SEBILLE Florian

TROTTIER Arthur

Compte rendu de

Codage

Sommaire

# Code d’Etalement

# 1) Création des fonctions utile pour le codeur Hadamard

# 2) Réalisation et choix du programme de test

# Code Pseudo Aléatoire

# 1) Création d’un code à longueur maximale

# 2) Création d’un code de Gold

# 3) Création d’un code JPL

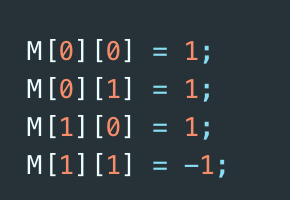
# Agent Life Cycle

Code d’Etalement

Objectif : Mise en œuvre d’un codeur de Hadamard.

# Création des fonctions utile pour le codeur d’Hadamard

Pour commencer, nous avons choisi de réfléchir sur la méthode de conception de la matrice d’Hadamard. Nous avons donc choisi de créer la première partie de cette matrice a la main (les quatre premières cases) pour pouvoir construire toutes les autres en fonction du nombre d’utilisateur. (Pour l’affichage nous avons choisi d’afficher des zéros a la place de -1 pour que ce soit plus lisible).



Ensuite en fonction du nombre d’utilisateurs on calcule la taille que doit avoir la matrice pour pouvoir assigner un mot code à chaque utilisateur. Une fois la taille de la matrice obtenue. On peut créer la matrice finale en remplissant toutes les cases.

Pour remplir le reste de la matrice on regarde où se situe la case en cours, si cette case se situe dans la partie inferieur droite de la matrice alors on regarde la valeur de la case dans la partie supérieure gauche de la matrice et on remplit la case avec l’inverse de cette case.

Exemple :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |

On parcourt la matrice en fonction des différentes tailles possible de matrice en commençant par la taille 2 car la matrice de base est de taille (2 x 2). On dit que la case d’indice [i] [j] est lié à la case d’indice [i - taille] [j - taille] et en fonction des valeurs de i et j, si les deux valeurs sont supérieures à la taille, par exemple comme la case jaune ou i et j sont égaux a trois et la taille est égale a deux alors la valeur de la case est égale à l’inverse de la case d’indice [i - taille] [j - taille], ici la case verte. Sinon si les deux indices ne sont pas supérieur à la taille comme les deux cases bleues ou il n’y a qu’une seule des deux valeurs qui est supérieure à la taille de la matrice alors elles sont égales à la valeur de la case d’indice [i - taille] [j - taille], ici la case orange. On répète cette action jusqu'à ce que la taille de la matrice voulue soit atteinte.

0

1

2

3

0 1 2 3

Une fois la matrice d’Hadamard créée on assigne un mot code à chaque utilisateur. Pour cela on créé une matrice avec autant de ligne que d’utilisateur. Pour choisir quelle ligne de la matrice d’Hadamard sera choisi pour chaque utilisateur nous avons choisi de prendre la première ligne de la matrice pour le premier utilisateur la deuxième pour le deuxième et ainsi de suite. Il aurait été préférable de choisir aléatoirement une ligne de la matrice d’Hadamard pour chaque utilisateur mais pour les tests et dans le cadre du TP nous avons choisis de ne prendre un ordre fixé.

Ensuite une fois que chaque utilisateur a son mot code. Il faut coder le mot qu’il souhaite envoyer à l’aide de son mot code. Pour cela on crée une matrice avec autant de ligne que d’utilisateurs et pour la taille des lignes on multiplie la taille des mots par la taille des mot code pour anticiper la taille de la séquence codée.

Une fois que la matrice qui contient les mots codés est créée il ne reste plus qu’à la remplir. Pour cela il faut copier autant de fois le mot code qu’il y a d’élément binaire dans le mot a coder en fonction de la valeur de ces élément. Si la valeur binaire du bit a codé est 1 alors il faut recopier le mot code à l’identique alors que si la valeur est 0 il faut copier les valeurs opposées du mot code.

Exemple :

Avec mot code : 1 0 1 0 (correspond à la deuxième ligne de la matrice d’Hadamard)

Et mot a codé : 1 0 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

Comme on peut le voir dans l’exemple la séquence obtenue est de taille 12 car le mot code est de taille quatre et le mot a codé est de taille 3. Le mot code est répéter trois fois car le mot a codé est composé de trois élément binaire si l’élément est 1 comme pour le premier élément du mot alors on recopie à l’identique le mot code alors que si l’élément binaire vaut 0 et recopie l’opposer du mot code.

# Réalisation et choix du programme de test

Pour le programme de test, nous avons choisi de demander combien il y avait d’utilisateur au début du programme puis une fois le nombre d’utilisateur connu on demande la taille des mots a codé. Ensuite une fois la taille des mots saisie il ne reste plus qu’à saisir les mots a codé bits à bits.

Générateur Pseudo-Aléatoire

Objectif : Mise en œuvre d’un code pseudo aléatoire à l’aide de Gold et JPL.

