Solut° final 21-22

MVA 004

Ex1 $H(r) = H(3)$, $r = 3$

1) $n = 2^r - 1 = 8 - 1 = 7$; $k = n - r = 7 - 3 = 4$ 1+1 = 2pts

2) $H' = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ Ses colonnes sont formées des éléments non nuls de $\mathbb{B}^r = \mathbb{B}^3$ dont la base est placée sur les colonnes c_i de rang $2^i = 1, 2^1, 2^2 = 4$. (3pts)

Soit $G = (c_1\ c_2 \dots c_7)$ la matrice génératrice de $H(3)$

c_i est une colonne de dimension $k = 4$, $i = 1 \dots 7$.

on a H et G donc $H' \begin{pmatrix} c_1 \\ \vdots \\ c_7 \end{pmatrix} = 0$ (1)

$$\begin{cases} c_1 + c_3 + c_5 + c_7 = 0 \\ c_2 + c_3 + c_6 + c_7 = 0 \\ c_4 + c_5 + c_6 + c_7 = 0 \end{cases}$$ (1)

$$\begin{cases} c_1 = c_3 + c_5 + c_7 \\ c_2 = c_3 + c_6 + c_7 \\ c_4 = c_5 + c_6 + c_7 \end{cases}$$

Posons $\{c_3, c_5, c_6, c_7\}$ comme la base canonique de $\mathbb{B}^4$

$c_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $c_5 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$, $c_6 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$, $c_7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$ (1)

$\Rightarrow c_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$, $c_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$, $c_4 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ (1)

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

Scanned with CamScanner

3) $\varphi : \mathbb{B}^4 \longrightarrow \mathbb{B}^7$ (2)

$m = 0101000 \mid 1110010$ (c'est un message de 2 mots

$m = \alpha\ \beta$ (1)

$H^t\alpha = (011)^t$ c'est la 6ième colonne de H^t

bien sûr $H^t\alpha \neq 0$ donc α n'est pas un mot de code,

il contient une erreur dans sa 6ième case, on corrige

α par 0101010 (2)

$H^t.\beta = (011)^t$ c'est la 6ième colonne de H^t donc $H^t\beta \neq 0$

donc β n'est pas un mot de code, β est corrigé par changement

de sa 6ième bit ça donne par 1110000 (2)

Décodage de α $\quad$ $\downarrow$ $\quad$ $\downarrow$

mot corrigé par : 0101010 $\quad$ 1110000 (1)

Décodage de m : $b_3 b_5 b_6 b_7$ .... ça donne

$0010\ 1000$. (2) + (1)

b_i : bit d'information $\in \mathbb{B}^k = \mathbb{B}^4$

Ex 2

Ex 2 (3)

$\varphi : 3^4 \longrightarrow 3^7$

1)

3^4	mots de Code
0000	0000000
1000	1000101
0100	0100110
0010	0010010
0001	0001111
1100	1100011
1010	1010111
1001	1001010
0101	0101001
0110	0110100
0011	0011101
1110	1110001
1101	1101100
1011	1011000
0111	0111011
1111	1111111

(8)

$d_{min} = d = 2$

2)

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

(4)

3) $\forall(R) = R \cdot H$

$= 001 \neq 0$ (2)

donc ce n'est pas un mot de code (2)

001 est le syndrome de $e = 0000001$ (1)

On corrige R par

$C = R + e$

$C = 1101100$ (1)

4) $d = 2$

$\begin{cases} d \geq 1+1 \\ d \geq 2\times 0+1 \end{cases}$

on détecte une erreur (1)

on corrige aucune erreur. (1)

Ex 3

1) $x^n + 1 = x^7 + 1 = (x^3 + x + 1)(x^4 + x^2 + x + 1)$ (3)

$g(x)$ divise $x^7 + 1$ donc c'est un code cyclique (1)

Scanned with CamScanner

(4)

2) $R = 1011111$

$P_R(x) = x^6 + x^4 + x^2 + x + 1$ (2)

$x^6 + x^4 + x^2 + x + 1 = (x^3 + x + 1)(x^3 + 1) + x^2$ (2)

le reste est $x^2 \equiv 100$

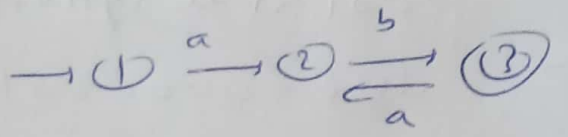
$P_{\nabla(R)}(x) = x^2$ et $\nabla(R) = 100 \neq 000$ donc R n'est pas (1)

un mot de code (1)

Ex4 $L(A)$:

4 est un état vide, non connecté vers un état acceptant par aucun chemin, on peut donc l'enlever.

(A) est équivalent à



mots de $L(A)$: ab, abab, ababab, ...

$L(A) = (ab)^+$ (10)

$L(B)$

mots de $L(B)$: ab, ab ε ab = abab, ab ε ab ε ab = ababab, ...

$L(B) = (ab)^+$ (10)

2) $L(A) = L(B)$ donc $A \sim B$. (5)

Scanned with CamScanner

2ème méthode

$L(\partial)$:
$$\begin{cases} \partial_1 = a\partial_2 + b\partial_4 \\ \partial_2 = b\partial_3 + a\partial_4 \\ \partial_3 = \varepsilon + a\partial_2 + b\partial_4 \\ \partial_4 = (a+b)\partial_4 = \phi \end{cases}$$

$$\begin{cases} \partial_1 = a\partial_2 \\ \partial_2 = b\partial_3 \\ \partial_3 = a\partial_2 + \varepsilon \end{cases} \Rightarrow \partial_3 = ab\partial_3 + \varepsilon \text{ et } \partial_3 = (ab)^*$$

$\partial_2 = b(ab)^*$ et $L(\partial) = \partial_1 = ab(ab)^* = (ab)^+$. (10)

$L(\beta)$:
$$\begin{cases} \partial_1 = a\partial_2 \\ \partial_2 = b\partial_3 \\ \partial_3 = \varepsilon\partial_1 + \varepsilon \end{cases}$$

$\partial_1 = ab(\varepsilon\partial_1 + \varepsilon)$

$\partial_1 = ab\partial_1 + ab \Rightarrow L(\beta) = (ab)^* ab = (ab)^+$. (10)

Ex5 $L = ab^*(\varepsilon + aE^*)$

1) on va d'abord construire un automate A qui reconnait L

$L = x_1 x_2^* (\varepsilon + x_3(x_4 + x_5)^*)$ et on continue par Glushkov.

