Computability Theory at Work: Factoring Polynomials and Finding Roots

Russell Miller

Queens College & CUNY Graduate Center
New York, NY

MAA MathFest
Portland, OR
7 August 2014
Basic Question for Today

Let $F$ be any field, and let $p \in F[X]$ be an arbitrary polynomial. Two problems immediately arise:

- Does $p(X)$ factor (nontrivially) in $F[X]$?
- Does $p(X)$ have a root in $F$? (That is, does $F$ contain a solution to $p(X) = 0$?)
Basic Question for Today

Let $F$ be any field, and let $p \in F[X]$ be an arbitrary polynomial. Two problems immediately arise:

- Does $p(X)$ factor (nontrivially) in $F[X]$?
- Does $p(X)$ have a root in $F$? (That is, does $F$ contain a solution to $p(X) = 0$?)

Question

Which of these two problems is more difficult?
Basic Question for Today

Let $F$ be any field, and let $p \in F[X]$ be an arbitrary polynomial. Two problems immediately arise:

- Does $p(X)$ factor (nontrivially) in $F[X]$?
- Does $p(X)$ have a root in $F$? (That is, does $F$ contain a solution to $p(X) = 0$?)

Question

Which of these two problems is more difficult?

For $p(X)$ of degree $\geq 2$, having a root implies having a factorization. So, finding a root seems harder than finding a factorization.
Basic Question for Today

Let $F$ be any field, and let $p \in F[X]$ be an arbitrary polynomial. Two problems immediately arise:

- Does $p(X)$ factor (nontrivially) in $F[X]$?
- Does $p(X)$ have a root in $F$? (That is, does $F$ contain a solution to $p(X) = 0$?)

Question

Which of these two problems is more difficult?

For $p(X)$ of degree $\geq 2$, having a root implies having a factorization. So, finding a root seems harder than finding a factorization.

But the negative answer is the hard one to prove! And if $p(X)$ has no factorization, then it has no root – so maybe the harder problem is the one about factorization?
Turing-Computable Fields

Defn.
A function \( \varphi : \mathbb{N} \rightarrow \mathbb{N} \) is \textit{computable} if there is a finite program (\( \equiv \) Turing machine) which computes it. (We allow \( \varphi \) to be a \textit{partial function}, i.e. with domain \( \subseteq \mathbb{N} \).)

A subset of \( \mathbb{N} \) is computable if its characteristic function is.

Defn.
A \textit{computable field} \( F \) is a (finite or countable) field whose elements are \( \{ x_0, x_1, x_2, \ldots \} \), in which the field operations \( + \) and \( \cdot \) are given by computable functions \( f \) and \( g \):

\[
x_i + x_j = x_{f(i,j)} \quad x_i \cdot x_j = x_{g(i,j)}
\]
Turing-Computable Fields

**Defn.**

A function \( \varphi : \mathbb{N} \to \mathbb{N} \) is *computable* if there is a finite program (≡ Turing machine) which computes it. (We allow \( \varphi \) to be a *partial function*, i.e. with domain \( \subseteq \mathbb{N} \).) A subset of \( \mathbb{N} \) is computable if its characteristic function is.

**Defn.**

A *computable field* \( F \) is a (finite or countable) field whose elements are \( \{ x_0, x_1, x_2, \ldots \} \), in which the field operations \( + \) and \( \cdot \) are given by computable functions \( f \) and \( g \):

\[
    x_i + x_j = x_{f(i,j)} \quad x_i \cdot x_j = x_{g(i,j)}
\]

The following fields are all isomorphic to computable fields:

\[
    \mathbb{Q}, \mathbb{F}_p, \mathbb{Q}(X_1, X_2, \ldots), \mathbb{F}_p(X_1, X_2, \ldots), \overline{\mathbb{Q}}, \overline{\mathbb{F}_p}
\]

and all finitely generated extensions of these.
Background in Computability

Useful Facts

- There is a noncomputable set $K$ which is\textit{computably enumerable} ($\equiv$ the image of a computable function with domain $\mathbb{N}$). The \textit{Halting Problem} is one example.
- There exists a \textit{universal Turing machine} $\psi : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that every partial computable $\varphi$ is given by $\psi(e, \cdot)$ for some $e$.
- There is a computable bijection from $\mathbb{N}$ onto $\mathbb{N}^* = \bigcup_k \mathbb{N}^k$.

