TP6 : Factorisation sur $F_p[X]$

Référence : *Cours d'algèbre*, M. Demazure.

L'algorithme de Berlekamp est une méthode de factorisation des polynômes à coefficients dans un corps fini, qui repose sur des calculs de PGCD et des opérations matricielles. Le polynôme de départ doit être sans facteur carré, c'est-à-dire non divisible par le carré d'un polynôme non constant ; les facteurs obtenus ne sont pas a priori irréductibles.

Toutefois, si nous savons calculer la *partie sans facteur carré* d'un polynôme P, alors, en itérant l'algorithme de Berlekamp, nous pouvons écrire un algorithme plus fort, capable de factoriser n'importe quel polynôme en produit de polynômes irréductibles. L'objectif de ce TP en est justement l'implémentation.

Il commence par une partie préliminaire, où sont recensées les questions techniques qui surgiront plus tard. Puis l'on construit et implémente `PSFC(P)` qui renvoie la partie sans facteur carré de P. La démonstration et la mise en oeuvre de l'algorithme de Berlekamp dans $F_p[X]$ font l'objet de la section suivante. Enfin, on est en mesure d'écrire l'algorithme de factorisation final `FactorisationViaBerlekamp(P)`, et de discuter ses performances.

1 Préliminaires

1. Sur SAGE, construire le corps `F`$= \mathbb{Z}/7\mathbb{Z}$ et l'anneau `Pol` des polynômes à une indéterminée sur ce corps. A l'aide de `factor`, factoriser le polynôme $X^5 + 3X^2 + 1$.
2. Dresser la liste des coefficients d'un polynôme de $F_7[X]$. Réciproquement, étant donnée une liste d'élements de F, reconstruire le polynôme correspondant.
3. Rappeler ce qu'est le morphisme de Frobenius.
4. Grâce aux fonctions `rank(M)`, `kernel(M)`, `.basis()`, déterminer le rang et une base du noyau de la matrice :
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 6 & 1 & 1 & 4 \\ 1 & 0 & 1 & 3 \\ 3 & 0 & 3 & 2 \end{pmatrix},$$
vue comme matrice à coefficients dans $\mathbb{Z}$, puis dans F_7.

2 Partie sans facteur carré

Etant donné un polynôme unitaire $P = \prod_{i=1}^{r} P_i^{a_i}$, où les P_i sont irréductibles, unitaires et différents deux à deux, on appelle *partie sans facteur carré* de P le polynôme $\prod_{i=1}^{r} P_i$.

5. Ecrire et justifier un mini-algorithme (une ligne...) pour déterminer la partie sans facteur carré de P dans le cas où $P \in \mathbb{Q}[X]$. Le tester pour $X^2(X+1)^7$.

On se place désormais, et jusqu'à la fin du TP, sur le corps fini $F_p = \mathbb{Z}/p\mathbb{Z}$, où p est un nombre premier.

6. Tester votre algorithme précédent sur le polynôme $X^2(X+1)^7 \in F_7[X]$. D'où provient le problème ?

Nous allons donc aménager cet algorithme afin d'exhiber la partie sans facteur carré de P sur un corps de caractéristique quelconque.

7. Justifier que l'on peut toujours écrire $P(X) = Q(X)R(X)^p$, avec $Q = \prod_{i=1}^{s} Q_i^{b_i}$, où $b_i \not\equiv 0 \bmod(p)$, Q et R unitaires et premiers entre eux.
8. Exprimer P' en fonction de Q' et R ; puis $pgcd(P, P')$ en fonction de $pgcd(Q, Q')$ et de R. En déduire que :
$$\frac{P}{pgcd(P, P')} = \prod_{i=1}^{s} Q_i \ ,$$
où $Q = \prod_{i=1}^{s} Q_i^{b_i}$, avec $b_i \not\equiv 0 \bmod(p)$.
9. Ecrire un algorithme qui, connaissant P et $\prod_{i=1}^{s} Q_i$, renvoie $R(X)^p$.
10. Justifier que $R(X)^p = R(X^p)$. En déduire un algorithme qui, connaissant $R(X)^p$, renvoie $R(X)$.
11. Ecrire et implémenter `QR(P)` qui renvoie $\text{psfc}(Q) * R$.
12. Justifier que $P = QR(P)$ si et seulement si P est sans facteur carré. En déduire l'algorithme `PSFC(P)`. Pourquoi termine-t-il ? Tester votre fonction pour $P_1 = X(X+1)^3(X+2)^7, P_2 = X^{14}(X+1), P_3 = X(X(X+3)^3)^{21}, P_4 = X^{49} \in F_7[X]$.

3 Algorithme de Berlekamp

Principe

Soit $P \in F_p[X]$ un polynôme unitaire de degré n, sans facteur carré. On peut l'écrire $P = P_1 ... P_r$, où les P_i sont irréductibles et deux à deux premiers entre eux.

Chacun des quotients $K_i := F_p[X]/(P_i)$ est donc un corps*, et leur produit est isomorphe* à l'anneau $A := F_p[X]/(P)$.

