Option Informatique en Spé MP et MP*

Devoir surveillé du mercredi 21 mars 2007

Autour des codes de Gray

Résumé

Nous nous intéressons aux codes de Gray réfléchis cycliques. Un tel code est une énumération des entiers de 0 à $2^n - 1$, satisfaisant la propriété suivante : les écritures en base 2 du k-ième et du $(k+1)$-ième entier ne diffèrent que par un seul bit ; de plus, le premier et le dernier entier vérifient cette même propriété. Nous étudions quelques propriétés combinatoires de ces codes ; nous concevons un circuit logique réalisant la conversion dans le cas $n = 3$; quelques applications sont proposées ; enfin, nous considérons une extension en deux dimensions de ces codes, extension intéressante pour la fabrication de certains types de puces à ADN.

Veuillez rédiger chaque partie sur une copie séparée.

Table des matières

1 Définitions et notations

▶ Notons Σ l'alphabet $\{\mathbf{0},\mathbf{1}\}$.

▶ Un *code de Gray* d'ordre n est une façon d'énumérer les entiers de 0 à $2^n - 1$ possdant la propriété suivante : lorsque l'on passe de l'écriture binaire du k-ième à celle du $(k+1)$-ième nombre, il n'y a qu'un bit qui change.

▶ Voyons ceci sur un exemple simple : si nous énumérons les écritures binaires des nombres de 0 à 3, nous obtenons **00**, **01**, **10** et **11**. Nous constatons que les deux bits de l'écriture **01** de 1 sont différents de leurs homologues dans l'écriture **10** de 2.

▶ En revanche, si nous énumérons les écritures binaires des nombres 0, 1, 3, 2, nous obtenons successivement **00**, **01**, **11** et **10**. Remarquons que le passage de chaque écriture à la suivante ne requiert qu'un changement de bit, de même que le passage de la dernière écriture (**10**) à la première (**00**). Ceci suggère une écriture *circulaire* et non plus linéaire.

▶ Le dispositif suivant illustre cette idée :

Un groupe de quatre capteurs optiques est placé au-dessus de cette roue. Lorsque celle-ci tourne, chaque capteur voit passer l'une des pistes où alternent des zone noires et des zones blanches. La disposition choisie pour celles-ci fait qu'il n'y a jamais deux changements simultanés, ce qui assure des changements d'état francs.

2 Quelques résultats

▶ Soient x une lettre et $\ell = (m_1, m2, \ldots, m_k)$ une liste de mots ; le miroir de ℓ est la liste $(m_k, m_{k-1}, \ldots, m_1)$; nous la noterons ℓ^R. Enfin, $x \cdot \ell$ est la liste $(xm_1, xm_2, \ldots, xm_k)$.

▶ Supposons que ℓ soit un code de Gray d'ordre n ; c'est donc une liste de 2^n mots sur l'alphabet Σ, tous de même longueur n. Par convention, le code de Gray d'ordre 0 est la liste $\Gamma_0 = (\varepsilon)$. Clairement, $\Gamma_1 = (\mathbf{0}, \mathbf{1})$.

Question 1 Montrez que la liste obtenue en concaténant dans cet ordre les listes $(\mathbf{0} \cdot \ell)$ et $(\mathbf{1} \cdot \ell^R)$ est un code de Gray d'ordre $n + 1$.

Question 2 Explicitez Γ_3 ; aucune preuve n'est demandée.

▶ Les codes de Gray obtenus avec la construction qui vient d'être décrite sont dits *réfléchis*.

▶ Nous indexons de 0 à $2^n - 1$ les mots qui constituent la liste Γ_n. Soient k et n deux entiers vérifiant $k < 2^n$. $\gamma(k)$ désigne le mot d'indice k dans la liste Γ_n ; $\mathbf{g}(k)$ désigne l'entier représenté en base 2 par ce mot. Par exemple, si $n = 5$, alors $\gamma(9) = \mathbf{01101}$ et $\mathbf{g}(9) = 13$.

Question 3 Soit $r \in [\![0, 2^n - 1]\!]$; notons $k = 2^n + r$. Prouvez la relation $\mathbf{g}(k) = 2^n + \mathbf{g}(2^n - 1 - r)$.

