Registres à décalage à rétroaction linéaire

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2013

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

Introduction

Un registre à décalage à rétroaction linéaire (ou LFSR, pour *Linear Feedback Shift Register* en anglais) est un circuit électronique ou une fonction logicielle qui permet de générer une suite récurrente linéaire donnée. Dans le cadre de ce sujet, nous nous restreindrons exclusivement au cas des suites récurrentes linéaires définies sur le corps des entiers modulo 2 : $\mathbb{Z}/2\mathbb{Z}$.

Définition 1 *Un LFSR de longueur ℓ, pour tout entier positif ℓ, est défini par la donnée de ℓ coefficients binaires $(c_1, c_2, \ldots, c_\ell) \in (\mathbb{Z}/2\mathbb{Z})^\ell$. À partir d'un état initial $S_0 = (s_0, s_1, \ldots, s_{\ell-1}) \in (\mathbb{Z}/2\mathbb{Z})^\ell$, cet LFSR va générer $(s_i)_{i\geq 0}$, la suite récurrente linéaire d'ordre ℓ définie sur $\mathbb{Z}/2\mathbb{Z}$ par la relation de rétroaction*

$$s_{i+\ell} = (c_1 s_{i+\ell-1} + c_2 s_{i+\ell-2} + \cdots + c_\ell s_i) \bmod 2, \quad \textit{pour tout } i \geq 0.$$

Définition 2 *Soit $\mathcal{R}$ un LFSR de longueur ℓ et de coefficients $(c_1, c_2, \ldots, c_\ell)$. Son polynôme de rétroaction est défini comme étant le polynôme $C(X) \in (\mathbb{Z}/2\mathbb{Z})[X]$ suivant :*

$$C(X) = 1 + c_1 X + c_2 X^2 + \cdots + c_\ell X^\ell.$$

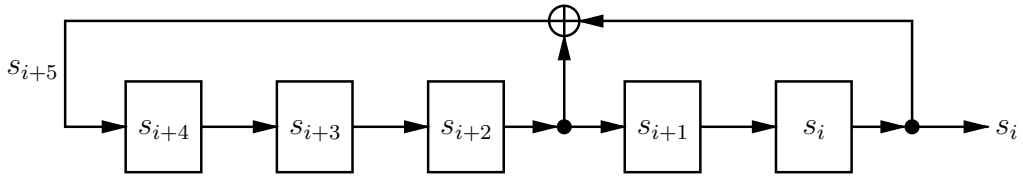
La donnée d'une longueur ℓ et d'un polynôme de rétroaction $C(X)$ permettent de caractériser uniquement un LFSR. On notera ainsi $\mathcal{R} = \langle \ell, C(X) \rangle$.

Définition 3 *Soit $\mathcal{R}$ un LFSR de longueur ℓ qui génère la suite $(s_i)_{i\geq 0}$ à partir de l'état initial $S_0 = (s_0, s_1, \ldots, s_{\ell-1})$. Pour tout entier $i \in \mathbb{N}$, l'état interne de $\mathcal{R}$ à l'itération i est défini comme la suite de ℓ bits $S_i = (s_i, s_{i+1}, \ldots, s_{i+\ell-1})$.*

Il est à noter que la connaissance de l'état interne S_i suffit à elle seule à déterminer la valeur du bit $s_{i+\ell}$.
Par exemple, le LFSR de longueur $\ell = 5$ et de polynôme de rétroaction $C(X) = 1 + X^3 + X^5$ génère une suite récurrente linéaire $(s_i)_{i\geq 0}$ qui vérifie l'équation

$$s_{i+5} = (s_{i+2} + s_i) \bmod 2 \quad \text{pour tout } i \geq 0.$$

Un circuit électronique implémentant cet LFSR pourra ainsi prendre la forme suivante :



