Structures discrètes

Licence Mathématiques-Informatique, 2ème année

Examen 22 janvier 2009

Durée de l'examen : 2 heures

Les documents hors livres sont autorisés. Les calculettes sont interdites. Toute réponse devra être argumentée. Les cinq exercices sont totalement indépendants.

Exercice 1 :

1. Le langage $a^*b^*a^*b^*$ contient-il le mot baa ?
2. Sur l'alphabet $\{a, b\}$, donner une expression rationnelle du langage des mots qui contiennent un nombre impair de b.
3. Sur l'alphabet $\{a, b, c\}$, donner une expression rationnelle du langage des mots qui ne contiennent pas le facteur ac.

Correction

1. $baa = a^0b^1a^2b^0$ appartient à $a^*b^*a^*b^*$.
2. $a^*b(a^*ba^*b)^*a^*$.
3. $(b \cup c \cup a^*b)a^*$.

Exercice 2 :

On se place sur l'alphabet $A = \{a, b\}$. Sur les langages de A^*, on considère, en plus des opérations rationnelles, les opérations suivantes. Si L est un langage de A^*, pour $n \in \mathbb{N}$,

$$L^n = \begin{cases} \{\varepsilon\} & \text{si } n = 0, \\ L.L^{n-1} & \text{sinon;} \end{cases} \qquad \text{et } L^{<n} = \bigcup_{0 \leqslant k < n} L^k.$$

Si L et K sont deux langages de A^*, le produit $L.K$ est *ambigu* s'il existe deux mots distincts u et u' de L et deux mots v et v' de K tels que $uv = u'v'$. On définit :

$$L \otimes K = \begin{cases} \emptyset & \text{si } L.K \text{ est ambigu,} \\ L.K & \text{sinon.} \end{cases}$$

Dans ce qui suit, C est un langage de A^* qui est un code. Montrer que, pour tout $n \in \mathbb{N}$,

1. $C^{<n} \cup (C^n \otimes C^*) = C^*$; montrer que $C^{<n} \cap (C^n \otimes C^*) = \emptyset$;
2. $C^{<n} \otimes ((C^n)^*) = C^*$.

Correction

1. Soit un mot w appartenant à $C^{<n} \cup (C^n \otimes C^*)$; si w est dans $C^{<n}$, $w = c_1c_2...c_k$, avec $k < n$ et les mots c_i sont dans C, donc w est dans C^* ; si w est dans $(C^n \otimes C^*)$, $w = c_1c_2...c_nu$, avec les mots c_i dans C et u dans C^*, donc w est dans C^*. Réciproquement, si w est dans C^*, $w = c_1c_2...c_k$ avec les mots c_i dans C ; si $k < n$, alors w est dans $C^{<n}$, sinon, $w = c_1c_2...c_n(c_{n+1}...c_k)$, avec $c_{n+1}...c_k$ dans C^*, donc w est dans $C^n.C^*$. Par ailleurs, comme C est un code, la décomposition de w en mots de C est unique, donc l'union est disjointe ($C^{<n} \cap (C^n \otimes C^*) = \emptyset$) et le produit est non ambigu $C^n.C^* = C^n \otimes C^*$.

2. De même, si w est dans C^*, w s'écrit de manière unique $w = c_1c_2...c_k$, avec les mots c_i dans C ; si on regroupe ces mots par paquets de n à partir de la fin, on obtient que w est dans $C^{<n} \otimes ((C^n)^*) = C^*$. La réciproque est évidente, le produit est non ambigu car C est un code.

Exercice 3 :

Soit $P(X) = 1 + X^3 + X^4$ un polynôme à coefficients dans $\mathbb{Z}/2\mathbb{Z}$.

1. Montrer que ce polynôme est primitif.
2. On l'utilise pour faire un codage de Hamming. Combien y a-t-il de bits d'information et de bits de correction dans un mot du code ?
3. On désire envoyer le message 11001011010 ; quel est le code associé ?
4. Décoder le bloc 110101100010100 en faisant l'hypothèse qu'au plus une erreur est survenue.

