Projet LFSR

Calcul Symbolique

—Licence Informatique / Deuxième Année—

Le projet consiste à implanter quelques fonctions de manipulation des LFSR. Le langage de programmation choisi sera OCaml et le travail sera effectué en binôme. Un rapport incluant la description des modules utilisés et la présentation des principales fonctions, ainsi que le code (source et exécutable) accompagné de jeux d'essai bien choisis, seront rassemblés en une unique archive `nom1_nom2.zip`, avec `nom1` et `nom2` les étudiants composant le binôme. L'archive sera déposée sur la plate-forme UniversiTICE selon votre groupe de TP avant le vendredi 5 mai 2017 à minuit. Une soutenance suivra.

1 Polynômes à coefficient dans $\mathbb{F}_2$

Le corps $\mathbb{F}_2$ est l'unique corps à deux éléments, notés 0 et 1. Les deux lois $\oplus$ et $\otimes$ sont définies de la manière suivante :

$\oplus$	0	1
0	0	1
1	1	0

$\otimes$	0	1
0	0	0
1	0	1

▸ **Exercice 1.**

1. Quelle est la définition d'un corps commutatif ?
2. Montrez que $\mathbb{F}_2$ est bien un corps commutatif.
3. Les opérations $\oplus$ et $\otimes$ sont assimilables à des opérations logiques. Lesquelles ?

▸ **Exercice 2.**

1. Trouvez une structure de donnée permettant de représenter les polynômes à coefficient dans $\mathbb{F}_2$.
2. Programmez la somme, le produit par la méthode de Karatsuba, la division rapide par la méthode de Newton, ainsi que l'algorithme d'Euclide de $\mathbb{F}_2[X]$.
3. L'**ordre** d'un polynôme $P(X) \in \mathbb{F}_2[X]$ de degré supérieur à 1 et de terme constant égal à 1 est le plus petit nombre n non nul tel que $X^n \equiv 1 \mod P(X)$. Ainsi l'ordre de $P(X) = X^3 + X^2 + X + 1$, qui vaut 4, est déterminé par les itérations suivantes :

$$\begin{aligned} X^1 &\equiv X \mod P(X) \\ X^2 &\equiv X^2 \mod P(X) \\ X^3 &\equiv X^2 + X + 1 \mod P(X) \\ X^4 &\equiv X.X^3 \equiv X.(X^2 + X + 1) \equiv X^3 + X^2 + X \equiv 1 \mod P(X) \end{aligned}$$

   Proposez une fonction qui calcule l'ordre d'un polynôme de degré supérieur à 1 et de terme constant égal à 1.
4. Un polynôme de $\mathbb{F}_2[X]$ est dit **irréductible** si aucun polynôme non constant, de degré strictement inférieur, ne le divise. De façon exhaustive, programmez une fonction qui teste le caractère irréductible d'un polynôme. Il faut diviser $P(X) = X^3 + X + 1$ par 6 polynômes pour pouvoir décider : X, $X + 1$, X^2, $X^2 + 1$, $X^2 + X$ et $X^2 + X + 1$. Vérifiez qu'il est bien irréductible. Écrivez une fonction qui calcule un polynôme irréductible de degré $n > 0$.
5. Un polynôme de degré $n > 0$ est **primitif** s'il est irréductible et d'ordre $2^n - 1$. Programmez une fonction qui calcule un tel polynôme.

▸ **Exercice 3.** Cherchez des informations sur le théorème de la division suivant les puissances croissantes. Décrire l'algorithme associé et le programmer dans le contexte des polynômes à coefficient dans $\mathbb{F}_2$