Dans ce schéma, le symbole $\oplus$ représente l'addition modulo 2 : $x \oplus y = (x + y) \bmod 2$. L'état interne S_i du registre à l'itération i est donné par les 5 bits $S_i = (s_i, s_{i+1}, \ldots, s_{i+4})$. À chaque itération du registre, ces bits sont décalés d'un cran vers la droite, et la nouvelle valeur s_{i+5} est injectée dans le registre par la gauche. Le i-ème bit s_i de la suite $(s_i)_{i\geq 0}$ générée par le LFSR est quant à lui lu sur la sortie à droite du registre lors de la i-ème itération.
En partant de l'état initial $(s_0, \ldots, s_4) = (0, 0, 1, 1, 0)$, cet LFSR va ainsi générer la suite $(s_i)_{i\geq 0} = (0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, \ldots)$.

Les suites de bits générées par certains LFSR présentent des propriétés statistiques intéressantes, ce qui les rend très utiles en cryptographie, particulièrement en ce qui concerne la génération de nombres pseudo-aléatoires et le chiffrement par flot. Ainsi, les systèmes de chiffrement A5/1 (utilisé pour la téléphonie GSM), E0 (pour Bluetooth) ou encore CSS (pour les DVD) reposent sur des LFSR.

Ce sujet se divise en deux parties relativement peu dépendantes l'une de l'autre. Dans la première partie, nous manipulerons des LFSR aléatoires pour nous familiariser avec ces objets et étudier certaines de leurs propriétés. L'objectif de la seconde partie sera de mettre au point des algorithmes pour retrouver le plus petit LFSR générant une suite de bits donnée.

Préliminaires

Avant toute chose, considérons la suite d'entiers $(u_k)_{k\geq 0}$ définie par

$$u_k = \begin{cases} \text{votre } u_0 \text{ (à reporter sur votre fiche réponse)} & \text{si } k = 0, \text{ et} \\ 15\,091 \times u_{k-1} \bmod 64\,007 & \text{si } k > 0. \end{cases}$$

Question 1 *Donnez les valeurs des éléments* **a)** u_{10}, **b)** u_{100}, **c)** u_{1000}.

Assurez-vous de pré-calculer et de stocker suffisamment de valeurs de cette suite (de l'ordre de 100 000) de manière à pouvoir y accéder en temps constant par la suite.
Comme nous allons manipuler principalement des éléments de $\mathbb{Z}/2\mathbb{Z}$ dans ce sujet, nous définissons la suite de bits $(v_k)_{k\geq 0}$ par $v_k = u_k \bmod 2$ pour tout $k \geq 0$. De même, pour tous ℓ et $k \in \mathbb{N}$, nous noterons $V_{\ell,k}$ le ℓ-uplet formé de ℓ éléments consécutifs de cette suite en commençant par l'élément v_k :

$$V_{\ell,k} = (v_k, v_{k+1}, \ldots, v_{k+\ell-1}).$$

1 Manipulation de LFSR

1.1 Génération de LFSR aléatoires

Pour tous ℓ et $k \in \mathbb{N}$, on note $C_{\ell,k}(X)$ le polynôme de rétroaction

$$C_{\ell,k}(X) = 1 + v_k X + v_{k+1} X^2 + \cdots + v_{k+\ell-1} X^\ell,$$

et $\mathcal{R}_{\ell,k} = \langle \ell, C_{\ell,k}(X) \rangle$ le LFSR de longueur ℓ correspondant. En d'autres termes, les coefficients $(c_1, c_2, \ldots, c_\ell)$ de $\mathcal{R}_{\ell,k}$ sont donnés par le ℓ-uplet $V_{\ell,k}$ défini ci-dessus.
Il est à noter que les polynômes $C_{\ell,k}(X)$ ne sont pas forcément de degré ℓ. Nous dirons ainsi d'un LFSR $\mathcal{R} = \langle \ell, C(X) \rangle$ qu'il est singulier lorsque son polynôme de rétroaction $C(X)$ est de degré strictement inférieur à ℓ, c'est-à-dire que $c_\ell = 0$.