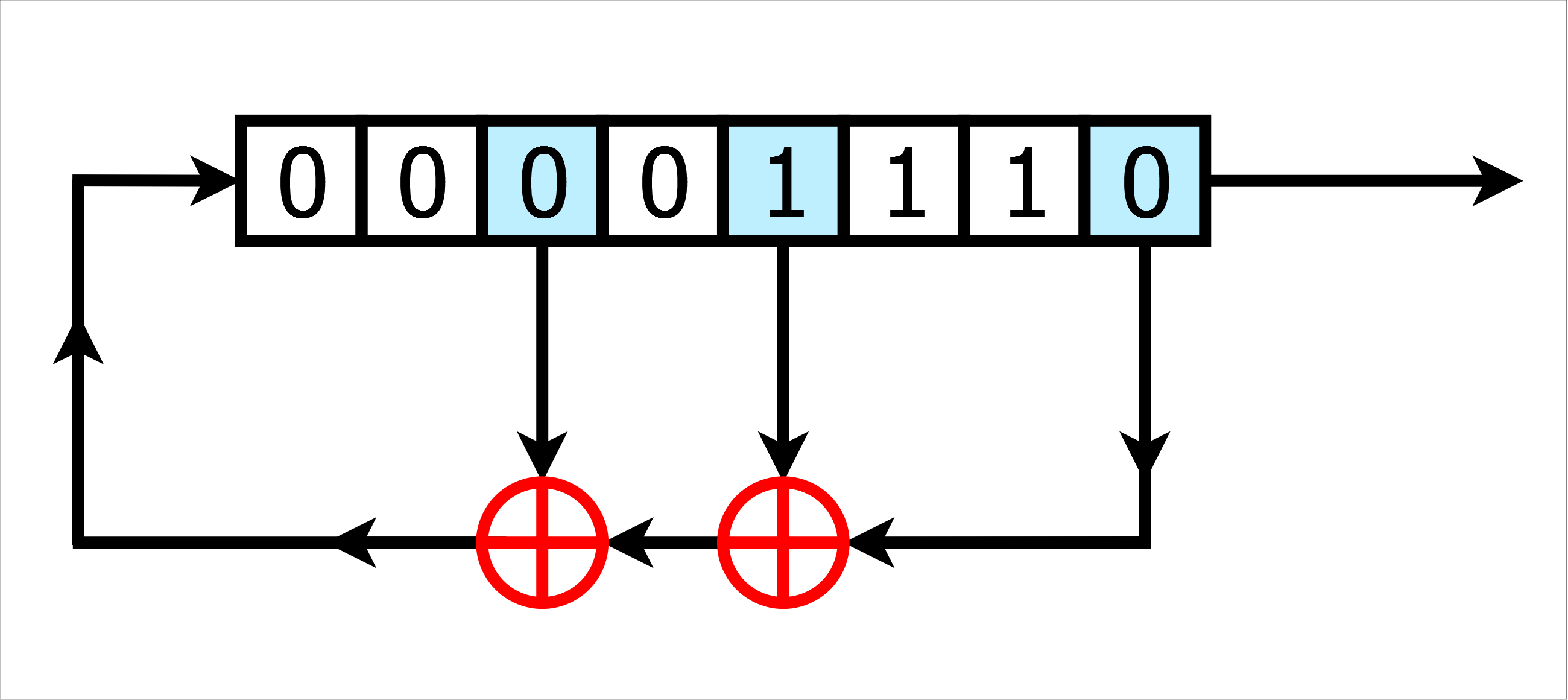
# Création d’un code à longueur maximale

La création d’un code LM (Longueur maximale) passe par la création d’un vecteur générateur contenant le nombre de registre ainsi que les différentes positions des portes XOR souhaitées.

Exemple : Si notre vecteur générateur est [2, 1] alors le code LM sera créé avec :

-> 2 registres

-> 1 porte XOR après le registre n°1



Ci-contre, le vecteur associé est [9, 3, 5, 9]

La longueur du code maximal associé sera ainsi : (2^n) - 1 avec n le nombre de registre.

Exemple : Effectuer (2^2) - 1 = 3 bascules permet d’obtenir le code maximal sans répétition.

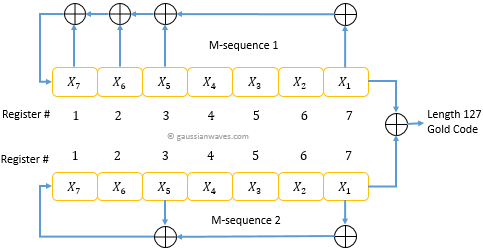
IMPORTANT : Initialiser les registres à 1 !

# 2) Création d’un code de Gold

La création d’un code de Gold revient à créer deux codes LM et effectuer un OU logique (XOR) entre les deux en sortie. Les vecteurs générateurs doivent donc contenir le même nombre de registre, les positions des portes XOR pouvant différer.

Exemple : deux vecteurs générateurs [5, 2] et [5, 3, 1] sont compatibles.

Le résultat sera un code à longueur maximal de longueur (2^5)-1 = 31



Ci-dessus, les vecteurs associés sont [7, 1, 2, 3, 4, 7] et [7, 3, 7]

La longueur est donc bien (2^7) - 1 = 127

# 3) Création d’un code JPL

La création d’un code JPL associe des codes LM de longueurs différentes. Le principe est différent du code de Gold car les vecteurs générateurs doivent avoir des nombres de registre premiers entre eux. Il existe une table spécifique permettant d’associer des vecteurs entre eux et ainsi d’obtenir un code à longueur non-maximal de longueur L1 x L2 x … x Ln avec n le nombre de code à LM et Ln la longueur du code LM associé.

Nous avons décidé de fournir au codeur une matrice contenant les vecteurs générateurs de chaque code LM ainsi qu’un indicateur de taille pour des raisons pratiques

Exemple : On fournit au codeur la matrice suivante :

2 | 2 | 1 -> vecteur générateur [2,1] pour le premier code LM

2 | 3 | 1 -> vecteur générateur [3,1] pour le deuxième code LM

2 | 5 | 2 -> vecteur générateur [5,2] pour le troisième code LM

Les nombres de registres 2, 3, 5 sont premiers entre eux.

La longueur (non-maximale) du code final est :

L = (L1 = (2^2)-1) \* (L2 = (2^3)-1) \* (L3 = (2^5)-1)

L = 651

Agent Life Cycle

Non-terminé.