DS 2

Option informatique, première année

Julien Reichert

Exercice 1 : Écrire en Caml une fonction qui calcule la somme des éléments d'un tableau d'entiers **en utilisant impérativement une méthode diviser pour régner** (même si cela n'a aucune utilité pour la complexité. Ensuite même exercice pour le maximum d'un tableau (les éléments du tableau supporteront la comparaison avec >) que l'on pourra supposer non vide (pas besoin de le vérifier).

Exercice 2 : Considérons la suite de Fibonacci, décrite par $u_0 = 0, u_1 = 1$ et $u_{n+2} = u_{n+1} + u_n$ pour $n \geq 0$. Montrer que si A est la matrice $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, alors le premier élément du vecteur $A^n \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ est u_n. Écrire alors une fonction qui détermine la valeur du terme d'indice `n` de la suite de Fibonacci, où `n` est l'argument, en temps logarithmique en la valeur de `n`, c'est-à-dire en utilisant l'exponentiation rapide appliquée aux matrices.

Pour les deux exercices suivants, nous utiliserons des arbres **binaires** définis par le type suivant : `type 'a arbin = Vide | Noeud of 'a arbin * 'a * 'a arbin;;` et pour les deux d'après le type sera `type arpasbin = N of int * arpasbin list;;` pour des arbres d'entiers non vides d'arité quelconque.

Exercice 3 : Écrire en Caml une fonction qui calcule la profondeur d'un arbre binaire. La convention est : profondeur `-1` pour un arbre vide, profondeur `0` pour un arbre réduit à sa racine et dans le cas général longueur maximale d'une branche (qu'on résumera à un chemin entre deux nœuds dans ce sujet, mais ici vu qu'on cherche à maximiser on ne s'intéresse qu'aux branches allant de la racine à une feuille), où la longueur d'une branche est le nombre de liens donc le nombre de nœuds traversés moins un.

Exercice 4 : Déterminer (avec preuve) ce que la fonction suivante calcule. La mention de « branche » est attendue.

```
let f arbre = let rec aux i a = match a with
  | Vide -> 0
  | Noeud(g, _, d) -> i + aux (i+1) g + aux (i+1) d
  in aux 0 arbre;;
```

Exercice 5 : Écrire en Caml une fonction qui calcule la somme des éléments d'un arbre d'entiers d'arité quelconque.

Exercice 6 : Écrire en Caml une fonction qui détermine si tout nœud d'un arbre d'entiers d'arité quelconque a pour étiquette la somme des étiquettes de ses fils. Un nœud sans enfant doit donc avoir 0 pour étiquette.

Exercice 7 : Écrire en Caml une fonction qui effectue une rotation d'un **huitième** de tour dans le sens horaire d'un tableau de tableaux vu comme une matrice rectangulaire. La taille du tableau ne pourra jamais être zéro, de même que les tailles communes de tous les éléments du tableau.

Par exemple, pour le tableau de tableaux `[|[|0; 1; 2|];[|3; 4; 5|]|]`, la fonction retournera (attention, ce ne sera plus rectangulaire !) `[|[|0|];[|3; 1|];[|4; 2|];[|5|]|];;`

Exercice 8 : Dans une société futuriste, les citoyens sont évalués sur une base biennale, et du résultat de cette évaluation dépend leur poste et leur nom.[1] Ainsi, le meilleur citoyen est le « Président » et porte pour nom A, s'ensuivent les vingt-cinq ministres dans l'ordre alphabétique, puis les « 2-lettrés » de AA à ZZ, et ainsi de suite pour l'ensemble de la population[2]. Écrire une fonction qui transforme un classement en nom, et une fonction réciproque. Le nom sera une chaîne de caractères, constituée uniquement de lettres capitales de l'alphabet français.