Question 2 *Donnez les degrés des polynômes* **a)** $C_{5,0}(X)$, **b)** $C_{10,0}(X)$, **c)** $C_{50,0}(X)$.

Nous souhaitons désormais pouvoir simuler l'exécution d'un LFSR afin d'obtenir la suite de bits correspondante $(s_i)_{i\geq 0}$.

Question 3 *Pour chacun des couples $(\mathcal{R}, S_0)$ ci-dessous, donnez les 10 bits suivants s_ℓ, $s_{\ell+1}$, …, $s_{\ell+9}$ de la suite générée par le LFSR $\mathcal{R}$ de longueur ℓ à partir de l'état initial $S_0 = (s_0, s_1, \ldots, s_{\ell-1})$:*
a) $(\mathcal{R}_{5,0}, V_{5,100})$, ***b)*** $(\mathcal{R}_{10,0}, V_{10,100})$, ***c)*** $(\mathcal{R}_{50,0}, V_{50,100})$.

1.2 LFSR périodiques

Question à développer pendant l'oral : Quel est le nombre d'états internes possibles d'un LFSR $\mathcal{R}$ de longueur ℓ donné ? Montez alors que la suite générée $(s_i)_{i\geq 0}$ est ultimement périodique, c'est-à-dire qu'il existe $i_0 \in \mathbb{N}$ et $N \in \mathbb{N}^*$ tels que pour tout $i \geq i_0$ on ait $s_{i+N} = s_i$.
On appellera période de la suite générée le plus petit entier $N > 0$ vérifiant cette condition. De plus, un LFSR sera dit périodique lorsque $i_0 = 0$.

En cherchant à caractériser ces LFSR périodiques, nous nous intéressons ici aux LFSR non-singuliers, c'est-à-dire ceux dont le coefficient c_ℓ est non-nul.

Question à développer pendant l'oral : Montrez que tout LFSR non-singulier est réversible, c'est-à-dire qu'étant donnés uniquement son polynôme de rétroaction et un état interne $S_i = (s_i, s_{i+1}, \ldots, s_{i+\ell-1})$ il est possible de retrouver les éléments précédents s_{i-1}, s_{i-2}, s_{i-3}, … de la suite $(s_i)_{i\geq 0}$ générée par cet LFSR. Qu'en est-il de la réciproque ? Déduisez-en alors qu'un LFSR est périodique si et seulement si il est non-singulier.

Pour tous ℓ et $k \in \mathbb{N}$, nous notons alors $C'_{\ell,k}(X)$ le polynôme de rétroaction

$$C'_{\ell,k}(X) = 1 + v_k X + v_{k+1} X^2 + \cdots + v_{k+\ell-2} X^{\ell-1} + X^\ell,$$

et $\mathcal{R}'_{\ell,k} = \langle \ell, C'_{\ell,k}(X) \rangle$ le LFSR non-singulier de longueur ℓ correspondant. De manière équivalente, il s'agit du LFSR dont les coefficients $(c_1, \ldots, c_{\ell-1})$ sont donnés par $V_{\ell-1,k}$, et dont le coefficient c_ℓ est égal à 1.

Question 4 *Pour chacun des couples $(\mathcal{R}, S_i)$ ci-dessous, donnez les 10 bits précédents s_{i-10}, …, s_{i-2}, s_{i-1} de la suite générée par le LFSR non-singulier $\mathcal{R}$ de longueur ℓ et dont l'état interne à l'itération i est donné par $S_i = (s_i, s_{i+1}, \ldots, s_{i+\ell-1})$:*
a) $(\mathcal{R}'_{5,0}, V_{5,100})$, ***b)*** $(\mathcal{R}'_{10,0}, V_{10,100})$, ***c)*** $(\mathcal{R}'_{50,0}, V_{50,100})$.