Question 4 Que pensez-vous de l'affirmation suivante : ≪dans un code de Gray réfléchi, on passe d'un nombre au suivant en changeant un bit : le dernier si le nombre de **1** est pair ; celui à gauche du **1** le plus à droite, sinon≫.

▶ Remarquons que Γ_n est préfixe de Γ_{n+1}, et que le k-ième mot de la liste Γ_n est suffixe du k-ième mot de la liste Γ_{n+1}. Ceci suggère une autre façon de présenter ces codes de Gray.

▶ Soient n et k deux naturels vérifiant $0 \leqslant k < 2^n$. L'écriture en base 2 de k sur n bits est un mot $m = b_{n-1}b_{n-2}\dots b_1 b_0$ sur l'alphabet Σ. Nous lui associons le mot infini $\psi(k) = b_0 b_1 \dots b_{n-2}b_{n-1}\mathbf{0}^\omega$ obtenu en «retournant» m et en le faisant suivre d'une infinité de $\mathbf{0}$. Pour illustrer ceci, prenons par exemple $n = 5$ et $k = 13$: alors $m = \mathbf{01101}$ et $\psi(13) = \mathbf{101100}^\omega = \mathbf{10110}^\omega$, où $\mathbf{0}^\omega$ est le mot infini consitué uniquement de $\mathbf{0}$. La remarque faite plus haut montre que $\psi\big(\mathbf{g}(k)\big)$ ne dépend pas du choix de n, du moment que $2^n > k$. Nous noterons $\overline{\psi}(k) = a_0 a_1 \dots$ ce mot infini. Ainsi, $\overline{\psi(13)} = \mathbf{11010}^\omega$.

Question 5 Prouvez la relation $b_j = a_j \oplus a_{j+1}$, où $\oplus$ est l'addition modulo 2.

Question 6 Expliquez pourquoi la notation $b_j \oplus b_{j+1} \oplus \cdots$ a un sens.

Question 7 Prouvez la relation $a_j = b_j \oplus b_{j+1} \oplus \cdots$

▶ L'opération $\oplus$ peut s'appliquer à deux mots u et v, sous réserve qu'ils soient de même longueur (finie ou infinie) : $(u \oplus v)_j = u_j \oplus v_j$ pour tout j.

Question 8 Justifiez alors la formule $\overline{\psi}(k) = \psi(k) \oplus \psi\big(\lfloor k/2 \rfloor\big)$.

3 Circuit logique de conversion

▶ Nous voulons construire un circuit à trois entrées e_0, e_1 et e_2 et trois sorties s_0, s_1 et s_2 défini comme suit : si nous notons $k = e_0 + 2e_1 + 4e_2$ le nombre présenté en binaire sur les entrées, alors $\mathbf{g}(k) = s_0 + 2s_1 + 4s_2$.

Question 9 Dressez un tableau possédant 8 lignes (numérotées de 0 à 7) et six colonnes ; dans les trois premières colonnes, vous placerez les valeurs de e_0, e_1 et e_2 ; dans les trois dernières, celles de s_0, s_1 et s_2.

Question 10 Donnez l'équation logique exprimant s_0 en fonction de e_0, e_1 et e_2. Donnez de même les équations logiques décrivant s_1 puis s_2.

▶ Les portes logiques *élémentaires* sont la porte ET à deux entrées, la porte OU à deux entrées, et la porte NON (à une entrée) ; elles sont décrites dans la figure 1.

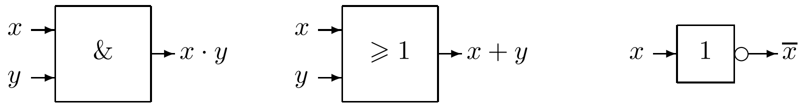

Figure 1: les trois portes logiques élémentaires

Question 11 Dessinez un circuit combinatoire à trois entrées (e_0, e_1 et e_2) et une sortie s_0 ; ce circuit calcule la valeur de $\mathbf{g}(k)$ mod 2. Vous n'utiliserez que des portes logiques élémentaires ; les entrées du circuit seront placées du côté gauche, la sortie du côté droit.

Question 12 Proposez une réalisation très simple du convertisseur précédent, utilisant une autre porte logique élémentaire que celles présentées plus haut.

