TP de programmation fonctionnelle et logique

TP 4 : le compte est bon

Ce TP est à rendre impérativement pour le vendredi 7 janvier, délai de rigueur. Vous devez m'envoyer votre programme par mail à l'adresse : `vivien@icps.u-strasbg.fr`. Vous pouvez réaliser ce TP seul ou en binôme.

Il s'agit ici, une fois de plus, de résoudre le problème du compte est bon. Nous ne nous intéresserons ici qu'aux solutions exactes (nous ne rechercherons donc pas la meilleure approximation).

Soit une formule permettant d'obtenir la valeur cible. La liste des entiers figurant dans cette formule est une permutation d'une sous-liste de la liste d'entiers arguments. Le principe du programme que vous allez devoir écrire est de fabriquer toutes les permutations de toutes les sous-listes possibles de la liste argument. Étant donnée une de ces permutations, les entiers apparaissant dans la formule à construire et l'ordre d'apparition de ces entiers sont fixés. Il ne reste plus qu'à combiner entre eux ces entiers au moyen des quatre opérations.

Conseil : pensez à tester vos prédicats au fur et à mesure.

1. Écrivez un prédicat `sousliste(A,B)` qui est vrai si `A` est une sous-liste de `B`.
2. (Ré-)écrivez un prédicat `permutation(A,B)` qui est vrai si la liste `B` est une permutation de la liste `A` dont la valeur est donnée en argument.
3. Écrivez un prédicat `décompose(A,B,C)` qui est vrai si et seulement si la liste `C` est la concaténation des listes `A` et `B` et si ni la liste `A`, ni la liste `B` n'est vide. Vous pouvez réutilisez le prédicat prédéfini `append` de concaténation des listes.
4. Écrivez un prédicat `construit_chaîne(A,B,Op,C)` où `A` et `B` sont les chaînes de caractères représentants deux nombres à combiner par l'opération représentée par la chaîne de caractères `Op`. `C` est la chaîne de caractères résultante. Si `A = "3"`, `B="4"` et `Op="*"`, nous voulons obtenir `C="(3*4)"`.

   En SWI-Prolog les chaînes de caractères sont représentées par des tableaux d'entiers, comme l'illustre l'exemple suivant :

   ```
   ?- X = "bonjour".

   X = [98, 111, 110, 106, 111, 117, 114]

   Yes
   ```

   Pour obtenir les chaînes de caractères sous une forme lisible, nous devons obligatoirement avoir recours aux fonctions ci-dessous :

   ```
   ?- string_to_list(X,"bonjour").

   X = "bonjour"

   Yes
   ```

   ```
   ?- string_to_atom(X,3).

   X = "3"

   Yes
   ```

   ```
   ?- string_to_list(X,"bon"),string_to_list(Y,"jour"),string_concat(X,Y,Z).

   X = "bon"
   Y = "jour"
   Z = "bonjour" ;

   No
   ```

5. Écrivez un prédicat `combine(A,B,C)` où `A`, `B` et `C` sont des listes à deux éléments : un nombre et la formule, sous forme de chaîne de caractères, permettant de l'obtenir. `A` et `B` sont les deux nombres à combiner et `C` est le résultat de cette combinaison via l'une des quatre opérations.

   Indication : on pourra utiliser l'opérateur `mod`.

6. Écrivez un prédicat `valeur(L,D)` où `L` est une liste d'entiers et `D` une liste à deux éléments : un entier et la formule, sous forme de chaîne de caractères, permettant de l'obtenir. Le prédicat `valeur(L,D)` doit construire à partir de `L` toutes les valeurs de `D`.

   Indication : ce prédicat décomposera, si possible, la liste `L` en deux sous-listes sur lesquelles le prédicat sera appelé récursivement.

7. Combinez les prédicats précédemment écrits pour obtenir un prédicat `lecompteestbon(L,C,D)` qui, étant donnés un entier cible `C` et une liste d'entiers `L`, construit, si possible, une liste à deux éléments constituée de l'entier cible et d'une formule, sous forme de chaîne de caractères, permettant de l'obtenir à partir des éléments de la liste `L`.