Interval arithmetic for function fields over finite fields (or, How to compute in $\mathbb{C}_p$ without really trying)

Computational Algebraic and Analytic Geometry for Low-dimensional Varieties
January 17, 2003

Kiran S. Kedlaya
University of California, Berkeley
kedlaya@math.berkeley.edu

A preprint is in progress; an older preprint with some of these results is available at `arxiv.org` as math.RA/0110089. These slides are available at `math.berkeley.edu/~kedlaya`.

The author is supported by an NSF postdoctoral fellowship.

The geometric question

Give local parametrizations of plane curves over a field K, i.e., find approximate roots of polynomials over $K[t]$. Example: if char(K) $\neq 2$,

$$x^3 - xt + t^3 = 0$$

has a parametrization at the origin:

$$x = t^{1/2} - \frac{1}{2}t^2 - \frac{3}{8}t^{7/2} - \frac{1}{2}t^5 + \cdots$$

If $K = \mathbb{C}$, an iteration using Newton polygons produces roots in the ring of Puiseux series

$$\bigcup_{i=1}^{\infty} \mathbb{C}((t^{1/i}));$$

one can compute with approximations to these.

But this is false if char(K) > 0, e.g., for finite fields!

A bad example in positive characteristic

Chevalley observed that if char(K) $= p > 0$, the polynomial

$$x^p - x - t^{-1}$$

over $K((t))$ has no roots in the ring of Puiseux series over K.

Abhyankar suggested it should have the roots

$$x = c + t^{-1/p} + t^{-1/p^2} + \cdots \quad (c \in \mathbb{F}_p);$$

this makes sense in a ring of “generalized power series”.

Is it possible to make sense of this remark in a “computable” fashion?

Reformulation

The field $\mathbb{C}$ is complete and algebraically closed, and it is easy to compute in $\mathbb{C}$ using floating-point approximations (and interval arithmetic).

The fields $\mathbb{Q}_p$ and $\mathbb{F}_p((t))$ are easy to compute in using rational approximations, but they are not algebraically closed.

Question: how to compute in their completed algebraic closures? Is there a reasonable analogue of "floating-point arithmetic"?

Generalized power series (after Hahn)

The field $k((t^{\mathbb{Q}}))$ of generalized power series over a field k is the set of expressions

$$\sum_{i \in \mathbb{Q}} c_i t^i,$$

where $c_i \in k$ and the set of i such that $c_i \neq 0$ is *well-ordered*, i.e., contains no infinite decreasing sequence. (Well-orderedness is needed for series multiplication to work.)

If k is perfect, then $\bigcup K((t^{\mathbb{Q}}))$ is algebraically closed, where K runs over all finite extensions of k.

Unfortunately, the truncation of a general series modulo t^i is not described by computable data. In earlier work, we gave a "recursive" characterization of the power series in $k((t^{\mathbb{Q}}))$ which are algebraic over $k((t))$.

Finite automata

A finite automaton is an object which produces a collection of strings using symbols from a given alphabet Σ.

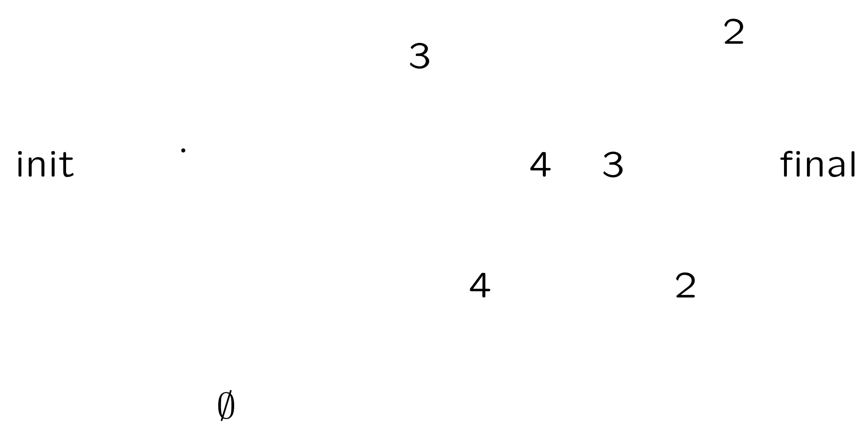
The data of an automaton includes:

- a finite collection Q of states;
- a *transition function* $F : Q \times \Sigma \to Q$;
- a designation of one state as the *initial state* and one or more states as *final states*.