4 Quelques applications des codes de Gray

4.1 Circuit hamiltonien sur un hypercube

▶ Soient u et v deux mots de même longueur n. La *distance de Hamming* de ces deux mots est le nombre d'indices $i \in [\![1, n]\!]$ tels que $u_i \neq v_i$. Elle est notée $d(u, v)$. Par exemple, $d(\mathbf{0010110}, \mathbf{0110011}) = 3$. Les deux relations $d(u, u) = 0$ et $d(u, v) = d(v, u)$ sont évidentes.

Question 13 • Prouvez que la distance de Hamming vérifie l'inégalité triangulaire : $d(u, w) \leqslant d(u, v) + d(v, w)$ et ce quels que soient les mots u, v, w de même longueur.

Question 14 • Rédigez en Caml une fonction :

```
distance : string -> string -> int
```

spécifiée comme suit : `distance u v` calcule $d(u, v)$. Vous lèverez une exception si $|u| \neq |v|$. Consultez les indications données au début de la partie 6.

▶ Soit $G = (S, A)$ un graphe non orienté. Un *circuit hamiltonien* sur G est un chemin qui part d'un sommet s, passe une fois et une seule par tous les autres sommets, et se termine en s.

▶ Soit $n \geqslant 1$. L'*hypercube* d'ordre n est un graphe non orienté, dont les sommets sont les mots de longueur n sur l'alphabet $\{\mathbf{0},\mathbf{1}\}$, et dont les arêtes sont les paires de mots (u, v) vérifiant $d(u, v) = 1$.

Question 15 Expliquez comment obtenir un circuit hamiltonien sur un hypercube, en utilisant un code de Gray réfléchi.

4.2 Tours de Hanoï et codes de Gray

▶ Dans le casse-tête des tours de Hanoï, n disques sont rangés par tailles décroissantes sur l'axe 1. Il s'agit d'amener cette pile sur l'axe 3, en ne déplaçant qu'un disque à la fois, et en ne formant que des piles décroissantes. Il est connu que la solution optimale requiert $2^n - 1$ déplacements.

Question 16 Numérotons les disques par taille croissante, de 1 à n. Montrez qu'un code de Gray réfléchi d'ordre n permet de déterminer, pour chaque valeur de $k \in [\![1, 2^n - 1]\!]$, le numéro du disque à déplacer à la k-ième étape.

4.3 La serrure du professeur Macheprot

▶ Le professeur Macheprot a muni la porte de son laboratoire d'une curieuse serrure de son invention. La chose se compose d'un clavier à cinq touches $\boxed{\mathbf{A}}$, $\boxed{\mathbf{B}}$, $\boxed{\mathbf{C}}$, $\boxed{\mathbf{D}}$ et $\boxed{\mathbf{X}}$; et de quatre tiges a, b, c et d, qui peuvent chacune être dans deux positions (haute ou basse). Appuyer sur une touche autre que $\boxed{\mathbf{X}}$ a pour effet d'inverser la position de la tige associée. Appuyer sur la touche $\boxed{\mathbf{X}}$ a pour effet d'ouvrir la porte si les quatre tiges sont dans la même position. Enfin, un dispositif de sécurité bloque la serrure pour une durée d'une heure, après huit essais infructueux.

Question 17 Proposez (preuve à l'appui) un mot (fini) w sur l'alphabet $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{X}\}$ contenant au plus 8 occurrences de la lettre $\mathbf{X}$, et permettant d'ouvrir la porte. Plus précisément : il existe un préfixe de w qui se termine par la lettre $\mathbf{X}$, et dont la frappe ouvre la porte. Indication : utilisez un code de Gray !