Correction

1.

$$\begin{aligned} X^0 &= 1 \mod P & X^1 &= X \mod P \\ X^2 &= X^2 \mod P & X^3 &= X^3 \mod P \\ X^4 &= 1 + X^3 \mod P & X^5 &= 1 + X + X^3 \mod P \\ X^6 &= 1 + X + X^2 + X^3 \mod P & X^7 &= 1 + X + X^2 \mod P \\ X^8 &= X + X^2 + X^3 \mod P & X^9 &= 1 + X^2 \mod P \\ X^{10} &= X + X^3 \mod P & X^{11} &= 1 + X^2 + X^3 \mod P \\ X^{12} &= 1 + X \mod P & X^{13} &= X + X^2 \mod P \\ X^{14} &= X^2 + X^3 \mod P & X^{15} &= 1 \mod P \end{aligned}$$

2. Les puissances de X prennent toutes les valeurs possibles dans $\mathbb{Z}/2\mathbb{Z}[X]$ modulo P, P est donc primitif.
3. Il y a 15 polynômes possibles, donc un mot du code a une longueur 15 ; P est de degré 4, donc parmi ces 15 bits, il y a 4 bits de correction et 11 d'information.
4. Le calcul des polynômes fait en 1. peut se représenter par le tableau suivant :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X^0	1	0	0	0	1	1	1	1	0	1	0	1	1	0	0
X^1	0	1	0	0	0	1	1	1	1	0	1	0	1	1	0
X^2	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1
X^3	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1

Pour calculer le code associé à $w = 11001011010$, on place ce mot sous les colonnes de droite :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
w					1	1	0	0	1	0	1	1	0	1	0

On fait la somme des colonnes qui correspondent à un 1 :

	4	5	8	10	11	13	*Somme*
X^0	1	1	0	0	1	0	1
X^1	0	1	1	1	0	1	0
X^2	0	0	1	0	1	1	1
X^3	1	1	1	1	1	0	1

La somme nous indique les bits de correction ; le mot du code est donc 101111001011010.

5. Pour décoder $c = 110101100010100$, on le place sous le tableau :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	1	1	0	1	0	1	1	0	0	0	1	0	1	0	0

On fait la somme des colonnes qui correspondent à un 1 :

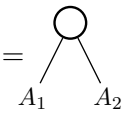
	0	1	3	5	6	10	12	*Somme*
X^0	1	0	0	1	1	0	1	1
X^1	0	1	0	1	1	1	1	1
X^2	0	0	0	0	1	0	0	1
X^3	0	0	1	1	1	1	0	0

La somme nous indique la colonne correspondant au bit faux : le bit 7. Le code corrigé est donc 110101110010100, le mot envoyé est 01110010100.

Exercice 4 :

On considère l'ensemble des arbres binaires complets, noté $\mathcal{B}$, défini inductivement selon le schéma suivant :

$$\begin{cases} B = \{\bigcirc\} \\ f(A_1, A_2) = \quad \pour\ A_1, A_2 \in \mathcal{B} \end{cases}$$

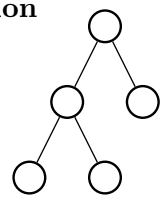


On définit l'application Φ inductivement sur $\mathcal{B}$ par :

$$\begin{cases} \Phi(\bigcirc) = \bigcirc \\ \Phi(f(A_1, A_2)) = \begin{cases} f(B_1, f(B_2, \Phi(A_2))) & \text{si } \Phi(A_1) = f(B_1, B_2) \\ f(\bigcirc, \Phi(A_2)) & \text{si } \Phi(A_1) = \bigcirc \end{cases} \end{cases}$$

1. Représenter l'arbre $T = f(f(\bigcirc, \bigcirc), \bigcirc)$.
2. Calculer $\Phi(T)$.
3. Montrer inductivement que l'image de n'importe quel arbre de $\mathcal{B}$ par Φ a le même nombre de feuilles et le même nombre de nœuds internes.
4. Calculer $\Phi(\Phi(T))$. L'application Φ est-elle une bijection ?

Correction

1. 

2. $\Phi(f(\bigcirc,\bigcirc)) = f(\bigcirc,\bigcirc)$, donc $\Phi(f(f(\bigcirc,\bigcirc),\bigcirc)) = f(\bigcirc, f(\bigcirc,\bigcirc))$.