Interesting Fields

1. There is a computable field $F_K$ isomorphic to $\mathbb{Q}[\sqrt{p_n} \mid n \in K]$. (Recall: $K$ is c.e. but not computable; $p_0, p_1, \ldots$ are the primes.) In $F_K$, factoring and having roots are not computable, since

$$n \in K \iff (X^2 - p_n) \text{ has a root} \iff (X^2 - p_n) \text{ factors}.$$ 

2. The field $\mathbb{Q}[\sqrt{p_n} \mid n \notin K]$ is not isomorphic to any computable field.

Russell Miller (CUNY)
The Root Set and the Splitting Set

Since we can enumerate all elements of a computable field $F$, we can also enumerate all polynomials over $F$:

$$F[X] = \{ f_0(X), f_1(X), f_2(X), \ldots \}.$$  

**Defn.**

The *splitting set* $S_F$ and the *root set* $R_F$ of a computable field $F$ are:

$$S_F = \{ n \in \mathbb{N} : (\exists \text{ nonconstant } g, h \in F[X]) \ g(X) \cdot h(X) = f_n(X) \}$$

$$R_F = \{ n \in \mathbb{N} : (\exists a \in F) \ f_n(a) = 0 \}.$$  

$F$ has a *splitting algorithm* if $S_F$ is computable, and a *root algorithm* if $R_F$ is computable.
The Root Set and the Splitting Set

Since we can enumerate all elements of a computable field $F$, we can also enumerate all polynomials over $F$:

$$F[X] = \{f_0(X), f_1(X), f_2(X), \ldots\}.$$ 

**Defn.**

The *splitting set* $S_F$ and the *root set* $R_F$ of a computable field $F$ are:

$$S_F = \{n \in \mathbb{N} : (\exists \text{ nonconstant } g, h \in F[X]) \ g(X) \cdot h(X) = f_n(X)\}$$

$$R_F = \{n \in \mathbb{N} : (\exists a \in F) \ f_n(a) = 0\}.$$ 

$F$ has a *splitting algorithm* if $S_F$ is computable, and a *root algorithm* if $R_F$ is computable.

Bigger questions: find the irreducible factors of $p(X)$, and find all its roots in $F$. These questions reduce to the splitting set and the root set.
Splitting Algorithms

Theorem (Kronecker, 1882)

- The field $\mathbb{Q}$ has a splitting algorithm: it is decidable which polynomials in $\mathbb{Q}[X]$ have factorizations in $\mathbb{Q}[X]$.
- Let $F$ be a computable field of characteristic 0 with a splitting algorithm. Every primitive extension $F(x)$ of $F$ also has a splitting algorithm, which may be found uniformly in the minimal polynomial of $x$ over $F$ (or uniformly knowing that $x$ is transcendental over $F$).

Recall that for $x \in E$ algebraic over $F$, the minimal polynomial of $x$ over $F$ is the unique monic irreducible $f(X) \in F[X]$ with $f(x) = 0$. 
Splitting Algorithms

Theorem (Kronecker, 1882)

- The field \( \mathbb{Q} \) has a splitting algorithm: it is decidable which polynomials in \( \mathbb{Q}[X] \) have factorizations in \( \mathbb{Q}[X] \).
- Let \( F \) be a computable field of characteristic 0 with a splitting algorithm. Every primitive extension \( F(x) \) of \( F \) also has a splitting algorithm, which may be found uniformly in the minimal polynomial of \( x \) over \( F \) (or uniformly knowing that \( x \) is transcendental over \( F \)).

Recall that for \( x \in E \) algebraic over \( F \), the minimal polynomial of \( x \) over \( F \) is the unique monic irreducible \( f(X) \in F[X] \) with \( f(x) = 0 \).

Corollary

For any algebraic computable field \( F \), every finitely generated subfield \( \mathbb{Q}(x_1, \ldots, x_n) \) or \( \mathbb{F}_p(x_1, \ldots, x_n) \) has a splitting algorithm, uniformly in the tuple \( \langle x_1, \ldots, x_d \rangle \).
Comparing $S_F$ and $R_F$

For all computable fields $F$, $S_F$ and $R_F$ are computably enumerable, but may not be computable. With an oracle for $S_F$, we can find all irreducible factors of any given polynomial $p \in F[X]$:

1. Use $S_F$ to determine whether $p$ is irreducible in $F[X]$.
2. If not, search through $F[X]$ for some nontrivial factorization of $p$, and return to Step 1 for each factor.

