Codes non linéaires

Pour avoir des nouvelles fonctions pour ce cours, il faut charger le fichier MAT16.sage sur le site du cours.

```
MAT16 = "http://iml.univ-mrs.fr/~kohel/tch/MAT16/"
exec(open(get_remote_file(MAT16 + "MAT16.sage")).read())
```

1. Rappeler qu'un codage $C : X \to \mathcal{A}^n$ est *parfait* s'il existe un t tel que $\mathcal{A}^n$ est l'union des boules disjoints de rayon t autour de x dans $C(X)$:

$$\mathcal{A}^n = \bigcup_{x \in C(X)} B(x,t)$$

et

$$B(x,t) \cap B(y,t) = \emptyset$$

pour tout $x \neq y$ dans $C(X)$.

a. Construire les mots de code $\{000000, 001110, 110001, 111111\}$.

```
B = BinaryStrings()
u0 = B("000000"); u1 = B("001110")
u2 = B("110001"); u3 = B("111111")
C = [u0,u1,u2,u3]
```

b. Trouver les éléments des boules de rayon 2 autour de 000000, 001110, 110001, et 111111. Combien d'éléments est-ce qu'il y a dans chaque boule ?

```
B0 = Ball(u0,2); B1 = Ball(u1,2)
B2 = Ball(u2,2); B3 = Ball(u3,2)
print "B0:"; print B0
print "B1:"; print B1
```

c. Trouver les éléments des sphères de rayon 1 et 2 autour de 000000, 001110, 110001, et 111111. Combien d'éléments est-ce qu'il y a dans chaque sphère ?

```
S0 = Sphere(u0,2); T0 = Sphere(u0,1)
S1 = Sphere(u1,2); T1 = Sphere(u1,1)
print S0.intersection(S1)
print S0.intersection(T1)
print S1.intersection(T0)
```

d. Trouver les nombres d'éléments dans

$$\bigcup_{x\in C(X)} B(x,1),\ \bigcup_{x\in C(X)} B(x,2),\ \text{et}\ \bigcup_{x\in C(X)} B(x,3).$$

Par exemple, utiliser :

```
for r in range(3):
    X = Set([])
    for v in C:
        X = X.union(Ball(v,r))
    print "|B(C,%s)|: %s" % (r, len(X))
```

Est-ce que c'est un codage parfait ?

2. Construire l'ensemble des mots de code $C(X) = \{000000, 110011, 111100, 001111\}$ et vérifier que ce n'est pas un codage parfait.

3. Construire l'ensemble des mots de code

$$C(X) = \left\{ \begin{array}{cccc} 0000000, & 0101110, & 1000110, & 1011100, \\ 0100011, & 0010111, & 1100101, & 1111111, \\ 0110100, & 0011010, & 1110010, & 1101000, \\ 0111001, & 0001101, & 1010001, & 1001011 \end{array} \right\}$$

et vérifier qu'il est parfait.

Corps finis

Dans cette partie, on souhaite retrouver sur des exemples certaines propriétés des corps finis. Les corps finis les plus simples sont les $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ où p est un nombre premier. On peut les définir en Sage par la commande

```
F=GF(p)
```

1. Créer le corps $F = \mathbb{F}_{31}$.

2. Vérifier que $\#F = 31$

3. Pour chaque élément non nul de F, vérifier qu'il possède un inverse.

Une fois définis ces corps, on peut bien sûr définir des anneaux de polynômes $\mathbb{F}_p[X]$:

```
A.<x>=PolynomialRing(F)
```

4. Définir l'anneau $A = \mathbb{F}_{31}[x]$.

5. Définir le polynôme $P = x^2 + 2x + 7$.

6. Factoriser P dans A.

7. Trouver un polynôme de degré 2 irréductible.

8. Vérifier que $x^{30} - 1 = \prod_{a \in F^*}(x - a)$.

Si P est un polynôme irréductible de degré n sur un corps fini $\mathbb{F}_p$ alors l'anneau $\mathbb{F}_p[x]/(P)$ est un corps et on peut calculer un inverse par l'algorithme d'Euclide étendu.

9. Calculer l'inverse de $x + 3$ dans A à la main.

Le théorème suivant affirme que pour tout n on peut trouver un tel polynôme et que l'extension de $\mathbb{F}_p$ ainsi définie est unique.

Théorème 1 *Soit p un nombre premier et n un entier. Il existe à isomorphisme près un unique corps de cardinal p^n, noté $\mathbb{F}_{p^n}$.*

Sage note ces corps

```
F=GF(p^n,'a')
```

où a est un élément tel que $1, a, \ldots, a^{n-1}$ est une base du $\mathbb{F}_p$-espace vectoriel F.

10. Factoriser le polynôme $x^2 + 2x + 2$ sur $F_2 = \mathbb{F}_{31^2}$.

Théorème 2 *Le groupe multiplicatif d'un corps fini est cyclique. Un élément générateur est appelé un élément primitif.*

11. Trouver un élément primitif de F.