2 Registre à décalage à rétroaction linéaire

Un registre à décalage à rétroaction linéaire (LFSR : Linear Feedback Shift Register) est un dispositif permettant de produire une suite de bits ultimement périodique à coefficient dans $\mathbb{F}_2$. C'est un dispositif relativement simple et léger qui peut être réalisé électroniquement. Les LFSR sont utilisés en cryptographie pour engendrer des suites de nombres pseudo-aléatoires. Ils sont définis par une suite récurrente :

$$\begin{cases} r_0, \ldots, r_{\ell-1} \in \mathbb{F}_2 \\ r_n = \alpha_1 \otimes r_{n-1} \oplus \cdots \oplus \alpha_\ell \otimes r_{n-\ell} \quad \text{si } n \geq \ell \end{cases}$$

où $\ell > 0$ est la longueur du LFSR. Les valeurs $\alpha_i \in \mathbb{F}_2$ sont appelés **branchements**. On note V_i ($i \geq 0$) le vecteur (ou encore **registre**) qui indique l'état du LFSR :

$$V_i = \begin{pmatrix} r_i \\ r_{i+1} \\ \vdots \\ r_{i+\ell-1} \end{pmatrix}.$$

Le registre V_0 correspond à l'état initial du LFSR.

▸ **Exercice 4.**

1. Déterminez une structure de donnée représentant les LFSR.
2. Connaissant le LFSR, programmez le calcul efficace de la nième valeur r_n ($n \geq 0$) de la suite $(r_i)_{i \geq 0}$.
3. On considère la matrice suivante :

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ \alpha_\ell & \alpha_{\ell-1} & \cdots & \alpha_3 & \alpha_2 & \alpha_1 \end{pmatrix}$$

   Prouvez que $M^n V_0 = V_n$.
4. Montrez qu'un LFSR produit bien une suite ultimement périodique, de période inférieure à $2^\ell - 1$. On pourra pour cela considérer le nombre d'états possibles du LFSR. En déduire dans le cas $\alpha_\ell = 1$ que la suite est périodique en étudiant le déterminant de la matrice M.
5. Déterminez les branchements des deux LFSR suivants :

$$\begin{cases} r_0 = r_3 = r_6 = r_9 = 1, \\ r_1 = r_2 = r_4 = r_5 = r_7 = r_8 = 0, \\ r_n = r_{n-1} \oplus r_{n-3} \oplus r_{n-4} \oplus r_{n-7} \oplus r_{n-10} \quad \text{si } n \geq 10\,; \end{cases}$$

$$\begin{cases} r_0 = 1, \\ r_1 = r_2 = 0, \\ r_n = r_{n-3} \qquad \text{si } n \geq 3. \end{cases}$$

   Quelles sont leurs vingt premières valeurs ? Que constatez-vous ?

À chaque LFSR, on a associe un couple de polynômes de $\mathbb{F}_2[X]$:

$$G(X) = \sum_{i=0}^{\ell-1} \left(\sum_{j=0}^{i} \alpha_{i-j} r_j \right) X^i,$$
$$R(X) = \alpha_0 + \alpha_1 X + \ldots + \alpha_\ell X^\ell,$$

où $\alpha_0 = 1$. Le polynôme $R(X)$ est appelé **polynôme de rétroaction**. Le couple $(G(X), R(X))$ caractérise un LFSR car il permet de retrouver les valeurs initiales de la suite et les branchements nécessaires à son fonctionnement.

▸ **Exercice 5.**

1. Montrez que $S(X)R(X) = G(X)$ où la série $S(X)$ est déterminée à partir des termes de la suite $(r_i)_{i\geq 0}$:

$$S = \sum_{i\geq 0} r_i X^i = r_0 + r_1 X + r_2 X^2 + r_3 X^3 + \cdots$$

2. Programmez la fonction qui calcule le couple $(G(X), R(X))$ à partir d'un LFSR.
3. Connaissant le couple $(G(X), R(X))$, programmez la fonction qui retourne le LFSR correspondant.
4. Si les polynômes $G(X)$ et $R(X)$ admettent un facteur commun $T(X)$, le LFSR associé au couple $(G(X)/T(X), R(X)/T(X))$ crée le même flux de valeurs que le LFSR associé au couple $(G(X), R(X))$. Vérifiez que les flux de valeurs des deux LFSR de la question 5 de l'exercice 4 sont bien identiques.
5. Connaissant le couple $(G(X), R(X))$ d'un LFSR, programmez la fonction qui calcule le couple du LFSR de plus petite longueur ℓ produisant le même flux de valeurs. En déduire la fonction qui prend en paramètre un LFSR et qui retourne le LFSR de longueur minimale produisant le même flux de valeurs.
6. Quels liens faîtes-vous avec le théorème de la division suivant les puissances croissantes ?