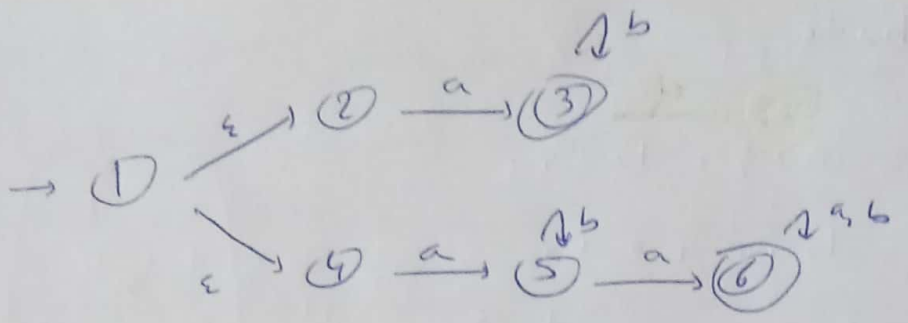
ou bien : $L = L_1 + L_2 = ab^* + ab^* a(a+b)^*$.

(4)

Scanned with CamScanner

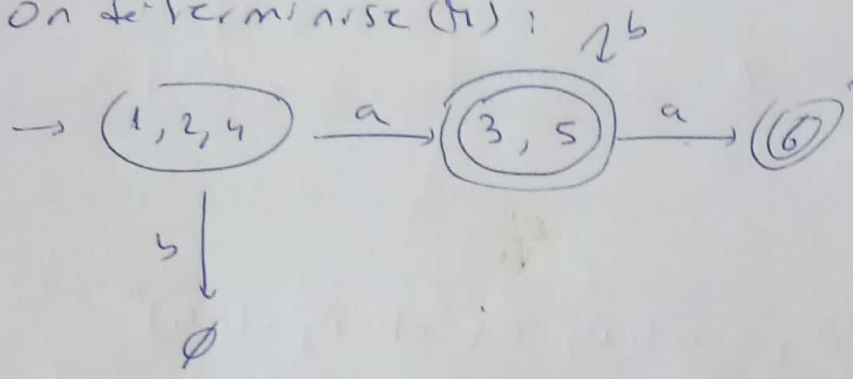
(6)

(M) :



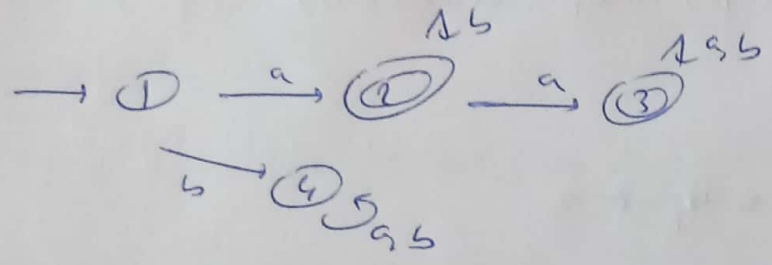
5

On déterminise (M) :



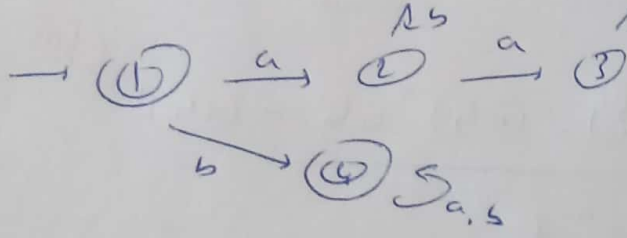
5

Le déterminisé de M est l'automate (A) :



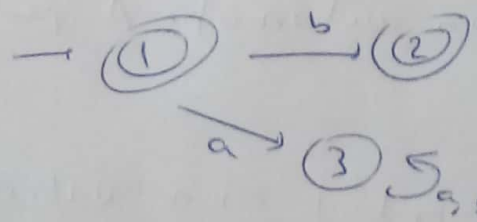
2

L'automate qui reconnait $\bar{L}$ est :

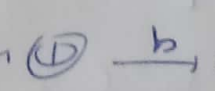


2

et l'automate minimal qui reconnait $\bar{L}$ est



Et sous forme incomplète



2

2) il est évident que si on complémente puis on minimise on obtient le bon résultat. Dans l'autre ordre, il faut vérifier que si on complémente un automate minimal, on obtient encore un automate minimal.

Scanned with CamScanner

D'un autre coté, si on part d'un automate minimal, tous ses états sont séparés, on n'a pas d'états équivalents. Ca'd l'ensemble d'états acceptants et l'ensemble d'états non acceptant vont être l'union d'états separés.

Si on complémente cet automate et on étudie l'équivalence entre les états. Rien ne va changer car cette fois ci l'ensemble d'états acceptants est celui d'états non acceptant de l'automate de départ donc forme d'états séparés, de même pour l'ensemble d'états non acceptants

Ainsi, l'ordre n'est pas important, les deux façons donnent le même résultat

5