some open dense subset of the affine t-line. Instead of studying the fibers separately as in the second method, the aim of the deformation is to describe how the action of Frobenius on $H^1_{MW}(\overline{C}(x,y,t)/\mathbb{Q}_q)$ alters as t varies. The crucial point is that the matrices of Frobenius on $H^1_{MW}(\overline{C}(x,y,t)/\mathbb{Q}_q)$ satisfy a differential equation, which expresses the quasi-commutativity of the so called Gauss-Manin connection with the Frobenius action. The resulting algorithm thus proceeds in two stages: first compute the matrix of Frobenius of some fiber, typically $t = 0$ by a type two method and then solve the differential equation to obtain the matrix of Frobenius of the fiber one is interested in. The complexity of this algorithm for a family defined over $\mathbb{F}_{p^a}$ and fibers over $\mathbb{F}_{p^{an}}$ is $\tilde{O}(n^{2.667}g^{6.376}a^3)$. The main use of this algorithm is to search for curves with $|J_C(\mathbb{F}_q)|$ divisible by a large prime by choosing a family $\overline{C}(x,y,t)$ over $\mathbb{F}_p$ and fibers over $\mathbb{F}_{p^n}$. After a precomputation that depends only on the family, computing the zeta function of each new fiber is very efficient.

Algorithms in Computer Algebra Systems

The most complete computer algebra system with respect to point counting algorithms is Magma: it includes all algorithms including most optimizations described in this article. PARI/GP contains an implementation of Schoof's algorithm and freely available code for Satoh's algorithm exists. SAGE contains a native implementation of Kedlaya's algorithm, including Harvey's optimization.

Literatur

[1] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren (Eds.) *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

Neues über Systeme

Galoisgruppen in Magma

Claus Fieker
University of Sydney

Jürgen Klüners
Heinrich-Heine-Universität Düsseldorf

claus@maths.usyd.edu.au
klueners@math.uni-duesseldorf.de

Die Galoistheorie ist eines der klassischen Gebiete der Mathematik. So konnte vor mehreren hundert Jahren gezeigt werden, dass alle Polynome bis zum Grad 4 durch Radikale (Wurzelausdrücke) gelöst werden können. Durch die Galoistheorie wird jeder Gleichung eine Permutationsgruppe, die sogenannte Galoisgruppe, zugeordnet. Eine Gleichung ist genau dann durch Radikale lösbar, wenn ihre Galoisgruppe auflösbar ist. In diesem Bericht wollen wir einen Überblick über das Berechnen von Galoisgruppen in Magma (`http://magma.maths.usyd.edu.au/magma/`) geben.

Algorithmisch ist die Berechnung ein interessantes Problem. Jeder Zuhörer einer Algebra-Vorlesung hat sicherlich schon mühsam per Hand die Galoisgruppe eines kleinen Polynoms berechnet bzw. gezielt erraten. Obwohl die ursprünglichen Beweise „fast" konstruktiv sind, sind sie selbst auf einem modernen Computer nicht praktikabel. Pakete zur Berechnung von Galoisgruppen sind daher schon seit vielen Jahren in Computeralgebrasystemen wie Maple, Magma, Kash oder Gap enthalten. Allerdings gab es immer Gradschranken, die im Laufe der Jahre von Grad 7 oder 8 je nach Paket bis auf Grad 23 angewachsen sind.

Ab der Version 2.13 von Magma ist es nun gradunabhängig möglich, Galoisgruppen von Polynomen über den rationalen Zahlen zu bestimmen, wobei es egal ist, ob das Polynom irreduzibel ist oder nicht. In der aktuellen Magma-Version (2.14) sind diese Algorithmen auch für Polynome über Zahlkörpern oder über dem rationalen Funktionenkörper $\mathbb{Q}(t)$ anwendbar.

Wir verwenden eine Variante des klassischen Ansatzes von Richard Stauduhar mit der wesentlichen Neuerung, dass alle benötigten Daten (Gruppentheorie, Invariantentheorie) zur Laufzeit berechnet werden und nicht wie sonst üblich Tabellen entnommen werden. Der Befehl *GaloisGroup* berechnet in allen Fällen

die Galoisgruppe als explizite Gruppe von Permutationen der Nullstellen des Polynomes in einem geeigneten p-adischen Körper (Potenzreihenkörper über einem p-adischen Körper für $\mathbb{Q}(t)$). Die explizite Operation auf den Nullstellen kann dann auch für weitere Berechnungen genutzt werden. So können wir zum Beispiel entscheiden, ob ein polynomieller Ausdruck in den Nullstellen verschwindet oder nicht. Weiterhin enthält Magma die Funktion *GaloisSubgroup*, welche einen Fixkörper, d. h. eine Gleichung für einen beliebigen Teilkörper des Zerfällungskörpers berechnet.