3. On prouve par induction que pour tout arbre T, Φ préserve le nombre de feuilles et de nœuds internes. On note n_T le nombre de nœuds internes dans un arbre et g_T le nombre de feuilles.
   Base : Si T est une feuille, $\Phi(T)$ est une feuille, donc $n_T = n_{\Phi(T)} = 0$ et $g_T = g_{\Phi(T)} = 1$.
   Induction : Si $T = f(\bigcirc, A)$, $\Phi(T) = f(\bigcirc, \Phi(A))$; par hypothèse d'induction $n_A = n_{\Phi(A)}$ et $g_A = g_{\Phi(A)}$, donc $n_T = 1 + n_A = 1 + n_{\Phi(A)} = n_{\Phi(T)}$ et $g_T = 1 + g_A = 1 + g_{\Phi(A)} = g_{\Phi(T)}$. Si $T = f(f(A_1, A_2), C)$, soit $f(B_1, b_2) = \Phi(f(A_1, A_2))$. On a $\Phi(T) = f(B_1, f(B_2, \Phi(C)))$; par hypothèse d'induction, on obtient, $n_{\Phi(T)} = n_{B_1} + n_{B_2} + n_C + 2 = (1 + n_{B_1} + n_{B_2}) + n_C + 1 = (1 + n_{A_1} + n_{A_2}) + n_C + 1 = n_T$ et $g_{\Phi(T)} = g_{B_1} + g_{B_2} + g_C = g_{A_1} + g_{A_2} + g_C + 1 = g_T$.

4. $\Phi(\Phi(T)) = \Phi(T)$, donc deux arbres différents (T et $\Phi(T)$) ont la même image ; Φ n'est donc pas injective, donc pas bijective.

Exercice 5 :

On se place sur l'alphabet $A = \{0, 1\}$. On considère l'application $f : A \longrightarrow A^*$ définie par :

$$\begin{aligned} f : 0 &\longmapsto 01 \\ 1 &\longmapsto 0 \qquad . \end{aligned}$$

On étend cette application aux mots : si $w = w_1 w_1 ... w_n$ est un mot de longueur n, alors $f(w) = f(w_1)f(w_2)...f(w_n)$. On considère la suite de mots définie par :

$$\begin{cases} u_0 = 0, \\ u_{k+1} = f(u_k), \quad k \in \mathbb{N}. \end{cases}$$

1. Calculer u_1, u_2 et u_3.
2. a) Montrer que pour tout $k \geqslant 0$, $u_{k+2} = u_{k+1}u_k$.
   b) On note ℓ_k la longueur du mot u_k. Montrer que $\ell_{k+2} = \ell_{k+1} + \ell_k$.
3. Montrer que quel que soit k, u_k ne contient aucun facteur 11, ni aucun facteur 000.
4. Montrer que, pour tout $k \geqslant 0$, le nombre de 0 dans u_{k+1} est égal à ℓ_k. En déduire que le nombre de 1 dans u_{k+2} est aussi égal à ℓ_k.
5. a) Montrer que, pour tout $k \geqslant 0$, $\ell_{k+1}^2 - \ell_{k+1}\ell_k - \ell_k^2 = (-1)^k$.
   b) Soit $p : x \mapsto x^2 - x - 1$. Montrer que $p(\ell_{k+1}/\ell_k) = \dfrac{(-1)^k}{\ell_k^2}$.
   c) Montrer que p restreint à $[1; 2]$ réalise une bijection continue de cet intervalle sur $[-1; 1]$. En déduire que la suite $(\ell_{k+1}/\ell_k)_{k\geqslant 0}$ converge et calculer la valeur de sa limite.
   d) Pour tout mot w de A^*, $|w|_0$ est le nombre de 0 dans w et $|w|_1$ le nombre de 1. Déduire des questions précédentes la limite de $\dfrac{|u_k|_0}{|u_k|_1}$, lorsque k tend vers l'infini.