Question 5 *Pour chacun des LFSR non-singuliers ci-dessous, donnez la période N de la suite $(s_i)_{i\geq 0}$ générée à partir de l'état initial $S_0 = (s_0, s_1, \ldots, s_{\ell-1}) = (0, 0, \ldots, 0, 1)$:*
a) $\mathcal{R}'_{5,0}$, ***b)*** $\mathcal{R}'_{10,0}$, ***c)*** $\mathcal{R}'_{15,0}$.
Indication. *Vous pouvez pour cela vous servir du fait que la suite de bits $(s_i)_{i\geq 0}$ est périodique de période N si et seulement si la suite des états internes $(S_i)_{i\geq 0}$ l'est aussi.*

Question à développer pendant l'oral : Donnez des bornes (inférieure et supérieure) sur la période que la suite de bits générée par un LFSR peut atteindre, en fonction de sa longueur ℓ. Vérifiez expérimentalement que ces bornes sont effectivement atteintes : donnez des exemples de couples $(\mathcal{R}, S_0)$ tels que la période de la suite générée par le LFSR $\mathcal{R}$ à partir de l'état initial S_0 soit minimale ou maximale par rapport à la longueur ℓ de cet LFSR.

Question à développer pendant l'oral : Justifiez la proposition suivante : si un LFSR de longueur ℓ génère une suite de bits de période maximale par rapport à ℓ à partir d'un état non-nul donné, alors ce même LFSR générera la même suite de bits (à un décalage d'indices près) à partir de tout état initial non-nul. On parlera alors de LFSR de période maximale.

2 Complexité linéaire

Étant donnée une suite de n bits $T = (t_0, t_1, \ldots, t_{n-1})$, nous appelons complexité linéaire de T le plus petit entier positif $L(T)$ tel qu'il existe un LFSR $\mathcal{R}$ de longueur $L(T)$ dont les n premiers bits $(s_0, s_1, \ldots, s_{n-1})$ générés à partir de l'état initial $S_0 = (t_0, t_1, \ldots, t_{L(T)-1})$ sont exactement la suite T.
Par exemple, la complexité linéaire de la suite $T = (0, 1, 1, 0, 0, 1, 0, 0)$ est $L(T) = 4$, car cette suite est générée par le LFSR $\mathcal{R} = \langle 4, 1 + X + X^4 \rangle$, mais n'est générée par aucun LFSR de longueur moindre.
Question à développer pendant l'oral : Montrez que, pour toute suite T de n bits, nous avons $0 \leq L(T) \leq n$. Ces bornes sont-elles atteintes ?

2.1 Recherche exhaustive

Dans un premier temps, nous allons calculer $L(T)$ grâce à un algorithme naïf, en recherchant exhaustivement le plus petit LFSR générant la suite T parmi tous les LFSR de longueur comprise entre 0 et n, la longueur de T.

Question 6 *Pour chacune des suites T de longueur n ci-dessous, indiquez leur complexité linéaire $L(T)$. Veuillez aussi préciser les 10 bits suivants s_n, s_{n+1}, …, s_{n+9} de la suite générée par le LFSR de longueur $L(T)$ correspondant.*
a) $V_{5,0}$, ***b)*** $V_{10,0}$, ***c)*** $V_{20,0}$, ***d)*** $V_{30,0}$.

Question à développer pendant l'oral : Quelle est la complexité de votre algorithme pour une suite T de longueur n et de complexité linéaire $L(T)$?

2.2 Résolution du système linéaire

Une méthode plus efficace pour calculer la complexité linéaire d'une suite $T = (t_0, t_1, \ldots, t_{n-1})$ est de remarquer qu'un LFSR de longueur ℓ génère T si et seulement si ses coefficients c_1, c_2, …, c_ℓ vérifient un système de $n - \ell$ équations linéaires définies sur $\mathbb{Z}/2\mathbb{Z}$:

$$\begin{cases} c_1 t_{\ell-1} + c_2 t_{\ell-2} + \cdots + c_\ell t_0 \equiv t_\ell \pmod 2, \\ c_1 t_\ell + c_2 t_{\ell-1} + \cdots + c_\ell t_1 \equiv t_{\ell+1} \pmod 2, \\ \quad\vdots \qquad\quad \vdots \qquad\qquad \vdots \\ c_1 t_{n-2} + c_2 t_{n-3} + \cdots + c_\ell t_{n-\ell-1} \equiv t_{n-1} \pmod 2. \end{cases}$$