The *language* generated by the automaton consists of all strings which yield a series of transitions from the initial state to some final state.

An example

For $\Sigma = \{., 0, 1, 2, 3, 4\}$, the automaton



with all unspecified transitions leading to $\emptyset$, accepts the language consisting of

$$.32, .342, .3432, .34342, \ldots$$

and

$$.42, .432, .4342, .43432, \ldots.$$

“Automatic” power series

Consider finite automata for the alphabet

$$\{., 0, \ldots, p-1\}.$$

A generalized power series $\sum c_i t^i$ over $\mathbb{F}_q$ (for $\mathrm{char}(\mathbb{F}_q) = p$) is called *automatic* if for each $\alpha \in \mathbb{F}_q \setminus \{0\}$, the set of $i \in \mathbb{Q}$ with $c_i = \alpha$ is generated by a finite automaton (if we identify each $i \in \mathbb{Q}$ with its base p expansion).

Theorem (Christol, K). *A generalized power series $x = \sum c_i t^i$ is algebraic over $\mathbb{F}_q[t]$ if and only if $\sum c_i t^{ni}$ is automatic for some integer n. (In particular, the support of x is then in $\frac{1}{n}\mathbb{Z}\left[\frac{1}{p}\right]$.)*

The result of Christol is the case of an ordinary power series, which is used as part of the proof.

Computing with automatic series I: Arithmetic operations

Given automata A_1, A_2 generating languages $\mathcal{L}_1, \mathcal{L}_2$ of well-formed base p expansions of rationals in $\mathbb{Z}\left[\frac{1}{p}\right] \cap [0, +\infty)$, there are operations to produce the following:

- A canonical minimal automaton A' generating $\mathcal{L}_1$.

- Automata generating $\mathcal{L}_1 \cup \mathcal{L}_2$, $\mathcal{L}_1 \cap \mathcal{L}_2$, and $\mathcal{L}_1 \setminus \mathcal{L}_2$.

- For each i, an automaton generating those rationals which occur with multiplicity i in $\mathcal{L}_1 + \mathcal{L}_2$ (only if $\mathcal{L}_1, \mathcal{L}_2$ are well ordered).

These enable equality testing, addition, and multiplication of automatic series.

Computing with automatic series II: Extracting roots

Over $\mathbb{F}_p$, Newton's method applied to Chevalley's polynomial

$$x^p - x - t^{-1}$$

extracts the terms $t^{-1/p}, t^{-1/p^2}, \ldots$ in succession and never terminates. Namely, if $x = t^{-1/p} + \cdots + t^{-1/p^k} + y$, we have

$$y^p - y - t^{-1/p^k} = 0$$

and we extract the next term by setting $y^p - t^{-1/p^k}$ to zero.

To avoid hangups like this, one can modify Newton's method by explicitly working around situations like this. This makes it possible to compute approximately with roots of polynomials over $\mathbb{F}_p[t]$.

What about $\mathbb{C}_p$?

Recall that $\mathbb{C}_p$ is the completed algebraic closure of $\mathbb{Q}_p$, which is both complete and algebraically closed.

Let R be the integral closure of $\mathbb{F}_p[\![t]\!]$ in $\mathbb{F}_p((t^{\mathbb{Q}}))$. Then there is an isomorphism

$$\mathcal{O}_{\mathbb{C}_p}/p\mathcal{O}_{\mathbb{C}_p} \cong R/tR.$$

In other words, the rings $\mathcal{O}_{\mathbb{C}_p}$ and R look the same "up to valuation 1".

Thus one can adopt the use of finite automata to compute approximately in $\mathbb{C}_p$ as well. The generalized power series in t are replaced (following Poonen) with "generalized power series in p."

Summary

One can represent approximations to elements of the algebraic closure of $\mathbb{F}_p[t]$ (i.e., approximate local expansions of plane curves over $\mathbb{F}_p$) using finite automata.

To do: implement this scheme and see if it is workable.