Correction

1. $u_1 = 01$, $u_2 = 010$, $u_3 = 01001$.
2. a) On montre par récurrence que pour tout $k \geqslant 0$, $u_{k+2} = u_{k+1}u_k$. C'est vrai pour $k = 0 : u_2 = 01.0 = u_1u_0$. Si c'est vrai pour $k-1$, alors $u_{k+2} = f(u_{k+1}) = f(u_ku_{k-1}) = f(u_k)f(u_{k-1}) = u_{k+1}u_k$, c'est donc vrai pour k. Par récurrence, la propriété est donc vraie pour tout k.
   b) $\ell_{k+2} = |u_{k+2}| = |u_{k+1}u_k| = |u_{k+1}| + |u_k| = \ell_{k+1} + \ell_k$.
3. Tout mot u_k commence par un 0 car u_0 et u_1 commencent par un 0 et par récurrence, si u_k commence par 0, $u_{k+1} = u_ku_{k-1}$ commence par un 0. On montre par récurrence que u_k ne contient pas de facteur 11. C'est vrai pour u_0 et u_1. Par récurrence, s u_k et u_{k+1} ne contiennent pas de facteur 11, alors $u_{k+2} = u_{k+1}u_k$ non plus, puisque u_k commence par un 0. les mots u_k ne contiennent pas non plus de facteur 000, car un 0 qui n'est pas suivi d'un 1 est forcément l'image de 1 par f, donc un facteur 000 ne peut être obtenu qu'à partir d'un facteur 11.
4. L'image de toute lettre contient exactement un 1. Donc $f(u_k)$ contient exactement $|u_k|$ 0, en d'autres termes, le nombre de 0 dans u_{k+1} est ℓ_k. L'image de 0 contient exactement un 1, alors que celle de 1 n'en contient pas. Le nombre de 1 dans $f(u_k)$ est donc égal au nombre de 0 dans u_k, en d'autres termes, le nombre de 1 dans $u_{k+2} = f(u_{k+1})$ est égal au nombre de 0 dans u_{k+1} soit ℓ_k.
5. a) On montre l'égalité par récurrence sur k. Pour $k = 0$, $\ell_0 = 1$ et $\ell_1 = 2$, donc l'égalité est vrai. Si elle est vraie pour un entier k, sachant que $\ell_k = \ell_{k+2} - \ell_{k+1}$, on a

$$\ell_{k+1}^2 - \ell_{k+1}\ell_k - \ell_k^2 = (-1)^k$$

$$\ell_{k+1}^2 - \ell_{k+1}(\ell_{k+2} - \ell_{k+1}) - (\ell_{k+2} - \ell_{k+1})^2 = (-1)^k$$

$$2\ell_{k+1}^2 - \ell_{k+1}\ell_{k+2} - (\ell_{k+2}^2 - 2\ell_{k+1}(\ell_{k+2} + \ell_{k+1}^2) = (-1)^k$$

$$\ell_{k+1}^2 + \ell_{k+1}\ell_{k+2} - \ell_{k+2}^2 = (-1)^k$$

$$\ell_{k+2}^2 - \ell_{k+2}\ell_{k+1} - \ell_{k+2}^2 = -(-1)^k = (-1)^{k+1}$$

   b)En divisant l'égalité ci-dessus par ℓ_k^2, on obtient

$$\frac{\ell_{k+1}^2}{\ell_k^2} - \frac{\ell_{k+1}}{\ell_k} - 1 = \frac{(-1)^k}{\ell_k^2}$$

   Ce qui répond à la question.
   c) $p'(x) = 2x-1$, donc p' est strictement croissante sur $[1;2]$; comme p est un polynôme, donc une fonction continue, p réalise donc une bijection de $[1;2]$ sur $[p(1);p(2)] = [-1;1]$. Sur $[-1;1]$, p^{-1} est donc une fonction continue croissante. ℓ_k tend vers l'infini, donc $\frac{(-1)^k}{\ell_k^2}$ tend vers 0 et $p^{-1}(\frac{(-1)^k}{\ell_k^2}) = \frac{\ell_{k+1}}{\ell_k}$ tend vers $p^{-1}(0)$, c'est-à-dire vers une racine de p. Les racines de p sont $(1+\sqrt{5})/2$ et $(1-\sqrt{5})/2$; seule $(1+\sqrt{5})/2$ appartient à $[1;2]$, c'est donc la limite de $\frac{\ell_{k+1}}{\ell_k}$.
   d) Le nombre de 0 dans u_k est ℓ_{k-1}, le nombre de 1 est ℓ_{k-2}, donc $\frac{|u_k|_0}{|u_k|_1}$ a la même limite que $\frac{\ell_{k-1}}{\ell_{k-2}}$, soit $(1+\sqrt{5})/2$.