

DS 3

Option informatique, deuxième année

Julien REICHERT

Durée : 3 heures.

Partie I – Logique et calcul des propositions

De nombreux travaux sont réalisés en Intelligence Artificielle pour construire un programme qui imite le raisonnement humain et soit capable de réussir le test de Turing, c'est-à-dire qu'il ne puisse pas être distingué d'un être humain dans une conversation en aveugle. Vous êtes chargé(e)s de vérifier la correction des réponses données par un tel programme lors des tests de bon fonctionnement. Dans le scénario de test considéré, le comportement attendu est le respect de la règle suivante : pour chaque question, le programme répondra par trois affirmations dont une seule sera correcte. Nous noterons A_1 , A_2 et A_3 les propositions associées aux affirmations effectuées par le programme.

Question 1 : Représenter le comportement attendu sous la forme d'une formule du calcul des propositions qui dépend de A_1 , A_2 et A_3 .

Premier cas

Vous demandez au programme : « Quels éléments doivent contenir les aliments que je dois consommer pour préserver ma santé ? »

Il répond les affirmations suivantes :

A_1 Consommez au moins des aliments qui contiennent des glucides, mais pas des lipides !

A_2 Si vous consommez des aliments qui contiennent des glucides alors ne consommez pas d'aliments qui contiennent des lipides !

A_3 Ne consommez aucun aliment qui contient des lipides !

Nous noterons G (resp. L), les variables propositionnelles qui correspondent au fait de consommer des aliments qui contiennent des glucides (resp. lipides).

Question 2 : Exprimer A_1 , A_2 et A_3 sous la forme de formules du calcul des propositions. Ces formules peuvent dépendre des variables G et L .

Question 3 : En utilisant le calcul des propositions (résolution avec les formules de De Morgan), déterminer ce que doivent contenir les aliments que vous devez consommer pour préserver votre santé.

Second cas

Vous demandez au programme : « Quelles activités dois-je pratiquer si je veux préserver ma santé ? »

Suite à une coupure de courant, la dernière affirmation est interrompue.

A_1 Ne faites des activités sportives que si vous prenez également du repos!

A_2 Si vous ne faites pas d'activité intellectuelle alors ne prenez pas de repos!

A_3 Prenez du repos ou faites des activités...

Nous noterons S , I et R les variables propositionnelles qui correspondent au fait de faire des activités sportives, des activités intellectuelles et de prendre du repos.

Question 4 : Exprimer A_1 , A_2 et A_3 sous la forme de formules du calcul des propositions. Ces formules peuvent dépendre de S , I et R .

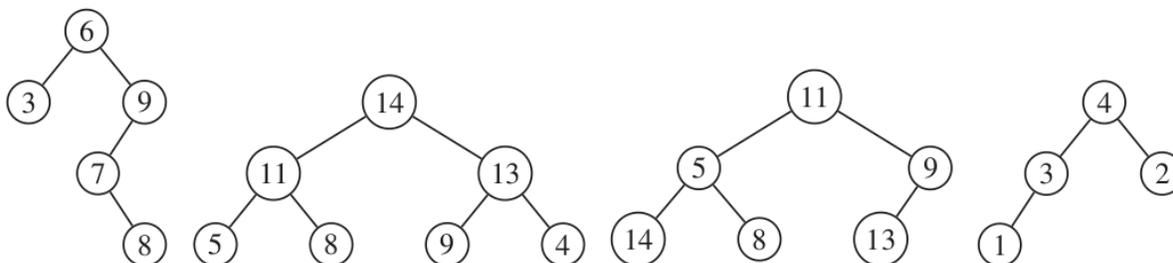
Question 5 : En utilisant le calcul des propositions (résolution avec les tables de vérité), déterminer quelle(s) activité(s) vous devez pratiquer pour préserver votre santé.

Partie II – Algorithmique et programmation

Question 6 : Écrire en Caml une fonction récursive de signature `lire_liste : int -> int list -> int` telle que `lire_liste i l` retourne le i -ième élément de l , pour i entre 1 et la taille de l .

On considère dans la suite des arbres binaires d'entiers, exprimés en Caml à l'aide du type construit `type arbre = Vide | Noeud of arbre * int * arbre;;`.

Voici quelques exemples de représentations graphiques d'arbres binaires d'entiers, en ne dessinant pas de Vide.



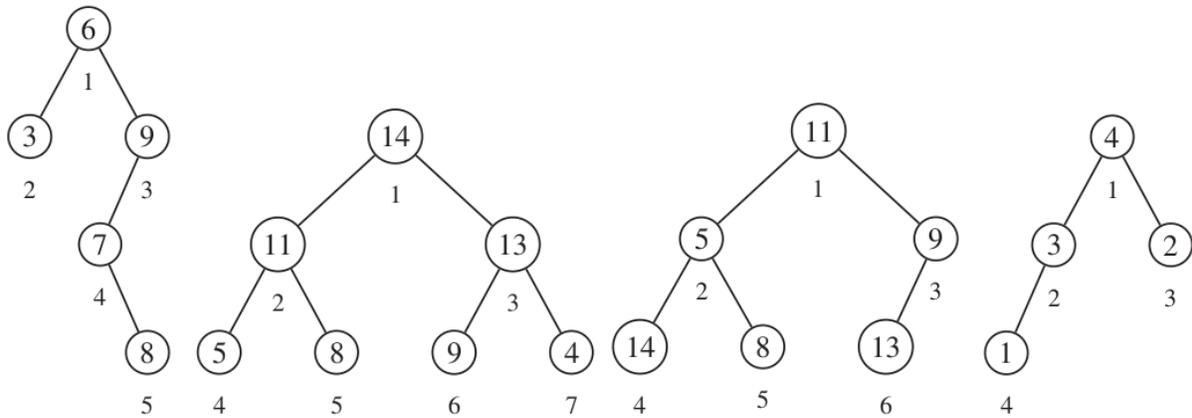
Question 7 : Donner l'expression en Caml des quatre arbres représentés.