4.4 Les gants de boxe du barman aveugle

▶ Padraig O'Shugrue est un ancien boxeur qui a perdu la vue au cours de son dernier combat. Aujourd'hui, il est le barman d'un lieu branché de Manhattan. Parfois, il propose à un client un jeu étrange qu'il a mis au point : sur un plateau tournant, il dispose quatre verres, formant les sommets d'un carré. Le client peut alors en retourner certains (ou aucun). Padraig enfile les gants du dernier combat (qui sont accrochés au mur du bar), ce qui l'empêche de savoir, au toucher, si un verre est ≪pied en bas≫ ou ≪pied en haut≫. Il peut alors retourner un, ou deux, ou trois verres ; si les quatre verres se trouvent tous dans la même position, le barman a gagné ; sinon, le client peut faire tourner le plateau d'un ou deux quarts de tour, dans un sens arbitraire ; puis c'est au tour de Padraig de jouer. Si ce dernier n'a pas trouvé au bout de huit essais, le client a gagné. Curieusement, aucun client n'a jamais gagné.

Question 18 Expliquez la dernière phrase du paragraphe précédent, en donnant la méthode du barman. Indication : revoyez la partie précédente !

Question 19 Montrez que le jeu peut être modélisé par un automate fini à quatre états que vous décrirez.

5 Puces à ADN et codes de Gray bidimensionnels

▶ Une *puce à ADN* d'ordre n se compose d'un substrat carré (d'environ 1 cm de côté), décomposé en 4^n zones carrées de même taille ; danas chacune de ces zones, on implante un grand nombre de molécules d'ADN monobrin identiques, de longueur n. Chacune de ces molécules est un mot sur l'alphabet $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$. Deux molécules implantées sur des zones différentes doivent différer par au moins une lettre. La figure 2 présente deux dispositions différentes pour une puce à ADN d'ordre 2.

AA	AC	CA	CC
AG	AT	CG	CT
GA	GC	TA	TC
GG	GT	TG	TT

AA	AC	CC	CA
AG	AT	CT	CG
CG	CT	TT	TG
CA	CC	TC	TA

Figure 2: implantation fractale (à gauche) et avec un code de Gray (à droite)

▶ La fabrication d'une telle puce requiert une succession de $4n$ étapes, que l'on peut représenter par le mot $m = (\mathsf{A\,C\,G\,T})^n$. Initialement, on dépose sur la puce des molécules dotées d'une base qui va adhérer au substrat, et d'un groupe protecteur photolabile. Lors de la k-ième étape, on pose un *masque* sur la puce et on l'expose aux ultra-violets : ceci élimine les groupes protecteur des zones exposées ; on enlève le masque et on déverse sur la puce une solution contenant la base m_k, à laquelle est attaché un groupe protecteur. La figure 3 donne une représentation (très approximative).

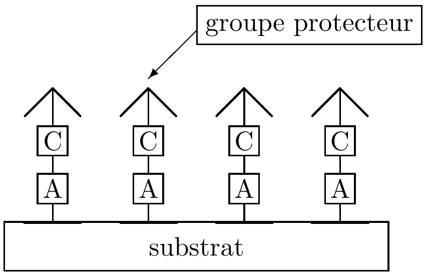

Figure 3: puce à ADN

▶ La figure 4 donne deux exemples de masques ; ce sont ceux associés à l'étape 5 (dépôt du deuxième A) pour chacune des deux dispositions présentées à la figure 2. Remarque : curieusement, les zones transparentes du masque sont représentées en noir !

Figure 4: masques pour l'étape $\mathsf{A}-2$

▶ Avec l'implantation fractale, les puces sont définies par les deux dessins ci-dessous ; la notation $\mathsf{X}\,\mathsf{F}_n$ désigne une puce d'ordre n, dans laquelle on a ajouté une lettre X en tête de chaque mot.

$$\mathsf{F}_1 = \begin{array}{|c|c|}\hline \mathsf{A} & \mathsf{C} \\ \hline \mathsf{G} & \mathsf{T} \\ \hline \end{array} \qquad \mathsf{F}_{n+1} = \begin{array}{|c|c|}\hline \mathsf{A}\,\mathsf{F}_n & \mathsf{C}\,\mathsf{F}_n \\ \hline \mathsf{G}\,\mathsf{F}_n & \mathsf{T}\,\mathsf{F}_n \\ \hline \end{array}$$

▶ À la frontière entre les parties cachées et les parties exposées, se produisent des phénomènes nuisibles (diffraction). On veut donc minimiser la longueur totale des bords des masques. Remarquons que les deux masques de la figure 4 ont la même longueur de bord, égale à 12 (on ne compte pas ce qui est «au bord» de la puce).

