tp_Berlekamp(1)

November 4, 2019

1 Corps finis

1.1 I- Définition et éléments

En Sage, pour définir un corps fini du type $\mathbf{Z}/p\mathbf{Z}$, avec p premier, on dispose de la commande `IntegerModRing`.

```
In [1]: A=IntegerModRing(13); A
```

```
Out[1]: Ring of integers modulo 13
```

En donnant un nom à l'anneau, on peut alors utiliser l'application quotient canonique $\mathbf{Z} \to \mathbf{Z}/p\mathbf{Z}$.

```
In [2]: A(54)
```

```
Out[2]: 2
```

Les "modèles" pour les corps finis à p^n éléments sont des quotients de l'anneau de polynômes $\mathbf{Z}/p\mathbf{Z}[X]$. En Sage, on ne peut pas introduire une nouvelle indéterminée sans la définir.

```
In [3]: 3*X+5
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)

<ipython-input-3-32f7bd0dc849> in <module>()
----> 1 Integer(3)*X+Integer(5)


NameError: name 'X' is not defined
```

Pour définir l'anneau de polynômes $\mathbf{Z}/p\mathbf{Z}[X]$, on procède comme suit:

```
In [4]: R.<X> = PolynomialRing(A);R
```

```
Out[4]: Univariate Polynomial Ring in X over Ring of integers modulo 13
```

Ou bien comme cela:

```
In [5]: R1 = PolynomialRing(A,'X'); R1

Out[5]: Univariate Polynomial Ring in X over Ring of integers modulo 13

In [6]: 3*X+5

Out[6]: 3*X + 5
```

On dispose des fonctions usuelles pour l'arithmétique des polynômes: `%`, `//`, `gcd`, `xgcd`.

```
In [7]: (3*X^5+5)%(5*X^2+5)

Out[7]: 3*X + 5
```

Normalement, vous devriez pouvoir les recoder en quelques lignes et minutes...

```
In [8]: def pgcd(P,Q):
            return 1
```

On peut alors définir le quotient de R par un idéal (quelconque).

```
In [9]: P=2*X^2+3*X+1

In [10]: RR=QuotientRing(R,ideal(P)); RR

Out[10]: Univariate Quotient Polynomial Ring in Xbar over Ring of integers modulo 13 with modul

In [11]: RR(5*X^6+4*X^5+5)^-1

Out[11]: Xbar + 12
```

Noter que la construction précédente est très générale: P n'est pas supposé irréductible!

Vérifiez que vous êtes capables de coder vous mêmes la fonction qui teste si un élément est inversible dans RR et le cas échéant en renvoie son inverse.

Au fait, le P précédent est-il irréductible?

```
In [12]: def inverse_dans_RR(Q):
             return 0
```

En Sage, il existe un raccourci pour définir un corps fini à p^n éléments, il s'agit de la fonction `GF()` (comme Galois Field).

```
In [13]: R2=GF(8); R2

Out[13]: Finite Field in z3 of size 2^3
```

Sage a défini notre corps à 8 éléments comme un quotient de $\mathbf{Z}/2\mathbf{Z}[x]$ par un certain polynôme irréductible de degré 3 qu'il a choisi tout seul (avantage ou inconvénient?). Notez qu'il a aussi choisi tout seul le nom du générateur...

On peut accéder à ce polynôme et aux éléments:

```
In [14]: R2.modulus()

Out[14]: x^3 + x + 1

In [15]: R2.gen()

Out[15]: z3
```

Ici, `gen` **veut bien entendu dire générateur. En quel sens** z_3 **est-il un générateur?**

```
In [16]: for i in R2:
             print (i)

0
z3
z3^2
z3 + 1
z3^2 + z3
z3^2 + z3 + 1
z3^2 + 1
1
```

On peut bien spécifier le nom du générateur et le polynôme:

```
In [17]: R3.<y>=GF(13^4,modulus=x^4 + 3*x^2 + 12*x + 1); R3

Out[17]: Finite Field in y of size 13^4

In [18]: R3.modulus()

Out[18]: x^4 + 3*x^2 + 12*x + 1
```