3 Application à la cryptographie

Les nombres aléatoires interviennent en cryptographie, notamment pour la création de clé de (dé)chiffrement. On devrait parler plutôt de nombres pseudo-aléatoires, produits par des générateurs qui, s'ils sont bien choisis vont simuler le hasard. Les LFSR sont des générateurs de 0 et de 1 dont l'efficacité en terme d'aléatoire est mesurable. Un "bon" LFSR de longueur ℓ assez grande, avec un état initial non nul, doit avoir un polynôme de rétroaction primitif. La suite $(r_i)_{i\geq 0}$ est de période maximale $2^\ell - 1$ et toute sous-suite apparaît avec la même fréquence.

▸ **Exercice 6.** Programmez la construction d'un "bon" LFSR de longueur ℓ.

Le chiffrement d'un **texte en clair** $t = t_0 t_1 \cdots t_k$ $(k \geq 0)$ à l'aide d'un "bon" LFSR de longueur ℓ telle que $2^\ell - 1 \leq k < 2^{\ell+1} - 1$ repose sur la relation $t_i \oplus r_i \oplus r_i = t_i$ si $r_i = 0$ ou $r_i = 1$. Le texte chiffré $c = c_0 c_1 \cdots c_k$ s'obtient en effectuant $c_i = t_i \oplus r_i$ $(0 \leq i \leq k)$. Le déchiffrement permet de retrouver le texte en clair en effectuant $t_i = c_i \oplus r_i$ $(0 \leq i \leq k)$. Le calcul des valeurs r_i $(0 \leq i \leq k)$ à l'aide du LFSR permet le chiffrement et le déchiffrement. Autant de valeurs que de bits dans le texte en clair sont nécessaires. À chaque nouveau chiffrement, de nouvelles valeurs de la suite $(r_i)_{i\geq 0}$ sont calculées. Le LFSR et son état initial doivent être connus par l'émetteur du texte en clair à chiffrer et par le destinataire qui doit le déchiffrer.

▸ **Exercice 7.** Programmez les fonctions de chiffrement et de déchiffrement. D'une utilisation à l'autre, prévoyez de conserver le dernier état connu du LFSR comme nouvel état initial pour éviter de retrouver la même suite de valeurs.

Une **attaque à texte chiffré** consiste à déchiffrer un message chiffré sans autre information que ce message lui-même. L'algorithme de Berlekamp-Massey la rend possible pour tout chiffrement avec un LFSR : il retourne le polynôme de rétroaction du LFSR de longueur minimale produisant la suite $(r_i)_{i\geq 0}$. Pour cela il suffit de connaître 2ℓ valeurs consécutives de la suite $(r_i)_{i\geq 0}$. Les étapes principales de l'algorithme de Berlekamp-Massey sont celles de l'algorithme d'Euclide étendu où, en suivant les notations de votre cours de Calcul Symbolique, les polynômes initiaux de $\mathbb{F}_2[X]$ sont :

$$R_0 = X^{2n} \quad R_1 = \textstyle\sum_{i=0}^{2n-1} r_i X^i \quad \alpha_0 = 1 \quad \beta_0 = 1 \quad \alpha_1 = 0 \quad \beta_1 = 0$$

Ici c'est le polynôme $\beta_m \in \mathbb{F}_2[X]$ qui nous intéresse, sachant que m est le premier entier positif strictement inférieur à n. Le polynôme de rétroaction recherché est alors $\mathrm{renv}_d(\beta_m)$ où $d = \sup(\deg(\beta_m), \deg(R_m) + 1)$.

▸ **Exercice 8.** Connaissant les premières valeurs d'une suite produite par un LFSR inconnu pour (dé)chiffrer un texte, réalisez une attaque en programmant le calcul du plus petit LFSR générant une suite commençant avec ces valeurs.

Une autre application de Berlekamp-Massey consiste à obtenir un "petit" générateur de clé : on génère une clé aléatoire assez grande puis on applique Berlekamp-Massey afin de trouver le plus petit LFSR qui la produit.