▶ Pour ce faire, nous disposerons les 4^n mots selon un *code de Gray bidimensionnel* noté G_n et défini par les équations suivantes :

$$\mathsf{G}_1 = \begin{array}{|c|c|}\hline \mathsf{A} & \mathsf{C} \\ \hline \mathsf{G} & \mathsf{T} \\ \hline \end{array} \qquad \mathsf{G}_{n+1} = \begin{array}{|c|c|}\hline \mathsf{A}\,\mathsf{G}_n & \mathsf{C}\,\overrightarrow{\mathsf{G}_n} \\ \hline \mathsf{G}\,\mathsf{G}_n\downarrow & \mathsf{T}\,\overrightarrow{\mathsf{G}_n\downarrow} \\ \hline \end{array}$$

La notation $\overrightarrow{\mathsf{G}_n}$ (respectivement : $\mathsf{G}_n\downarrow$) signifie que G_n a subi une symétrie autour de son axe vertical (respectivement : horizontal). La signification de $\overrightarrow{\mathsf{G}_n\downarrow}$ est claire !

Question 20 Calculez la longueur de bord totale des huit masques requis pour la fabrication d'une puce d'ordre 2, avec chacune des deux dispositions proposées.

Question 21 Donnez des formules exactes pour la longueur de bord totale des $4n$ masques requis pour la fabrication d'une puce d'ordre n, avec chacune des deux dispositions proposées.

Question 22 Évaluer le gain fourni par l'utilisation du code de Gray bidimensionnel, par rapport à l'ordre fractal.

6 Programmation

► Chaque question de programmation fixe un *objectif* : c'est le nombre maximal de lignes que vous devez écrire. Vous compterez pour une ligne l'en-tête d'une fonction, comme :

```
let toto x y = function
```

Vous compterez chaque motif pour une ligne ; notez bien qu'un motif union, comme :

```
| [] | [true] | [_;false;_] -> ...
```

compte pour une seule ligne.

► L'emploi de références ou de structures de données mutables est interdit.

► L'emploi de boucles `for` ne semble pas nécessaire.

Question 23 Rédigez en Caml une fonction de signature :

```
ajoute : 'a -> 'a list list -> 'a list list
```

spécifiée comme suit : `ajoute x q` ajoute l'élément x en tête de chacune des listes qui composent ℓ. Par exemple, `ajoute 8 [[3;1];[];[0;1;0]]` doit rendre la liste `[[8;3;1];[8];[8;0;1;0]]`. Objectif : deux lignes.

Question 24 Rédigez en Caml une fonction de signature :

```
gray : int -> int list list
```

spécifiée comme suit : `gray n` construit le code de Gray réfléchi d'ordre n, en appliquant la formule établie à la question 3. Par exemple, `gray 2` doit rendre la liste `[[0;0];[0;1];[1;1];[1;0]]`. Vous pourrez utiliser `rev` et `@`. Objectif : quatre lignes.

Question 25 Rédigez en Caml une fonction de signature :

```
puiss2 : int -> int list
```

spécifiée comme suit : `puiss2 k` construit la liste, présentée en ordre décroissant, des puissances de 2 au plus égales à k. Objectif : quatre lignes.

Question 26 Rédigez en Caml une fonction de signature :

```
g : int -> int
```

spécifiée comme suit : `g k` calcule $\mathsf{g}(k)$. Par exemple, `gray 4` doit rendre la valeur 6. Vous utiliserez la formule établie à la question 3, ainsi que la fonction `puiss2` demandée à la question précédente. Objectif : neuf lignes.

Question 27 Rédigez en Caml une fonction de signature :

```
base2 : int -> int list
```

spécifiée comme suit : `base2 k` fournit la représentation de k en base 2. Par exemple, `base2 29` rend la liste `[1;1;1;0;1]`. Objectif : quatre lignes.

Question 28 Rédigez en Caml une fonction de signature :

```
g2 : int -> int
```

spécifiée comme suit : `g2 k` calcule $\mathtt{g}(k)$, en utilisant la relation établie à la question 8 et la fonction `base2` demandée à la question précédente. Objectif : huit lignes, dont quatre pour la fonction réalisant la somme modulo 2 de deux listes.

FIN