On s'intéresse à l'équation $a^p = a$. Celle-ci admet exactement p solutions* dans chaque K_i, qui sont les éléments du sous-corps premier F_p. En vertu de notre isomorphisme, on obtient p^r solutions* dans l'anneau A ; chaque solution a pour image un r-uplet $(a_1, ..., a_r) \in F_p^r$.

Mais A, en plus d'être un anneau, dispose d'une structure d'espace vectoriel* sur F_p ; et l'application de Frobenius $\phi : A \to A, Q \mapsto Q^p$ peut alors être vue comme un endomorphisme* d'espace vectoriel. Ainsi, par ce qui précède, pour tout polynôme $Q \in N := ker(\phi - Id)$, il existe $(a_1, ..., a_r) \in F_p^r$ tel que $\forall i$, $Q \equiv a_i \bmod P_i$.

Soit à présent $a \in F_p$ et considérons le polynôme $Q - a$. Remarquons que P_i divise $Q - a$ si et seulement si $Q - a \equiv 0 \bmod P_i$ si et seulement si $a_i = a$. On en déduit* que $pgcd(P, Q - a) = \prod_{i/a_i=a} P_i$. D'où :
$$P = \prod_{i=1}^{r} P_i = \prod_{a \in F_p} \prod_{i/a_i=a} P_i = \prod_{a \in F_p} pgcd(P, Q - a).$$
Cette décomposition n'est pas triviale dès lors que le polynôme Q n'est pas constant*.

Questions de compréhension

13. Pourquoi les K_i sont-ils des corps ? Pourquoi leur produit est-il isomorphe à A ? Expliciter l'isomorphisme.
14. Justifier que l'équation $a^p = a$ a exactement p solutions dans chaque K_i.
15. N'est-il pas étonnant que le polynôme $X^p - X \in A[X]$ ait a priori plus de p racines ?
16. Quelle est la dimension du F_p-ev A ?
17. Justifier que l'application de Frobenius ϕ est un endomorphisme de l'espace vectoriel A.
18. Soit $Q \in N$ et $a \in F_p$. Démontrer que $pgcd(P, Q - a) = \prod_{i/a_i=a} P_i$.
19. Prouver que la décomposition est triviale (ie : l'un des facteurs est associé à P, les autres sont constants) si et seulement si le polynôme Q est constant.

Sur un exemple simple...

Dans $F_3[X]$, on considère le polynôme sans facteur carré $P = X(X+1)$.

20. Déterminer une base de l'espace vectoriel A.
21. Expliciter dans cette base la matrice M de l'application de Frobenius.
22. Calculer la dimension de $ker(M-I)$. Montrer que, dans le cas général, $dim(ker(M-I)) \geq 1$? Quand est-ce que cette inégalité est stricte ?
23. Déterminer un vecteur de $ker(M-I)$ qui ne représente pas un polynôme constant de A.
24. Déterminer pour quels $a \in F_3$ on a $pgcd(Q-a, P) \neq 1$. En déduire la factorisation de P.

Description de l'algorithme

— Ecrire la matrice M de ϕ.
— Calculer son rang. Si $rang(M-I) = n-1$, alors $ker(\phi - id) = F_p$ et P est irréductible.
— Sinon, choisir un élément $Q \in ker(M-I)$ non constant.
— Trouver alors tous les $a \in F_p$ tels que $pgcd(Q-a, P) \neq 1$.
— Renvoyer la liste de ces pgcd.

Implémentation

Implémenter soigneusement l'algorithme, en faisant apparaître si besoin les résultats intermédiaires.

4 Algorithme de factorisation final

25. A partir de `PSFC(P)` et `Berlekamp(P)`, écrire un algorithme `FactorisationViaBerlekamp(P)` qui, étant donné P un polynôme quelconque sur F_p, renvoie la liste de ses facteurs **irréductibles**, accompagnés de leurs multiplicités.
    Ex : `FactorisationViaBerlekamp(X*X*(X+1)) = [(X,2),(X+1,1)]`
    Bravo, vous pouvez maintenant profiter de votre algorithme de factorisation !
26. Estimer la complexité de cet algorithme. Quelle est la complexité de l'algorithme naïf, qui consiste à lister tous les polynômes de degrés inférieurs à $deg(P)$ et tester la divisibilité ?
27. La démarche et les preuves s'adaptent-elles pour F_q un corps fini quelconque de caractéristique p ? Tester vos algorithmes sur F_{121}.
    Morale : Contrairement aux entiers, nous disposons pour les polynômes (sur les corps finis...) d'algorithmes de factorisation efficaces !

♠ Pour aller plus loin : factorisation d'un polynôme sur $\mathbb{Q}[X]$

Connaissant une décomposition du polynôme projeté sur $F_p[X]$, il est possible de remonter vers une décomposition sur $\mathbb{Z}$ via le théorème des restes chinois et le lemme de Hensel. Le lemme de Gauss permet alors d'en déduire une décomposition dans $\mathbb{Q}[X]$.