

IFT 3245

Simulation et modèles

Fabian Bastin
DIRO
Université de Montréal

Automne 2016

Récurrances linéaires dans \mathcal{F}_2

Considérons \mathcal{F}_2 , l'ensemble $\{0, 1\}$ sur lequel sont définies les opérations d'addition et de multiplication modulo 2.

En d'autres termes,

- l'addition revient à appliquer la porte logique XOR
- la multiplication revient à appliquer la porte logique AND

Réurrences linéaires dans \mathcal{F}_2

À partir du vecteur d'état \mathbf{x}_{n-1} , à l'étape $n - 1$, nous définissons comme précédemment la récurrence linéaire:

$$\mathbf{x}_n = X\mathbf{x}_{n-1}.$$

Toutefois, \mathbf{x}_n est exprimé à présent comme un vecteurs de k bits, de sorte que chaque composante se trouve bien dans \mathcal{F}_2 . Par exemple, $\mathbf{x}_n = (0, 1, 1, 0, 0, 1, 0, 1, 1, 0)$.

Le vecteur de sortie \mathbf{y}_n , défini sur w bits, est obtenu comme suit

$$\mathbf{y}_n = B\mathbf{x}_n.$$

Récurrentes linéaires dans \mathcal{F}_2

Il nous reste à définir la sortie, à savoir un réel compris dans l'intervalle $[0, 1)$:

$$u_n = \sum_{j=1}^w y_{n,j-1} 2^{-j} = .y_{n,0} y_{n,1} y_{n,2} \cdots$$

Chaque coordonnée de \mathbf{x}_n et de \mathbf{y}_n suit la récurrence linéaire

$$x_{n,j} = (\alpha_1 x_{n-1,j} + \cdots + \alpha_k x_{n-k,j}) \bmod 2,$$

où $\alpha_1, \alpha_2, \dots, \alpha_k \in \{0, 1\}$, ce qui permet de définir le polynôme caractéristique

$$P(z) = z^k - \alpha_1 z^{k-1} - \cdots - \alpha_{k-1} z - \alpha_k = \det(X - zI).$$

La période maximale $\rho = 2^k - 1$ est atteinte ssi $P(z)$ est primitif sur \mathcal{F}_2 (c'est-à-dire qu'il n'est pas factorisable en produit de polynômes).

Réurrences linéaires dans \mathcal{F}_2

De même que pour les MRG's, il est possible de sauter en avant dans la séquence ainsi produite, par blocs de i étapes, en procédant comme suit:

$$\mathbf{x}_{n+i} = \underbrace{(X^i \bmod 2)}_{\text{précalculer}} \mathbf{x}_n \bmod 2.$$

Générateur de Tausworthe (ou LFSR: linear feedback shift register)

De tels générateurs ont été introduits par Tausworthe, en 1965. Nous définissons une séquence x_1, x_2, \dots de chiffres binaires par la récurrence

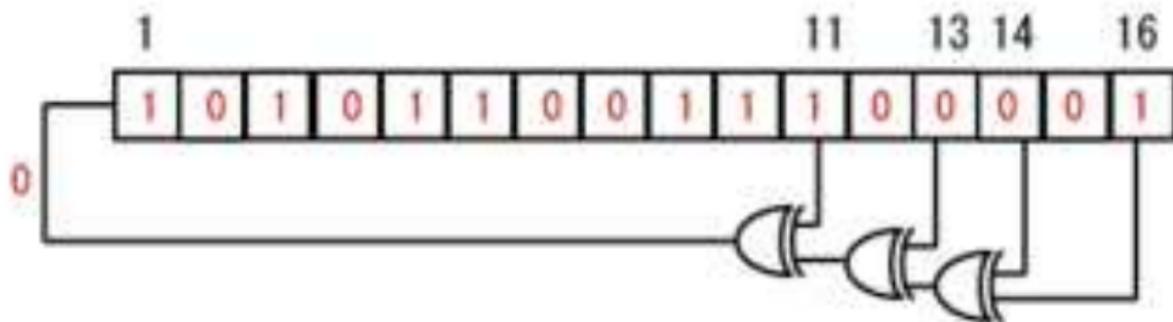
$$x_n = (a_1 x_{n-1} + \dots + a_k x_{n-k}) \pmod{2},$$

où $a_1, a_2, \dots, a_k \in \{0, 1\}$, et $a_k \neq 0$.

Le nom du générateur vient de la possibilité d'utilisation d'un registre de décalage (shift register), alimenté par une fonction de feedback linéaire, avec 16 bits. La fonction de sortie est définie comme

$$u_n = \sum_{l=1}^w x_{n\nu+l-1} 2^{-l} = .x_{n\nu} x_{n\nu+1} x_{n\nu+2} \dots x_{n\nu+w-1}.$$

Registre de décalage à feedback linéaire.