Mit Hilfe der Funktion *GaloisQuotient* können wir Polynome konstruieren, die zu anderen Permutationsdarstellungen derselben Gruppe korrespondieren: Die alternierende Gruppe auf 5 Punkten, $A(5)$, kann auch als Untergruppe der symmetrischen Gruppe auf 6 Punkten realisiert werden. Wir können nun ausgehend von einem Polynom von Grad 5 mit Galoisgruppe $A(5)$ ein Polynom vom Grad 6 mit isomorpher Galoisgruppe konstruieren. Als letzte Anwendung wollen wir noch die Auflösbarkeit durch Radikale erwähnen, welche im auflösbaren Fall mit Hilfe der Funktion *SolveByRadicals* berechnet werden kann.

Um weitere Verbesserungen des Verfahrens bzw. Anwendungen zu ermöglichen, sind viele der für die Berechnung von Galoisgruppen notwendigen Teilschritte separat für Benutzer zugreifbar. So sind zum Beispiel alle Routinen, die invariante Polynome für (maximale) Untergruppen finden, verfügbar. Darauf aufbauend kann dann ein Stauduhar–Schritt für ein beliebiges Gruppenpaar aufgerufen werden.

Wir merken an, dass der erstmalig implementierte Algorithmus für reduzible Polynome dafür verwendet werden kann, um zu testen, ob die Zerfällungskörper zweier Polynome linear disjunkt sind.

Mit der vorgestellen Implementierung wurden Galoisgruppen von Polynomen vom Grad > 120 erfolgreich berechnet. Je nach dem Grad variiert die Laufzeit von wenigen Sekunden (Grad < 10) bis zu mehreren Stunden (Grad > 100). Die tatsächliche Laufzeit ist aber nicht nur vom Grad, sondern auch von den zu berechnenden Galoisgruppen abhängig und kann daher selbst bei fixiertem Grad stark variieren.

Wir illustrieren einige der Funktionen an einem Beispiel. Ausgangspunkt ist das Polynom $x^4+3x^3-4x^2-11x+4$, welches die alternierende Gruppe A_4 als Galoisgruppe hat. Wir berechnen zunächst die Galoisgruppe und dann den Fixkörper des Zerfällungskörpers (ohne diesen zu berechnen) unter einer Untergruppe der Ordnung 2. Dieser wird von einem Polynom vom Grad 6 erzeugt, welches wir dann durch Radikale auflösen. Die Magma-Ausgaben sind an verschiedenen Stellen gekürzt. Wir merken an, dass die Nullstellen von f in der unverzweigten Erweiterung vom Grad 3 über den 7-adischen Zahlen approximiert werden. Wir sehen, dass die Nullstellen des Polynoms g als Polynome in $a_1, \ldots, a_4$ dargestellt sind, wobei die a_i selber Radikalgleichungen erfüllen.

```
> Zx<x> := PolynomialRing(Integers());
> f := x^4 + 3*x^3 - 4*x^2 - 11*x + 4;
> G, R := GaloisGroup(f);
> G;
Permutation group G acting on a set of cardinality 4
Order = 12 = 2^2 * 3
    (1, 2)(3, 4)
    (1, 2, 3)
> TransitiveGroupDescription(G);
A(4)
> R;
[ -2883 + O(7^5), 840*$.1^2 - 6136*$.1 - 4585 + O(7^5), ... ]
> Universe(R);
Unramified extension defined by the polynomial
  (1 + O(7^20))*x^3 ... over 7-adic ring
> s := Subgroups(G:OrderEqual := 2);
> g := GaloisSubgroup(f, s[1]`subgroup);
> g;
x^6 - 17*x^5 - 3*x^4 + 1137*x^3 - 2726*x^2 - 18984*x + 53512
> TransitiveGroupDescription(GaloisGroup(g));
A_4(6) = [2^2]3
> K, R := SolveByRadicals(g:Name := "a");
> K:Maximal;
...
K  : a1^2 + 1/3912*(3*a4 - 25)*a3^2 + 1/6*a3 - 59/12
$1 : a2^2 + 1/1956*(-7*a4 + 4)*a3^2 + 1/12*(-a4 - 1)*a3 - 59/12
$2 : a3^3 + 1956*a4 + 16300
%$3 : a4^2 + 3
> R;
[ a1 + 1/1956*(-7*a4 + 4)*a3^2 + 1/12*(-a4 - 1)*a3 + 17/6, ...    ]
> Evaluate(g, R[1]);
0
```