LFSR Linear Feedback Shift Register

An LFSR has maximal period if its associated/connection polynomial is primitive.

In that case if n is the length of the LFSR, the period will be $2^n - 1$

In [1]:
```
p:2
```
Out[1]: 2

In [2]:
```
n:4
```
Out[2]: 4

A poly of degree m over $\mathbb{Z}_p$ is primitive if its order is $p^m - 1$

Here $p = 2, n = 4$, therefore $p^m - 1 = 2^4 - 1 = 15$

In [3]:
```
f:gf_primitive_poly(p,n)
```
Out[3]: $x^4 + x + 1$

In [4]:
```
modulus:2
```
Out[4]: 2

In [5]:
```
gf_set_data(p,n)
```
Out[5]: $\text{Structure [GF-DATA]}$

In [6]:
```
gf_order(f(x))
```
Out[6]: 15

f(x) is a primitive polynomial so we can expect a period of $2^4 - 1 = 15$ from its LFSR

In [7]:
```
modulus:2
```
Out[7]: 2

In [8]:
```
seed:[0,1,0,1]
```
Out[8]: $[0, 1, 0, 1]$

This matrix does exactly what a LFSR does: shifts right and replaces first bit with the xor of the taps

In [9]:
```
mlfsr:matrix([1,1,0,0],
             [0,0,1,0],
             [0,0,0,1],
             [1,0,0,0])
```
Out[9]:
$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

In [10]:
```
expand(charpoly(mlfsr,lambda))
```
Out[10]: $\lambda^4 - \lambda^3 - 1$

In [11]:
```
gf_primitive_poly_p(%,p)
```
Out[11]: **true**

In [14]:
```
for i:1 thru p^n-1 do ( seed:seed . mlfsr, seed:mod(seed,2),print(i,seed) )
```
```
1 [ 0  1  1  0 ]
2 [ 0  0  1  1 ]
3 [ 1  0  0  1 ]
4 [ 0  1  0  0 ]
5 [ 0  0  1  0 ]
6 [ 0  0  0  1 ]
7 [ 1  0  0  0 ]
8 [ 1  1  0  0 ]
9 [ 1  1  1  0 ]
10 [ 1  1  1  1 ]
11 [ 0  1  1  1 ]
12 [ 1  0  1  1 ]
13 [ 0  1  0  1 ]
14 [ 1  0  1  0 ]
15 [ 1  1  0  1 ]
```
Out[14]: **done**

In [64]:
```
p:3
```
Out[64]: 3

In [64]:
```
n:3
```
Out[64]: 3

In [64]:
```
gf_primitive_poly(p,n)
```
Out[64]: $x^3 + 2\,x + 1$

In [64]:
```
seed:[0,2,1]
```
Out[64]: $[0, 2, 1]$

In [64]:
```
mlfsr:matrix([2,1,0],
             [0,0,1],
             [1,0,0])
```
Out[64]: $\begin{pmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

In [64]:
```
expand(charpoly(mlfsr,lambda))
```
Out[64]: $-\lambda^3 + 2\,\lambda^2 + 1$

In [64]:
```
gf_primitive_poly_p(%,p)
```
Out[64]: **false**

In [64]:
```
for i:1 thru p^n do ( seed:seed . mlfsr, seed:mod(seed,p),print(i,seed) )
```

```
1 [ 1  0  2 ]
2 [ 1  1  0 ]
3 [ 2  1  1 ]
4 [ 2  2  1 ]
5 [ 2  2  2 ]
6 [ 0  2  2 ]
7 [ 2  0  2 ]
8 [ 0  2  0 ]
9 [ 0  0  2 ]
10 [ 2  0  0 ]
11 [ 1  2  0 ]
12 [ 2  1  2 ]
13 [ 0  2  1 ]
14 [ 1  0  2 ]
15 [ 1  1  0 ]
16 [ 2  1  1 ]
17 [ 2  2  1 ]
18 [ 2  2  2 ]
19 [ 0  2  2 ]
20 [ 2  0  2 ]
21 [ 0  2  0 ]
22 [ 0  0  2 ]
23 [ 2  0  0 ]
24 [ 1  2  0 ]
25 [ 2  1  2 ]
26 [ 0  2  1 ]
27 [ 1  0  2 ]
```

Out[64]: **done**

In [ ]: