DS 3

Option informatique, deuxième année

Julien Reichert

Partie 1 : Programmation

Exercice 1 : Écrire en Caml la fonction `somme_consecutifs` qui prend en entrée une liste et retourne la liste composée de toutes les sommes d'éléments consécutifs identiques de la liste. La taille de la liste retournée sera donc inférieure ou égale à la taille de la liste en entrée. Écrire aussi une fonction similaire travaillant sur des tableaux. On ne convertira à aucun moment les objets.

Exercice 2 : Écrire en Caml la fonction `somme_successifs` qui prend en entrée une liste et retourne la liste composée de toutes les valeurs obtenues en sommant les sept éléments de la liste en partant de l'indice correspondant inclus et en se déplaçant vers la gauche. S'il y a moins de sept éléments en tout, on s'arrête au début de la liste sans déclencher d'erreur. Écrire aussi une fonction similaire travaillant sur des tableaux. On ne convertira à aucun moment les objets.

Exercice 3 : Écrire en Caml la réciproque des fonctions de l'exercice précédent.

Exercice 4 : Écrire en Caml la fonction `motif_commun` qui prend en entrée deux tableaux `t` et `tt` et un entier naturel `k` et qui retourne un tableau de taille `k` apparaissant à la fois dans `t` et dans `tt` en tant que sous-tableau. Si plusieurs tels tableaux existent, on retournera n'importe lequel.

Exercice 5 : Écrire en Caml la fonction `tous_lancers` qui prend en entrée un entier naturel `n` et retourne tous les tableaux possibles de `n` entiers entre `0` et `5` inclus. Rappel : il y en a `6` à la puissance `n`. L'ordre n'est pas imposé.

Partie 2 : Logique

Les deux exercices ci-après sont inspirés de l'excellent *Livre qui rend fou* de Raymond Smullyvan.

Exercice 6 : Princesse ou tigre

Le principe est simple : un certain nombre de salles fermées (sans possibilité de connaître leur contenu) peuvent abriter une princesse ou un tigre. Sur la porte de chaque salle est apposé un écriteau, et on garantit que l'écriteau énonce la vérité si la salle abrite une princesse et qu'il ment si elle abrite un tigre (si la salle est vide, l'affirmation peut indifféremment être vraie ou fausse).

Déterminer pour chacune des configurations ci-dessous le contenu des salles. Il est recommandé d'utiliser formules et notations de la logique propositionnelle.

— Écriteau de gauche : « Une au moins des cellules contient une princesse. » ; écriteau de droite : « Il y a un tigre dans l'autre cellule. » (Il n'y a pas de cellule vide ici.)
— Écriteau de gauche : « Il y a un tigre dans cette cellule ou il y a une princesse dans l'autre » ; écriteau de droite : « Il y a une princesse dans l'autre cellule. » (Ici non plus.)
— Écriteau de gauche : « La cellule de droite est vide. » ; écriteau du milieu : « Le tigre est dans la cellule de gauche. » ; écriteau de droite : « Cette cellule est vide. ». (Il y a une cellule vide, un tigre et une princesse ici.)

Exercice 7 : Vampires et fous

Considérons une population en Transylvanie où l'on peut trouver des vampires et des humains, dans les deux cas fous ou sains d'esprit. Les vampires mentent toujours et les humains sont toujours sincères. En outre, les fous se trompent toujours et les personnes saines d'esprit ne se trompent jamais. Ainsi, les humains sains d'esprit et les vampires fous diront que la Terre est ronde (ces derniers car ils pensent qu'elle est plate), par exemple.

Identifier parmi les deux personnages qui est le vampire et qui est l'humain dans chacune des situations suivantes :

— A : « Nous sommes fous. » ; B : « C'est faux ! »
— A et B : « Je suis humain. » ; A : « B est sain d'esprit. »
— A et B : « Il y a au moins un fou parmi nous. » ; A : « Je suis humain. »
— A : « B est un vampire. » ; B : « A est fou. » (On précise ici qu'exactement un des deux personnages est fou.)

Si ces deux derniers exercices ont été agréables, la bibliographie de Raymond Smullyvan occupera l'esprit durant de nombreuses heures.

Partie 3 : Langages formels

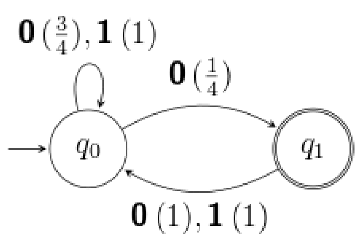
On fixe dans cet exercice un alphabet $\Sigma = \{0, 1\}$.

Un *automate probabiliste* sur l'alphabet Σ est un quadruplet $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$, où :
— Q est un ensemble fini non vide d'*états* ;
— $q_0 \in Q$ est appelé *état initial* ;
— $F \subseteq Q$ est un ensemble dont les éléments sont appelés *états finals* ;
— Pr: $Q \times \Sigma \times Q \to [0; 1]$ est une application appelée *fonction probabiliste de transition*, pour laquelle on suppose que pour tout $q \in Q$, pour tout $a \in \Sigma$, on a $\sum_{q' \in Q} \mathrm{Pr}(q, a, q') = 1$. On note $\mathrm{Pr}(q \xrightarrow{a} q')$ pour $\mathrm{Pr}(q, a, q')$.

Une *transition* est un triplet $(q, a, q') \in Q \times \Sigma \times Q$, noté $(q \xrightarrow{a} q')$, tel que $\mathrm{Pr}(q \xrightarrow{a} q') > 0$.

On représente un automate probabiliste de manière graphique, de façon similaire à la représentation des automates non-déterministes classiques : les états sont représentés par des cercles, l'état initial par une flèche arrivant sur le cercle correspondant, les états finals par des cercles doubles (ou une flèche sortant des cercles correspondants). La fonction probabiliste de transition est représentée par une flèche entre états, dont l'étiquette est cette fois $a(p)$ pour a la lettre et p la probabilité si elle est non nulle.

Ainsi, l'automate $\mathcal{A}_0 = (\{q_0, q_1\}, q_0, \{q_1\}, \mathrm{Pr})$ représenté ci-dessous :



a pour fonction probabiliste de transition Pr la fonction suivante (seules les valeurs non nulles sont mentionnées) :

q	α	q'	$\mathrm{Pr}(q \xrightarrow{\alpha} q')$
q_0	**0**	q_0	3/4
q_0	**0**	q_1	1/4
q_0	**1**	q_0	1
q_1	**0**	q_0	1
q_1	**1**	q_0	1

Étant donné un automate probabiliste $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ sur Σ, un *chemin* ρ est une suite finie de transitions $q_{i_1} \xrightarrow{a_1} q_{i_2} \xrightarrow{a_2} \dots \xrightarrow{a_n} q_{i_{n+1}}$; on dit que ρ a pour *étiquette* le mot $a_1 a_2 \dots a_n \in \Sigma^*$, pour *état de départ* l'état $q_{i_1} \in Q$ et pour *état d'arrivée* l'état $q_{i_{n+1}} \in Q$. La *probabilité* de ρ, notée $\mathrm{Pr}(\rho)$, est définie comme $\Pi_{k=1}^{n} \mathrm{Pr}(q_{i_k} \xrightarrow{a_k} q_{i_{k+1}})$. Un état peut être vu comme un chemin de longueur nulle, la probabilité d'un chemin de longueur nulle est égale à 1. Un *chemin pour le mot* $u \in \Sigma^*$ est un chemin dont l'étiquette est u et l'état de départ est q_0. Ce chemin est *acceptant* si l'état d'arrivée est un état de F, *non-acceptant* sinon. La *probabilité d'un mot* $u \in \Sigma^*$, notée $\mathrm{Pr}(u)$, est par définition la somme des probabilités de tous les chemins acceptants pour le mot u.

Question 1 : Calculer les probabilités des mots ε (le mot vide), 0 et 010 pour l'automate $\mathcal{A}_0$.

Question 2 : Montrer, pour tout automate probabiliste $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ et tout mot $u \in \Sigma^*$, l'égalité suivante en utilisant une récurrence sur la longueur du mot u : $\mathrm{Pr}(u) = 1 - \Sigma_{\rho \text{ chemin non acceptant sur } u} \mathrm{Pr}(\rho)$.

Question 3 : On revient à l'automate probabiliste $\mathcal{A}_0$. Quels sont les mots u dont la probabilité $\mathrm{Pr}(u)$ pour $\mathcal{A}_0$ est égale à 0 ? Quels sont ceux dont la probabilité est égale à 1 ?

Question 4 : Proposer (sans justification) une expression rationnelle pour le langage des mots u dont la probabilité $\mathrm{Pr}(u)$ pour $\mathcal{A}_0$ est non nulle.

Question 5 : Montrer que pour tout automate probabiliste $\mathcal{A}$, il existe un automate non nécessairement déterministe $\mathcal{A}'$ qui accepte exactement les mots dont la probabilité pour $\mathcal{A}$ est non nulle.