Soient a un arbre binaire et n un nœud de a . La hauteur de n dans a est égale au nombre de nœuds du chemin sans cycle le plus long reliant n à un sous-arbre vide. Nous la noterons $\eta(n)$. La hauteur d'un arbre est la hauteur de sa racine. Nous associerons la hauteur 0 à l'arbre binaire vide. La profondeur de n dans a est la différence entre la hauteur de a et celle de n . Un niveau dans un arbre binaire est une séquence de gauche à droite de tous les nœuds de même profondeur dans l'arbre.

Question 8 : Donner une définition mathématique de la hauteur d'un arbre binaire impliquant l'arbre vide, le fils gauche et le fils droit de l'arbre.

Question 9 : Soit un ensemble non vide d'étiquettes donné, quelle est la structure d'un arbre binaire contenant ces étiquettes et dont la hauteur est maximale? Quelle est la structure d'un arbre binaire contenant ces étiquettes et dont la hauteur est minimale? Justifier.

La numérotation hiérarchique des nœuds d'un arbre binaire a consiste à associer à chaque nœud un numéro entre 1 et $|a|$ par un parcours en largeur partant de la racine (numéro 1) et en parcourant chaque niveau de gauche à droite jusqu'au dernier nœud : le plus profond et le plus à droite (numéro $|a|$). Nous noterons $N_i(a)$ le nœud de a de numéro i . Dans les exemples suivants, le numéro de chaque nœud sera noté en-dessous de son étiquette.



Question 10 : Écrire en Caml une fonction récursive (ou faisant appel à des sous-fonctions récursives) de signature `lire : int -> arbre -> int` telle que `lire i a` retourne $N_i(a)$ si i est entre 1 et $|a|$. Cette fonction devra au plus parcourir une seule fois chaque élément de l'arbre a .

Question 11 : Écrire en Caml une fonction récursive (ou faisant appel à des sous-fonctions récursives) qui vérifie si un arbre binaire est un tas (on ne demande pas dans un premier temps de vérifier que l'ordre militaire est respecté). Cette fonction devra au plus parcourir une seule fois chaque élément de l'arbre a .

Un arbre binaire complet est un arbre binaire dont tous les niveaux sont complets, c'est-à-dire que tous les nœuds d'un même niveau ont deux fils non vides sauf les nœuds du niveau le plus profond qui n'ont aucun fils (c'est-à-dire deux fils vides).

Question 12 : Montrer que, dans un arbre binaire complet non vide, le niveau de profondeur p contient 2^p nœuds. En déduire le nombre de nœuds d'un arbre binaire complet en fonction de sa hauteur.

Question 13 : Écrire en Caml une fonction récursive (ou faisant appel à des sous-fonctions récursives) qui détermine si un arbre binaire est complet ou, à défaut, s'il respecte l'ordre militaire¹. On retournera donc par exemple une chaîne de caractères explicite. Cette fonction devra au plus parcourir une seule fois chaque élément de l'arbre a .

Question 14 : Soit un nœud de numéro n dans le niveau de profondeur p d'un arbre binaire respectant l'ordre militaire, calculer le nombre de nœuds qui se trouvent à sa gauche dans le niveau de profondeur p .

Question 15 : Dans le niveau de profondeur $p + 1$ d'un arbre binaire respectant l'ordre militaire, quel est le nombre de nœuds qui se trouvent à la gauche des fils du nœud de numéro n (n faisant partie du niveau de profondeur p) ?

Question 16 : Soit un nœud de numéro n d'un arbre binaire respectant l'ordre militaire, calculer les numéros de ses fils gauche et droit. En déduire le numéro du père du nœud de numéro n dans un tel arbre.

Question 17 : Écrire en Caml une fonction récursive (ou faisant appel à des sous-fonctions récursives) de signature `lirebis : int -> arbre -> int` telle que `lirebis i a` retourne $N_i(a)$ si i est entre 1 et $|a|$ en supposant que l'arbre binaire a respecte l'ordre militaire. Cette fonction devra seulement parcourir (une seule fois) la branche de l'arbre a où se situe le nœud de numéro i .

Question 18 : Écrire finalement en Caml une fonction récursive (ou faisant appel à des sous-fonctions récursives) vérifiant si un arbre binaire est un tas binaire respectant l'ordre militaire, le tout en un seul parcours plutôt que deux.

1. Pour rappel : tous les niveaux sont complets sauf éventuellement le dernier pour lequel les nœuds sont alignés à gauche.