Cela revient à choisir

$$X = \begin{pmatrix} a_k & a_{k-1} & \dots & a_1 \\ 1 & & & 0 \\ & \ddots & & 0 \\ & & 1 & 0 \end{pmatrix}^{\nu} \text{ et } B = I.$$

Comme précédemment, la période maximale est $\rho = 2^k - 1$, laquelle est atteinte ssi $Q(z) = z^k - a_1 z^{k-1} - \dots - a_{k-1} z - a_k$ est primitif et le plus grand commun diviseur de ν et $2^k - 1$ vaut 1.

Dans la plupart des applications, seulement deux des coefficients sont non nuls pour simplifier l'implantation, ce qui donne le trinôme: $Q(z) = z^k - a_r z^{k-r} - a_k$.

Comme on travaille dans \mathcal{F}_2 , cela donne la récurrence

$$x_n = (x_{n-r} + x_{n-k}) \pmod{2}.$$

L'exécution de l'addition modulo 2 est équivalente à l'instruction ou-exclusif (xor) sur les bits:

$$x_n = \begin{cases} 0 & \text{si } x_{n-r} = x_{n-k}, \\ 1 & \text{si } x_{n-r} \neq x_{n-k}. \end{cases}$$

Plus généralement, on construit une implantation très rapide par des shifts, xors, masques, etc., si $\nu \leq r$ et $2r > k$.

Les générateurs LFSR sont connus pour avoir des déficiences statistiques. On peut cependant en améliorer les propriétés en considérant des LFSR combinés.

Generalized feedback shift register (GFSR)

Introduit par Lewis et Payne (1973), ce générateur se base sur la récurrence

$$\mathbf{v}_n = (a_1 \mathbf{v}_{n-1} + \cdots + a_r \mathbf{v}_{n-r}) \bmod 2 = (v_{n,0}, \dots, v_{n,w-1})^T,$$

et

$$\mathbf{y}_n = \mathbf{v}_n.$$

Generalized feedback shift register (GFSR)

Si $P(z)$ est un trinôme, ce qui est courant dans les implantations, nous avons

$$\mathbf{v}_n = (\mathbf{v}_{n+m-r} + \mathbf{v}_{n-r}) \pmod{2},$$

ce qui donne

$$X = \begin{pmatrix} & & & I_w & & I_w \\ & I_w & & & & \\ & & I_w & & & \\ & & & I_w & & \\ & & & & \ddots & \\ & & & & & I_w \end{pmatrix}$$

Generalized feedback shift register (GFSR)

Plus généralement, $P(z) = z^r - a_1 z^{r-1} - \dots - a_{r-1} z - a_r$ et la période maximale est $2^r - 1$ même si l'état a rw bits.

Ceci signifie que nous utilisons w copies de la récurrence du LFSR, avec des valeurs initiales différentes, et on utilise une copie pour chaque chiffre de l'expansion fractionnelle de u_n . Si $\{x_{j,n}\}$ désigne la j^{e} copie et si $x_{j,n} = x_{n+d_j}$ pour tout j, n ,

$$u_n = \sum_{j=1}^w x_{n+d_j} 2^{-j}.$$

Souvent, $d_j = (j - 1)d$, pour un certain d fixé

Ces générateurs, proposés par Matsumoto et Kurita 1992, 1994, généralisent les GFSR comme suit:

$$\begin{aligned}\mathbf{v}_n &= (\mathbf{v}_{n+m-r} + A\mathbf{v}_{n-r}) \pmod{2} \\ \mathbf{y}_n &= \mathbf{v}_n \text{ ou } \mathbf{y}_n = T\mathbf{v}_n, \\ X &= \begin{pmatrix} & & & I_w & & A \\ & & & & & \\ I_w & & & & & \\ & I_w & & & & \\ & & I_w & & & \\ & & & \ddots & & \\ & & & & I_w & \end{pmatrix}.\end{aligned}$$

La période maximale est $2^{rw} - 1$, atteinte ssi $Q(z^r + z^m)$ est primitif de degré rw , où Q est le polynôme caractéristique de A (i.e. $\det(XI - A)$). L'exemple le plus connu est le générateur TT800, qui atteint une période de $2^{800} - 1$.

