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)

Final 2021-2022 **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 (*2o points*) On considère le code de Hamming $H(3)$.

1. Donner k et n le nombre des bits d'informations et le nombre des bits des mots de code respectivement.
2. Donner la matrice de contrôle H^t et déduire la matrice génératrice.
3. On reçoit le message : $m = 01010001110010$. Corriger si besoin et décoder m.

Exercice 2 (*20 points*) On considère un code linéaire $C = (7,4)$: $n = 7$ longueur du code, $k = 4$ longeur du message initial ; dont la matrice génératirce est donnée par

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

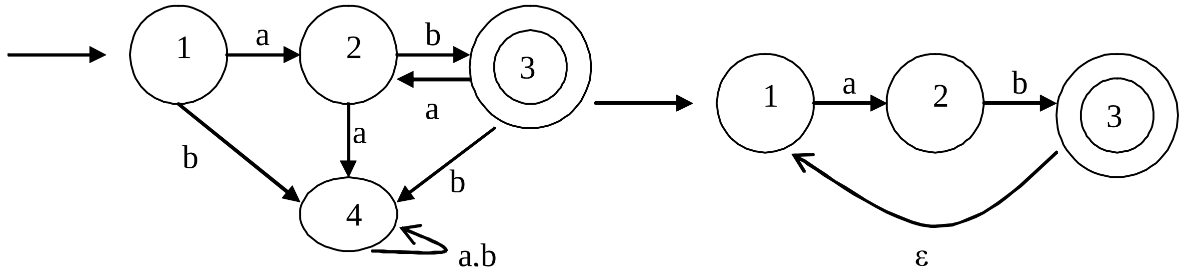
1. Donner tous les mots de code et donner $d_{\min}$ la distance minimale associée à C
2. Trouver H la matrice de contrôle associée à ce code
3. Calculer le syndrôme du vecteur récu $R = [1101101]$. Est-il un mot de code ? Sinon, corriger R.
4. Combien ce code peut-il détecter ? corriger ?

Exercice 3 (*10 points*) On considère le code plolynômial de $B^4 \longrightarrow B^7$ dont le polynôme générateur est

$$g(x) = x^3 \oplus x \oplus 1$$

1. Ce code est-il cyclique ?
2. On reçoit le mot $R = 1010111$. Vérifier, en cherchant $P_{\sigma(R)}(x)$ grâce à $g(x)$, si R est un mot de code.

Exercice 4 (*25 points*) Sur l'alphabet $\sum = \{a, b\}$, soient les automates fini donnés par :



A *B*

Deux automates sont dits équivalents s'ils ont le même langage.

1. En resolvant, si nécessaire, un système d'equations de langages, trouver les langages de A et B..
2. En déduire que A et B sont équivalents

Exercice 5 (*25 points*) On considère sur l'alphabet $\Sigma = \{a, b\}$ le langage $L = ab^*(\varepsilon + a\Sigma^*)$

1. Construire un automate minimal du langage complémentaire de L, soit un automate minimal qui reconnait $\overline{L}$
2. Lorsqu'on calcule l'automate minimal du language complémentaire reconnu par un automate A, c'est à dire, on cherche l'automate minimal qui reconnait $\bar{L}$ avec $L(A) = L$, faut-il :
   (a) calculer d'abord l'automate minimal de A puis complémenter ;
   (b) calculer d'abord le complémentaire de A et puis l'automate minimal ;
   (c) l'ordre n'est pas important.
   Justifier.

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.