Ainsi, pour déterminer s'il existe un LFSR de longueur ℓ générant T, il suffit de tenter de résoudre ce système à ℓ inconnues dans $\mathbb{Z}/2\mathbb{Z}$: chaque solution nous donnera un tel LFSR ; et réciproquement, une absence de solution nous permettra d'affirmer que $L(T) > \ell$.

Afin de résoudre un tel système, une méthode est de l'exprimer sous la forme matricielle $\mathbf{M} \cdot \mathbf{c} \equiv \mathbf{t} \pmod 2$, avec

$$\mathbf{M} = \begin{pmatrix} t_{\ell-1} & t_{\ell-2} & \cdots & t_0 \\ t_\ell & t_{\ell-1} & \cdots & t_1 \\ \vdots & \vdots & & \vdots \\ t_{n-2} & t_{n-3} & \cdots & t_{n-\ell-1} \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_\ell \end{pmatrix}, \text{ et } \quad \mathbf{t} = \begin{pmatrix} t_\ell \\ t_{\ell+1} \\ \vdots \\ t_{n-1} \end{pmatrix}.$$

Dans la suite, nous notons $\mathbf{L}_i$, pour $0 \leq i < n - \ell$, les lignes de la matrice $\mathbf{M}$ ainsi construite.

Nous appliquons alors la méthode du pivot de Gauss : grâce à des manipulations élémentaires des lignes de cette matrice, comme l'addition modulo 2 d'une ligne à une autre (notée $\mathbf{L}_i \leftarrow \mathbf{L}_i \oplus \mathbf{L}_j$) ou l'échange de deux lignes (noté $\mathbf{L}_i \leftrightarrow \mathbf{L}_j$), et en appliquant ces mêmes transformations au vecteur de droite $\mathbf{t} = {}^t(t_\ell, t_{\ell+1}, \ldots, t_{n-1})$, il est possible d'obtenir une représentation équivalente du système de la forme $\mathbf{M}' \cdot \mathbf{c} \equiv \mathbf{t}' \pmod 2$ où $\mathbf{M}'$ est une matrice échelonnée réduite.

Une matrice à coefficients dans $\mathbb{Z}/2\mathbb{Z}$ est dite échelonnée réduite lorsque

– le nombre de 0 au début de chaque ligne augmente strictement de ligne en ligne, jusqu'à ce qu'il ne reste plus que des lignes nulles ; et que
– le premier élément non-nul de chaque ligne non-nulle (appelé pivot) est le seul élément non-nul dans sa colonne.

Par exemple, la matrice 5×6 suivante (où les $*$ représentent des coefficients arbitraires) est échelonnée réduite :

$$\begin{pmatrix} 1 & * & 0 & 0 & * & * & 0 \\ 0 & 0 & 1 & 0 & * & * & 0 \\ 0 & 0 & 0 & 1 & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Une fois le système linéaire mis sous forme échelonnée réduite, sa résolution est immédiate. Ainsi, par exemple, pour la suite $T = (1, 0, 1, 0, 1, 1, 1)$, le système linéaire correspondant pour $\ell = 4$ est donné par l'équation matricielle suivante :

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \pmod 2.$$

En appliquant dans l'ordre les transformations $\mathbf{L}_1 \leftrightarrow \mathbf{L}_2$, $\mathbf{L}_3 \leftarrow \mathbf{L}_3 \oplus \mathbf{L}_1$, $\mathbf{L}_3 \leftarrow \mathbf{L}_3 \oplus \mathbf{L}_2$ et $\mathbf{L}_1 \leftarrow \mathbf{L}_1 \oplus \mathbf{L}_3$ à ce système, on obtient alors le système équivalent suivant, dont la matrice est échelonnée réduite :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \pmod 2.$$

On peut alors résoudre directement pour obtenir $c_1 = 0$, $c_2 = (c_4 + 1) \bmod 2$, $c_3 = 1$ et $c_4 \in \{0, 1\}$. Il existe donc deux LFSR de longueur 4 qui génèrent la suite T. Leurs polynômes de rétroaction sont $1 + X^2 + X^3$ et $1 + X^3 + X^4$, respectivement.

