?

Question 5 Analysez vous-même la résolution du prédicat `member` vu en cours pour différentes requêtes closes et non closes. Assurez-vous de bien comprendre les traces et d'être capable de les prédire !

2 Combinaisons

Question 6 Écrivez le prédicat ternaire `combine1/3` qui combine les éléments de deux listes dans une troisième :

```
?-combine1([a,b,c],[1,2,3],X).
X = [a,1,b,2,c,3]
?-combine1([foo,bar,yip,yup],[glub,glab,glib,glob],X).
X = [foo,glub,bar,glab,yip,glib,yup,glob]
```

Question 7 Écrivez un prédicat ternaire `combine2/3` qui combine maintenant deux listes dans une troisième de la façon suivante :

```
?-combine1([a,b,c],[1,2,3],X).
X = [[a,1],[b,2],[c,3]]
?-combine1([foo,bar,yip,yup],[glub,glab,glib,glob],X).
X = [[foo,glub],[bar,glab],[yip,glib],[yup,glob]]
```

Question 8 Écrivez enfin un prédicat ternaire `combine3/3` qui combine maintenant deux listes dans une troisième de la façon suivante :

```
?-combine1([a,b,c],[1,2,3],X).
X = [paire(a,1),paire(b,2),paire(c,3)]
?-combine1([foo,bar,yip,yup],[glub,glab,glib,glob],X).
X = [paire(foo,glub),paire(bar,glab),paire(yip,glib),paire(yup,glob)]
```

3 Ensembles

Question 9 Écrivez un prédicat `sous_ensemble/2` satisfait lorsque la liste donnée en premier argument est un sous-ensemble de la liste donnée en deuxième argument.
Question 10 Écrivez un prédicat `parties_d_ensemble/2` satisfait lorsque la liste donnée en premier argument est un ensemble de parties de la liste donnée en deuxième argument.
Question 11 Écrivez un prédicat `union/3` satisfait lorsque la liste donnée en premier argument est l'union des listes données en deuxième et troisième arguments.
Question 12 Écrivez un prédicat `intersection/3` satisfait lorsque la liste donnée en premier argument est l'intersection des listes données en deuxième et troisième arguments.