1. Le thème de cet exercice reprend le contexte d'un roman d'Alain Damasio.
2. les noms sont limités à 5 lettres dans le roman, pas dans l'exercice

Exercice 9 : Avec les deux types ci-dessous pour représenter les formules propositionnelles, écrire deux fonctions transformant une formule représentée selon un type en la même formule représentée selon l'autre, une par sens.

```
type f_b = Vrai | Faux | Var of string | Non of f_b | Et of f_b list | Ou of f_b list;;

type feuille_formule = V | F | Vb of string;;
type operateur = And | Or | Not;;
type arbre_formule = Feuille of feuille_formule | Noeud of operateur * arbre_formule list;;
```

. . . et un exercice de logique de CCINP !

Imaginez-vous ethnologue. Vous étudiez une peuplade primitive qui présente un comportement manichéen extrême : lorsque plusieurs personnes participent à une même conversation sur un sujet donné, elles vont toutes avoir le même comportement manichéen tant que la conversation reste sur le même sujet, c'est-à-dire que toutes les affirmations seront soit des vérités, soit des mensonges. Par contre, si le sujet de la conversation change, la nature des affirmations, soit mensonge, soit vérité, peut changer, mais toutes les affirmations seront de la même nature tant que le sujet ne changera pas à nouveau. Pour être autorisé à séjourner dans cette peuplade, vous devez respecter cette règle. Vous participez à une conversation avec trois de leurs membres que nous appellerons X, Y et Z. Ceux-ci vous indiquent comment rejoindre leur village. Si vous n'arrivez pas à le rejoindre, vous ne serez pas autorisé à y séjourner. Le premier sujet abordé est la région dans laquelle se trouve le village :

— X indique : « Le village se trouve dans la vallée » ;
— Z réplique : « Non, il ne s'y trouve pas » ;
— X reprend : « Ou alors dans les collines ».

Nous noterons V et C les variables propositionnelles associées à la région dans laquelle se trouve le village. Nous noterons X_1 et Z_1 les formules propositionnelles correspondant aux affirmations de X et de Z sur le premier sujet. Puis, le second sujet est abordé : le chemin qui permet de rejoindre le village dans la région concernée.

— X dit : « Le chemin de gauche conduit au village » ;
— Z répond : « Tu as raison » ;
— X complète : « Le chemin de droite y conduit aussi » ;
— Y affirme : « Si le chemin du milieu y conduit, alors celui de droite n'y conduit pas » ;
— Z indique : « Celui du milieu n'y conduit pas ».

Nous noterons G, M, D les variables propositionnelles correspondant respectivement au fait que le chemin de gauche, du milieu et de droite, conduit au village. Nous noterons X_2, Y_2 et Z_2 les formules propositionnelles correspondant aux affirmations de X, de Y et de Z sur le second sujet.

Question 1 : Représenter le comportement manichéen des interlocuteurs dans le premier sujet abordé sous la forme d'une formule du calcul des propositions dépendant des formules propositionnelles X_1 et Z_1.

Question 2 : Représenter les informations données par les participants sous la forme de deux formules du calcul des propositions X_1 et Z_1 dépendant des variables V et C.

Question 3 : En utilisant la résolution avec les propriétés des opérateurs booléens et les formules de De Morgan en calcul des propositions, déterminer dans quelle région vous devez vous rendre pour rejoindre le village.

Question 4 : Représenter le comportement manichéen des interlocuteurs dans le second sujet abordé sous la forme d'une formule du calcul des propositions dépendant des formules propositionnelles X_2, Y_2 et Z_2.

Question 5 : Représenter les informations données par les participants sous la forme de trois formules du calcul des propositions X_2, Y_2 et Z_2 dépendant des variables G, M et D.

Question 6 : En utilisant la résolution avec les tables de vérité en calcul des propositions, déterminer quel chemin vous devez suivre pour rejoindre le village.

Question 7 : En admettant que les trois participants aient menti, pouviez-vous prendre d'autres chemins ? Si oui, le ou lesquels ?