Institut Supérieur de Sciences Appliquées et Économiques
Centre du Liban associé au
Conservatoire National des Arts et Métiers

le cnam Liban

Automates, Codes, Graphes (MVA004)

Examen Final 2017-2018 **2h :00**

Centres de : Beyrouth, Tripoli, Bickfaya, Nahr Ibrahim, Baalbek

Documents et Téléphones : STRICTEMENT INTERDITS
En Annexe un formulaire regroupant les principaux algorithmes

Sujet proposé par :
J.SAAB

Exercice 1 (*12 points*) Lors d'un transfert de données, vous recevez les messages suivants codés grâce au code Hamming $(7,4)$. Des erreurs s'y sont insérées. Retrouvez-les et corrigez-les.
- 0101000
- 1110010
- 1100011

Exercice 2 (18 *points*) Soit le code linéaire $C[5,3]$ de matrice génératrice

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

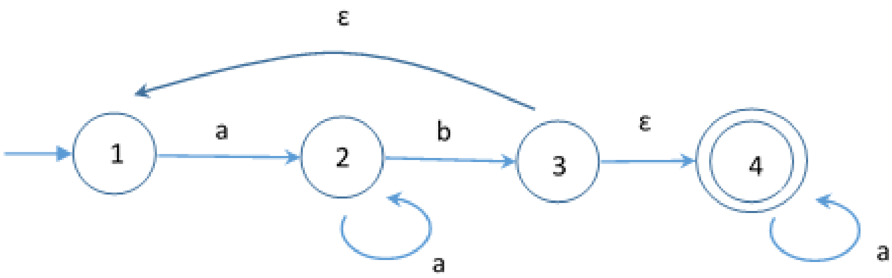
1. Donner la matrice de contrôle du code C
2. Décrire le code orthogonal $C^{\perp}$ de C (c'est à dire donner ses mots de code)
3. Le code (C) est-il un code polynômial

Exercice 3 (25 *points*) Soit (C) un code polynômial de longueur 5 engendré par le polynôme

$$g(x) = x^3 + x^2 + x + 1$$

1. Donner le nombre de bits de données et le nombre de bits de contrôle relatifs à ce code
2. Ce code est-il cyclique ? Justifier !
3. Vérifier que $m = 11110$ est un mot de code
4. Soit $\alpha = 11$. Donner le polynôme de code de α. coder le mot α.
5. Soit $m^{'} = 11011$. Donner $\sigma(m^{'})$ le syndrôme de $m^{'}$ en utilisant l'égalité $\sigma(P_a) = P_{\sigma(a)}$. En déduire que $m^{'}$ n'est pas un mot de code.

Exercice 4 (25 *points*) Sur l'alphabet $\sum = \{a, b\}$, soit l'automate fini non déterministe $(AFN - \varepsilon)$ $\mathcal{A}$, défini par le diagramme :



On note L le langage reconnu par l'automate $\mathcal{A}$.

1. Donner la liste des mots de L de longueur inférieur ou égale à 3.
2. Ecrire la matrice des transitions de $\mathcal{A}$.
3. En déduire la matrice des transitions et le diagramme d'un automate déterministe $\mathcal{B}$ qui reconnaît aussi le langage L.
4. Ecrire le système de départ pour $\mathcal{B}$, et le résoudre. En déduire une expression régulière pour le langage L.
5. En comparant les langages du départ de $\mathcal{B}$, montrer que l'automate $\mathcal{B}$ est minimal.

Exercice 5 (20 *points*) On considère les matrices réelles

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad , \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{et} \quad C = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

1. Calculer AB, BA et A^2 en fonction de A et B.
2. Montrer par récurrence, que

$$3A^n = [2^n - (-1)^n]A + [2^{n-1} + (-1)^n]B, \forall n \geq 1.$$

3. Calculer $|C|$, le déterminant de C en déduire que C est inversible. Trouver C^{-1}, la matrice inverse de C.

Formulaires

Automates :

1. Système de départ :
   - A chaque état q_i on associe un langage de départ (inconnu) D_i
   - Si les symboles $\alpha_j,\ j = 1 \cdots k$, font passer q_i vers les états q_j,alors on établit l'équation

$$D_i = \begin{cases} \sum_{j=1}^{k} \alpha_j D_j & \text{si} \quad q_i \text{ n'est pas acceptant} \\ \sum_{j=1}^{k} \alpha_j D_j + \varepsilon & \text{si} \quad q_i \text{ est acceptant} \end{cases}$$

   - $L(A_d) = D_0$ le langage associé à l'état initial.
   - La résolution du système de départ se base sur la formule :

$$X = uX + v \text{ alors } X = \begin{cases} u^* v & \text{si} \quad \varepsilon \notin u \\ u^*(v + w) & \text{si} \quad \varepsilon \in u \end{cases} \text{ où } w \text{ est quelconque}$$

2. Un automate spontanné $A_s = AFN - \varepsilon = (\Sigma \cup \{\varepsilon\}, E, I, F, \delta)$ c'est un AFN où en plus les symboles de Σ, ε agit sur les états de A_s : $\delta(\varepsilon, q_i) = q_j$
3. Algorithme de déterminisation d'un A_s : On cherche A_d tel que $L(A_d) = L(A_s)$:
   - $q_0 = I \cup \{$ les états de A_s,recevant une $\varepsilon-$ transition depuis les éléments de $I\}$
   - Construire le tableau de multiplication mais attention un symbole $a \in \Sigma$ peut prendre la forme $a = \varepsilon^n a \varepsilon^m,\ n, m \in \mathbb{N}$.
   - Un état acceptant de A_d est l'état qui contient au moins un état acceptant de A_s.
4. Minimisation ou réduction d'un automate A :
   - On élimine les états qui ne reçoivent aucune transition
   - On écrit E l'ensemble des états, comme l'union de deux sous ensembles, l'un contient les états acceptants et l'autre contient les non acceptants $E = E_1 \cup E_2$
   - On cherche les états obtenus par action des symboles de Σ sur les états de E_1 puis de E_2 et on regroupe les états équivalents ensemble, sachant que q_1 est équivalent à q_2 si et seulement si $\forall \alpha_i \in \Sigma,\ \alpha_i(q_1)$ appartient au même sous ensemble E_1 ou E_2 que $\alpha_i(q_2)$.
   - On reécrit comme union de sous ensembles E_1 et E_2 selon que les états de E_i sont ou ne sont pas stables par action des symboles de Σ

$$E_1 = E_{11} \cup E_{12} \cup \cdots; \quad E_2 = E_{21} \cup E_{22} \cup \cdots$$

     où les éléments de chaque sous ensemble sont équivalents
   - On repète cette itération jusqu'à ne plus pouvoir ajouter de sous ensembles. Finalement $E = G_1 \cup G_2 \cup \cdots \cup G_k$
   - L'automate minimal, ayant le même langage que A, a k états, $G_1, \cdots, G_k$, l'état initial étant celui qui contient l'état initial de A et tout état contenant un état acceptant de A est un état acceptant pour l'autonate minimal.
5. Minimisation ou réduction d'un automate A ($2^{\text{ième}}$ façon) :

   On peut réduire un automate en remarquant que deux états ayant le même langage dans le système de départ sont supposés être équivalents.