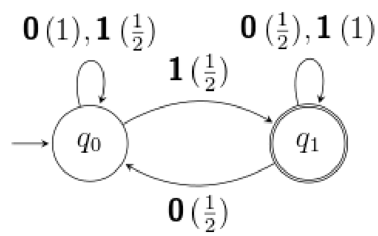
Question 6 : Appliquer la construction de la question précédente à l'automate $\mathcal{A}_0$ pour obtenir un automate non-déterministe qui accepte exactement les mots dont la probabilité pour $\mathcal{A}_\prime$ est non nulle. Déterminiser cet automate.

Soit $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ un automate probabiliste sur Σ. Pour un réel $\eta \in [0; 1[$, le η-langage reconnu par $\mathcal{A}$, noté $L_\eta(\mathcal{A})$, est défini par : $L_\eta(\mathcal{A}) = \{u \in \Sigma^* \mid \mathrm{Pr}(u) > \eta\}$. On dit qu'un langage $L \subseteq \Sigma^*$ est *stochastique* s'il existe un automate probabiliste et un réel $\eta \in [0; 1[$ tel que L soit le η-langage reconnu par l'automate.

Question 7 : Démontrer que tout langage rationnel est stochastique.

Étant donné un mot $\alpha_1 \alpha_2 \dots \alpha_n$ sur l'alphabet $\{0, 1\}$, on dit que l'expression $\overline{0, \alpha_1 \alpha_2 \dots \alpha_n}^2$ est une écriture (finie) en base deux du nombre réel $\Sigma_{i=1}^{n} 2^{-i} \alpha_i$.

On considère maintenant l'automate $\mathcal{A}_1 = (\{q_0, q_1\}, q_0, \{q_1\}, \mathrm{Pr})$ ci-dessous :



Question 8 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_0 \xrightarrow{1} q_1)$ et en donner une écriture finie en base deux.

Question 9 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(10)$ et en donner une écriture finie en base deux.

Question 10 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(1101)$ et en donner une écriture finie en base deux.

Question 11 : Soit $u \in \{0, 1\}^*$ un mot arbitraire. Montrer que $\mathrm{Pr}(u)$ pour $\mathcal{A}_1$ admet une écriture finie en base deux, et en donner une expression. Prouver que cette écriture est correcte.

Question 12 : Soit $\eta \in [0;1[$. Prouver l'égalité suivante :

$$L_\eta(\mathcal{A}_1) = \{\alpha_1\alpha_2\ldots\alpha_n \in \{0,1\}^* \mid \overline{0,\alpha_n\alpha_{n-1}\ldots\alpha_1}^2 > \eta\}.$$

Question 13 : En déduire qu'il existe des langages stochastiques qui ne sont pas rationnels.

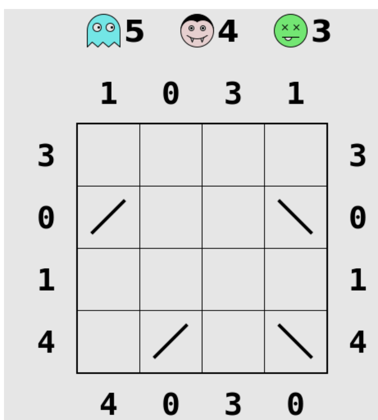
Partie 4 : Énigme

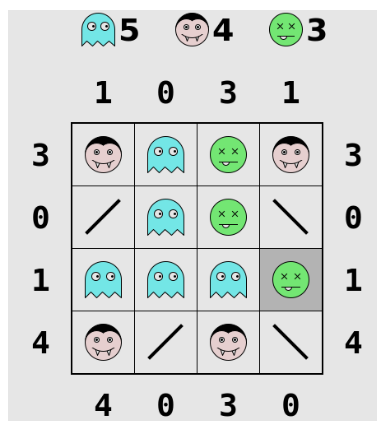
Résoudre les deux instances du jeu « monstres » ci-après.

Le principe du jeu est de remplir toutes les cases vides d'une grille avec des dessins ou symboles de monstres, parmi des vampires (V), des fantômes (G) et des zombies (Z). Les cases occupées de base sont des miroirs orientés en diagonale.

Il y a deux règles de placement :
- Le nombre d'exemplaires total de chaque monstre est imposé et signalé en haut de la grille.
- Chaque ligne et chaque colonne indique le nombre de monstres successivement visibles sur le trajet d'un rayon de lumière qui se réfléchirait sur les miroirs jusqu'à sortir de la grille. Il faut alors savoir qu'on voit un vampire directement, mais jamais son reflet, qu'on voit le reflet d'un fantôme, mais qu'on ne le voit pas directement, et qu'on voit un zombie dans tous les cas.

Un exemple et sa solution permettent de mieux fixer les idées.





Et voici les grilles de l'exercice.