Il est à noter aussi que si nous avions essayé de résoudre le système correspondant à $\ell = 3$, nous aurions obtenu le système réduit suivant :

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \pmod 2.$$

La dernière ligne de ce système, qui implique l'égalité $0 \cdot c_1 + 0 \cdot c_2 + 0 \cdot c_3 \equiv 1 \pmod 2$, est une contradiction manifeste, indiquant ainsi que le système n'a pas de solution.
Une fois la matrice mise sous forme échelonnée réduite, seules des lignes de ce type peuvent vous empêcher de résoudre le système, car les lignes non-nulles de la matrice sont toutes linéairement indépendantes et ne peuvent donc mener à de telles contradictions. Ainsi, le système considéré admet au moins une solution si et seulement si les éléments du vecteur de droite $\mathbf{t}'$ correspondants aux lignes nulles de la matrice échelonnée réduite $\mathbf{M}'$ sont eux aussi tous nuls.

Utilisez les explications précédentes pour écrire un algorithme qui résout de tels systèmes linéaires afin de trouver le plus petit LFSR générant une suite donnée, et ainsi calculer la complexité linéaire de cette suite.

Question 7 *Pour chacune des suites T de longueur n ci-dessous, indiquez leur complexité linéaire $L(T)$. Veuillez aussi préciser les 10 bits suivants s_n, s_{n+1}, …, s_{n+9} de la suite générée par le LFSR de longueur $L(T)$ correspondant.*
a) $V_{50,0}$, ***b)*** $V_{100,0}$, ***c)*** $V_{150,0}$.

Question à développer pendant l'oral : Quelle est la complexité de votre algorithme pour une suite T de longueur n et de complexité linéaire $L(T)$?

2.3 Algorithme de Berlekamp-Massey

Initialement découvert par Elwyn Berlekamp en 1967 pour décoder les codes BCH, c'est en 1969 que James Massey verra l'application possible de cet algorithme aux LFSR et en proposera une formulation simplifiée.
Dans le contexte qui nous intéresse, l'algorithme de Berlekamp-Massey repose principalement sur la propriété suivante.

Propriété 4 *Soient $T_n = (t_0, t_1, \ldots, t_{n-1})$ une suite de n bits de complexité linéaire $\ell_n = L(T_n)$, et $\mathcal{R}_n$ un LFSR de longueur ℓ_n qui génère la suite T_n. Nous notons $C_n(X) = 1 + c_{n,1}X + c_{n,2}X^2 + \cdots + c_{n,\ell}X^\ell$ son polynôme de rétroaction.*
Étant donné un bit supplémentaire t_n, nous définissons alors $d_n \in \mathbb{Z}/2\mathbb{Z}$ comme la différence entre t_n et le $(n+1)$-ième bit effectivement généré par $\mathcal{R}_n$:

$$d_n = (t_n + c_{n,1}t_{n-1} + c_{n,2}t_{n-2} + \cdots + c_{n,\ell}t_{n-\ell}) \bmod 2.$$

Nous avons alors :

i) Le LFSR $\mathcal{R}_n$ génère la suite $T_{n+1} = (t_0, t_1, \ldots, t_{n-1}, t_n)$ si et seulement si $d_n = 0$.
ii) Si $d_n = 0$, alors $L(T_{n+1}) = \ell_n$.

