Durée : 3 heures

Exercice 1

1. On considère la fonction récursive suivante, qui calcule les termes d'une suite $(u_n)_{n\in\mathbb{N}}$:

```
let rec u n =
  match n with
    | 0 -> 4
    | 1 -> 7
    | _ -> u (n-1) + 2 * (u (n-2))
;;
```

   (a) Que valent u_0 et u_1 ? Expliciter une relation de récurrence vérifiée par la suite u.

   (b) Montrer que l'appel `u n` termine pour toute entrée $n \in \mathbb{N}$.

   (c) Déterminer le nombre d'opérations arithmétiques nécessaires au calcul de `u n`.

   (d) Quelle est la complexité de la fonction `u` ?

2. Une première amélioration :

   (a) Ecrire une fonction récursive `u_term` qui prend en entrée trois entiers a, b et n et qui, pour $m \in \mathbb{N}$, renvoie u_{n+m} lorsque $a = u_m$ et $b = u_{m+1}$.

   (b) En déduire une fonction `u2` calculant les termes de la suite u plus rapidement que `u`.

   (c) Déterminer le nombre d'opérations arithmétiques nécessaires au calcul de `u2 n`.

   (d) Quelle est la complexité de la fonction `u2` ?

3. Ecrire une fonction `u3` non récursive calculant les termes de la suite u.

Exercice 2

1. Ecrire une fonction non récursive `indice_min_tableau` de type `'a array -> int` qui prend en entrée un tableau d'éléments appartenant à un ensemble ordonné et qui renvoie l'indice du premier des plus petits éléments de ce tableau. la complexité doit être linéaire en la taille du tableau. Démontrer sa correction.
2. Ecrire une fonction récursive `max_liste` de type `'a list -> 'a` qui prend en entrée une liste d'éléments appartenant à un ensemble ordonné et qui renvoie le plus grand élément de cette liste. la complexité doit être linéaire en la taille de la liste. Démontrer sa correction.

Exercice 3

1. Ecrire une fonction récursive `insertion x l` de type `'a -> 'a list -> 'a list` qui à un élément `x` et une liste `l` d'éléments du même type triée dans l'ordre croissant renvoie une liste contenant les éléments de `l` avec en plus `x` et toujours triée dans l'ordre croissant.
2. Ecrire une fonction récursive `tri_insertion l` de type `'a list -> 'a list` qui à une liste `l` renvoie une liste composée des même éléments mais triée dans l'ordre croissant. Cette fonction devra appeler la fonction précédente.
3. Donner la complexité de `tri_insertion l` dans le pire cas et le meilleur cas en fonction de la taille de la liste `l`. Justifier votre réponse.

Exercice 4

Soit Σ un ensemble fini contenant deux éléments particulier [et] appelés respectivement parenthèse gauche et parenthèse droite. On définit alors l'ensemble E les mots bien parenthésés sur Σ par induction :

— le mot vide, noté ϵ appartient à E ;

— pour tout $x \in \Sigma \setminus \{[,]\}$ on a $x \in E$;

— et enfin, pour tout éléments u et v de E on a $[u]v$ qui appartient également à E.

Il s'agit donc d'un sous-ensemble de l'ensemble des suites finies d'éléments de E. On notera $x_1x_2\dots x_n$ une telle suite de n éléments dans laquelle chaque x_i appartient à σ et on remarquera que ϵ désigne simplement la suite vide.

On définit alors une fonction p de poids sur l'ensemble Σ de la manière suivante :

$$p([) = 1, \quad p(]) = -1, \quad p(x) = 0 \text{ si } x \in \Sigma \setminus \{[,]\}$$

On étend alors cette fonction à toutes les suites finies d'éléments de Σ en posant :

$$p(x_1x_2\dots x_n) = p(x_1) + p(x_2) + \dots + p(x_n)$$

En particulier $p(\epsilon) = 0$.

Enfin, si $u = x_1x_2\dots x_n$ est une suite finie d'éléments de Σ, on appelle préfixe de taille p de u une suite de p éléments de Σ coïncidant avec les p premiers éléments formant u (c'est à dire $x_1x_2\dots x_p$).

1. Montrer que tout mot bien parenthésé $u \in E$ satisfait les deux proriétés suivantes :
   (a) $p(u) = 0$;
   (b) pour tout préfixe v de u, on a $p(v) \geqslant 0$.
2. Montrer, réciproquement, que si u est une suite finie d'éléments de Σ satisfaisant les deux propriétés (a) et (b) de la question précédente, alors u est un mot bien parenthésé.
3. En déduire un algorithme qui permet de tester si une suite finie d'éléments de Σ est un mot bien parenthésé.

Exercice 5

Dans un futur lointain, l'espèce humaine a découvert une autre espèce consciente. L'étude de cette espèce a permis de découvrir qu'elle est capable de percevoir si quelqu'un dit la vérité ou un mensonge. Les membres de cette espèce respectent les règles de politesse suivantes lors des discussions au sein d'un groupe : « Les orateurs doivent rester constants au cours d'une discussion : soit ils disent toujours la vérité, soit ils mentent toujours. De plus, si un orateur dit la vérité alors l'orateur suivant doit également dire la vérité. Si le sujet de la discussion change, les orateurs sont libres de changer leurs comportements. ».

Vous assistez à une discussion sur les moyens d'attaque et de défense que peut posséder la faune de cette planète entre trois membres de cette espèce que nous appellerons A, B et C.

A « Le kjalt peut avoir un dard ou des griffes. »
B « Non, il n'a pas de dard. »
C « Il a des pinces et des griffes. »

Nous noterons D, G et P les variables propositionnelles associées au fait qu'un kjalt possède respectivement un dard, des griffes et des pinces.

Nous noterons A_1, B_1 et C_1 les formules propositionnelles associées aux déclarations de A, B et C dans cette première discussion.

C quitte le groupe et la discussion change de sujet pour parler de la flore de la planète.

A « Un lyop peut être de couleur mauve mais pas de couleur jaune. »
B « Il ne peut pas être de couleur verte. »
A « Il ne peut être de couleur verte que s'il peut être de couleur jaune. »

Nous noterons J,M et V les variables propositionnelles associées au fait qu'un lyop peut être respectivement de couleur jaune, mauve et verte.

Nous noterons A_2, A_3 et B_2 les formules propositionnelles associées aux déclarations de A et B dans cette seconde discussion.

1. Représenter les règles de politesse appliquées à la première discussion sous la forme d'une formule du calcul des propositions dépendant des formules A_1, B_1 et C_1.

2. Représenter les informations données par les participants de la première discussion sous la forme de formules du calcul des propositions A_1, B_1 et C_1 dépendant des variables D, G et P.
3. En utilisant le calcul des propositions (résolution avec les formules de De Morgan), déterminer le (ou les) moyen(s) d'attaque et de défense que peut posséder un kjalt.
4. Représenter les règles de politesse appliquées à la seconde discussion sous la forme d'une formule du calcul des propositions dépendant des formules A_2, A_3 et B_2.
5. Représenter les informations données par les participants lors de la seconde discussion sous la forme de trois formules du calcul des propositions A_2, A_3 et B_2 dépendant des variables J, M et V .
6. En utilisant le calcul des propositions (résolution avec les tables de vérité), déterminer la (ou les) couleur(s) possible(s) pour un lyop.