Il est utile d'avoir accès aux fonctions suivantes:

```
In [19]: R3.cardinality()

Out[19]: 28561

In [20]: z=R3.multiplicative_generator(); z

Out[20]: 8*y^3 + 12*y^2 + 9*y + 9

In [21]: z.minimal_polynomial()

Out[21]: x^4 + 12*x^3 + x + 11
```

```
In [22]: y.multiplicative_order()

Out[22]: 595

In [23]: z1=z.polynomial(); z1.coefficients()

Out[23]: [9, 9, 12, 8]
```

Il y a aussi une fonction renvoyant l'ordre additif d'un élément. Qu'en pensez-vous?
Codez vos propres fonctions `ordre` **et** `generateur` **pour R3.**

```
In [24]: def ordre(a):
             return 0

In [25]: def generateur():
             return 0
```

Coder votre fonction `pol_min` en utilisant les fonctions d'algèbre linéaire suivantes pour définir des matrices, en calculer le rang et le noyau.

```
In [26]: M=Matrix(A,3,3,[i for i in range(9)]); M

Out[26]: [0 1 2]
         [3 4 5]
         [6 7 8]

In [27]: M.rank()

Out[27]: 2

In [28]: V=M.right_kernel(); V

Out[28]: Vector space of degree 3 and dimension 1 over Ring of integers modulo 13
         Basis matrix:
         [ 1 11  1]

In [29]: V.gen()

Out[29]: (1, 11, 1)

In [ ]:

In [ ]:
```

1.2 II- Factorisation de polynômes et algorithme de Berlekamp

Sage sait factoriser les polynômes à coefficients dans un corps fini, grâce à la commande `factor`.

```
In [30]: R1.<X>=PolynomialRing(Zmod(7)); R1
```

```
Out[30]: Univariate Polynomial Ring in X over Ring of integers modulo 7
```

```
In [31]: P=X^6 + X^4 + 5*X^3 + 4*X^2 + 6*X + 3
```

```
In [32]: factor(P)
```

```
Out[32]: X^6 + X^4 + 5*X^3 + 4*X^2 + 6*X + 3
```

Qu'en déduit-on sur P?

```
In [33]: K1.<X>=PolynomialRing(GF(7^6));
```

```
In [34]: factor(K1(P))
```

```
Out[34]: (X + 6*z6) * (X + 2*z6^4 + 2*z6^3 + 3*z6^2 + 4*z6 + 6) * (X + z6^5 + 5*z6^4 + 4*z6^3 +
```

Était-ce prévisible?

```
In [35]: K2.<X>=PolynomialRing(GF(7^3));
```

```
In [36]: factor(K2(P))
```

```
Out[36]: (X^2 + (z3^2 + z3 + 4)*X + 4*z3^2 + z3 + 1) * (X^2 + (2*z3^2 + 2*z3 + 1)*X + 3*z3^2 +
```

Était-il prévisible d'avoir 3 facteurs de degré 2?

1.2.1 Exercice

a) Montrer que le polynôme $P = X^3 + 3X + 3 \in \mathbf{F}_5[X]$ est sans facteur carré.

b) Utiliser l'algorithme de Berlekamp pour montrer que P est irréductible.

c) Utiliser cet algorithme pour factoriser les polynômes suivants dans $(\mathbf{F}_5)[X]$: $P_1 := X^5 + 3X^2 + 3X + 1$ et $P_2 = X^6 + 2X^5 + 3X^3 + 4X^2 + 2X + 2$.

d) Ecrire une fonction `Berlekamp(p,P)` qui factorise le polynôme P dans $\mathbf{F}_p[X]$.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

1.3 Sous-corps

Exercice : définir en Sage un corps K à 5^4 éléments.

A partir de quel polynôme irréductible est-il construit?

Déterminer un sous-corps K_1 à 5^2 éléments de K: - faire la liste des éléments de K_1, - trouver un générateur z de K_1 comme algèbre - déterminer le polynôme minimal d'un générateur x de K sur K_1. *(Pour cela, on pourra déterminer a priori le degré d de ce polynôme, et utiliser une $\mathbf{F}_p$-base de K de la forme $x^i z^j$ pour trouver une relation linéaire adéquate.)*

```
In [ ]:
```

```
In [ ]:
```