iii) *Si $d_n = 1$, soient $m < n$ le plus grand entier tel que $L(T_m) < \ell_n$ pour $T_m = (t_0, t_1, \ldots, t_{m-1})$, ainsi que $\mathcal{R}_m = \langle L(T_m), C_m(X) \rangle$ un LFSR de longueur minimale qui génère T_m. Par convention, si $\ell_n = 0$, nous poserons $m = -1$ et $C_m(X) = 1$. Alors, le LFSR $\mathcal{R}_{n+1} = \langle \ell_{n+1}, C_{n+1}(X) \rangle$ génère T_{n+1} et est de longueur minimale pour cette propriété, avec*

$$\ell_{n+1} = \begin{cases} \ell_n, & \text{si } \ell_n > n/2, \text{ ou} \\ n + 1 - \ell_n, & \text{si } \ell_n \leq n/2, \end{cases}$$

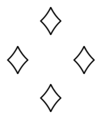
et $C_{n+1}(X) = (C_n(X) + C_m(X) \cdot X^{n-m}) \bmod 2$.

Grâce à cette propriété, retrouvez et implantez l'algorithme de Berlekamp-Massey : en partant du LFSR trivial $\mathcal{R}_0 = \langle 0, 1 \rangle$ générant la suite vide T_0, ajustez cet LFSR en ajoutant les bits de T les uns après les autres à la suite qu'il génère.

Question 8 *Pour chacune des suites T de longueur n ci-dessous, indiquez leur complexité linéaire $L(T)$. Veuillez aussi préciser les 10 bits suivants s_n, s_{n+1}, …, s_{n+9} de la suite générée par le LFSR de longueur $L(T)$ correspondant.*
a) $V_{600,0}$, ***b)*** $V_{1200,0}$, ***c)*** $V_{4000,0}$.

Question à développer pendant l'oral : Quelle est la complexité de votre algorithme pour une suite T de longueur n et de complexité linéaire $L(T)$?

Question à développer pendant l'oral : Démontrez la Propriété 4.



Fiche réponse type: Registres à décalage à rétroaction linéaire

$\widetilde{\mathbf{u_0}}$ **:** 42

Question 1

a) 15421

b) 50807

c) 44156

Question 2

a) 4

b) 10

c) 49

Question 3

a) $1, 1, 1, 0, 0, 1, 1, 1, 1, 0$

b) $1, 0, 1, 1, 1, 1, 1, 1, 0, 1$

c) $1, 1, 0, 1, 0, 0, 0, 0, 0, 0$

Question 4

a) $0, 0, 0, 1, 1, 1, 1, 0, 1, 0$

b) $0, 0, 1, 0, 0, 0, 0, 0, 0, 1$

c) $0, 1, 1, 1, 1, 1, 1, 0, 1, 1$

Question 5

a) 15

b) 315

c) 5115

Question 6

a) $2,\ 1, 0, 1, 0, 1, 0, 1, 0, 1, 0$

b) $5,\ 1, 0, 0, 0, 1, 1, 1, 1, 1, 0$

c) $10,\ 0, 0, 1, 1, 0, 0, 1, 1, 1, 1$

d) $15,\ 0, 0, 0, 1, 1, 0, 1, 0, 0, 0$

Question 7

a) $25,\ 1, 0, 0, 0, 1, 0, 1, 1, 1, 0$

b) $50,\ 0, 1, 0, 0, 0, 0, 1, 0, 0, 0$

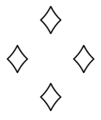
c) $73,\ 0, 1, 0, 0, 0, 1, 0, 1, 1, 0$

Question 8

a) $299,\ 1, 1, 0, 0, 1, 1, 0, 0, 0, 1$

b) $599,\ 0, 1, 0, 0, 0, 0, 0, 0, 1, 1$

c) $2000,\ 0, 1, 1, 1, 0, 1, 0, 1, 0, 0$



Fiche réponse: Registres à décalage à rétroaction linéaire

Nom, prénom, u_0: .....

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

d)

Question 7

a)

b)

c)

Question 8

a)

b)

c)