Récurrance multiple matricielle

(Niederreiter 1995)

$$\mathbf{v}_n = A_1 \mathbf{v}_{n-1} + A_2 \mathbf{v}_{n-2} + \cdots + A_r \mathbf{v}_{n-r},$$

$$\mathbf{y}_n = \mathbf{v}_n,$$

$$X = \begin{pmatrix} A_1 & A_2 & \cdots & A_{r-1} & A_r \\ I_w & & & & \\ & I_w & & & \\ & & \ddots & & \\ & & & I_w & \end{pmatrix}.$$

La période maximale est $2^{rw} - 1$.

Mersenne Twister

La période peut être considérablement améliorée en considérant une récurrence de la forme

$$\begin{aligned}\mathbf{v}_n &= (\mathbf{v}_{n+m-r} + A(\mathbf{v}_{n-r}^u | \mathbf{v}_{n-r+1}^l)) \pmod 2, \\ \mathbf{y}_n &= T \mathbf{v}_n.\end{aligned}$$

L'exposant u signifie que nous prenons les $(w - p)$ bits de poids forts, et l , les p bits de poids faible. Cette relation est connue sous le nom Mersenne Twister, comme introduit par Matsumoto et Nishimura (1998).

La récurrence peut être réécrite

$$\mathbf{v}_n = \left(\mathbf{v}_{n+m-r} + A \begin{pmatrix} 0 & 0 \\ 0 & I_p \end{pmatrix} \mathbf{v}_{n-r+1} \begin{pmatrix} I_{w-p} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{v}_{n-r} \right) \pmod 2,$$

Nous pouvons exprimer X comme la matrice $(nw - p) \times (nw - p)$

$$\begin{pmatrix} & & & & I_w & & 0 \\ & A \text{ rot}_p(I) & & & & & \\ & I_w & & & & & \\ & & I_w & & & & \\ & & & I_w & & & \\ & & & & \ddots & & \\ & & & & & I_w & 0 & 0 \\ 0 & & & & & 0 & I_{w-p} & 0 \end{pmatrix} \cdot$$

en considérant dans le vecteur d'état les r vecteurs v_n, \dots, v_{n-r+2} ainsi que, de manière répétée, les p bits de poids fort de v_{n-r+1} .

$\text{rot}_p(I)$ est défini comme

$$\begin{pmatrix} 0 & I_{w-p} \\ I_p & 0 \end{pmatrix}.$$

La période maximale est $2^{rw-p} - 1$. Un exemple populaire est le générateur MT19937, dont la période de $2^{19937} - 1$.

Générateurs combinés sur \mathcal{F}_2

J générateurs \mathcal{F}_2 -linéaires de paramètres $(k_j, w, \mathbf{A}_j, \mathbf{B}_j)$ et états $\mathbf{x}_{j,i}$.

$$\mathbf{y}_n = \mathbf{B}_1 \mathbf{x}_{1,n} \oplus \cdots \oplus \mathbf{B}_J \mathbf{x}_{J,n},$$
$$u_n = \sum_{\ell=1}^w y_{n,\ell-1} 2^{-\ell},$$

Equivalent à un générateur \mathcal{F}_2 -linéaire ayant $k = k_1 + \cdots + k_J$, $\mathbf{A} = \text{diag}(\mathbf{A}_1, \dots, \mathbf{A}_J)$, et $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_J)$.

Générateurs combinés sur \mathcal{F}_2

Si on combine des LFSRs ayant des polynômes caractéristiques $P_j(z)$, le générateur combiné a comme polynôme caractéristique $P(z) = P_1(z) \cdots P_J(z)$ et sa période peut atteindre le produit des périodes.

En combinant des LFSR, TGFSR, ou Mersenne twister entre eux, on obtient des générateurs ayant de bien meilleures équidistributions.

```
unsigned long z1, z2, z3, z4;
```

```
double lfsr113 ()  
{ /* Generates numbers between 0 and 1. */  
  unsigned long b;  
  b = (((z1 << 6) ^ z1) >> 13);  
  z1 = (((z1 & 4294967294) << 18) ^ b);  
  b = (((z2 << 2) ^ z2) >> 27);  
  z2 = (((z2 & 4294967288) << 2) ^ b);  
  b = (((z3 << 13) ^ z3) >> 21);  
  z3 = (((z3 & 4294967280) << 7) ^ b);  
  b = (((z4 << 3) ^ z4) >> 12);  
  z4 = (((z4 & 4294967168) << 13) ^ b);  
  return ((z1^z2^z3^z4)*2.3283064365387e-10);  
}
```

Les opérations utilisées sont

- $\&$: opérateur et.
- \wedge : ou exclusif.
- \ll : décale à gauche; revient à multiplier par une puissance de 2.
- \gg : décale à droite; revient à diviser par une puissance de 2.