Therefore, $R_F$ is decidable if one has access to an $S_F$-oracle. (In particular, if $S_F$ is computable, so is $R_F$.) We say that $R_F$ is Turing-reducible to $S_F$, written $R_F \leq_T S_F$. 
Comparing $S_F$ and $R_F$

For all computable fields $F$, $S_F$ and $R_F$ are computably enumerable, but may not be computable. With an oracle for $S_F$, we can find all irreducible factors of any given polynomial $p \in F[X]$: 

1. Use $S_F$ to determine whether $p$ is irreducible in $F[X]$.
2. If not, search through $F[X]$ for some nontrivial factorization of $p$, and return to Step 1 for each factor.

Therefore, $R_F$ is decidable if one has access to an $S_F$-oracle. (In particular, if $S_F$ is computable, so is $R_F$.) We say that $R_F$ is Turing-reducible to $S_F$, written $R_F \leq_T S_F$.

But can we compute $S_F$ from an $R_F$-oracle?
Theorem (Rabin 1960; Frohlich & Shepherdson 1956)

For every computable field $F$, $S_F \subseteq_T R_F$. 

$S_F \equiv_T R_F$
Theorem (Rabin 1960; Frohlich & Shepherdson 1956)

For every computable field $F$, $S_F \leq_T R_F$.

The first proof, by Frohlich & Shepherdson, uses symmetric polynomials. The more elegant proof, by Rabin, embeds $F$ as a subfield $g(F)$ in a computable presentation of its algebraic closure $\overline{F}$. (Rabin’s Theorem also shows that $g(F) \equiv_T S_F$, with $g(F)$ viewed as a subset of $\overline{F}$.)
Comparing $R_F$ and $S_F$

We know that $R_F \equiv_T S_F$. Is there any way to distinguish the complexity of these sets?
Comparing $R_F$ and $S_F$

We know that $R_F \equiv_T S_F$. Is there any way to distinguish the complexity of these sets?

**Defn.**

For sets $A, B \subseteq \mathbb{N}$, we say that $A$ is $m$-reducible to $B$, written $A \leq_m B$, if there is a computable function $f$ such that:

$$(\forall x)[x \in A \iff f(x) \in B].$$
Comparing $R_F$ and $S_F$

We know that $R_F \equiv_T S_F$. Is there any way to distinguish the complexity of these sets?

**Defn.**

For sets $A, B \subseteq \mathbb{N}$, we say that $A$ is *m-reducible to B*, written $A \leq_m B$, if there is a computable function $f$ such that:

$$(\forall x)[x \in A \iff f(x) \in B].$$

**Theorem (M, 2010)**

For all algebraic computable fields $F$, $S_F \leq_m R_F$. However, there exists such a field $F$ with $R_F \not\leq_m S_F$. 
Comparing $R_F$ and $S_F$

We know that $R_F \equiv_T S_F$. Is there any way to distinguish the complexity of these sets?

**Defn.**

For sets $A, B \subseteq \mathbb{N}$, we say that $A$ is \textit{m-reducible to} $B$, written $A \leq_m B$, if there is a computable function $f$ such that:

$$(\forall x)[x \in A \iff f(x) \in B].$$

**Theorem (M, 2010)**

For all algebraic computable fields $F$, $S_F \leq_m R_F$. However, there exists such a field $F$ with $R_F \not\leq_m S_F$.

Problem: Given a polynomial $p(X) \in F[X]$, compute another polynomial $q(X) \in F[X]$ such that

$$p(X) \text{ factors } \iff q(X) \text{ has a root.}$$
$p(X)$ factors in $F[X] \iff q(X)$ has a root in $F$.

Let $F_t$ be the subfield $\mathbb{Q}[x_0, \ldots, x_{t-1}] \subseteq F$ (or $\mathbb{F}_m[x_0, \ldots, x_{t-1}] \subseteq F$). So every $F_t$ has a splitting algorithm.

For a given $p(X)$, find a $t$ with $p \in F_t[X]$. Check first whether $p$ splits there. If so, pick its $q(X)$ to be a linear polynomial. If not, find the splitting field $K_t$ of $p(X)$ over $F_t$, and the roots $r_1, \ldots, r_d$ of $p(X)$ in $K_t$. 
\( p(X) \) factors in \( F[X] \) \iff \( q(X) \) has a root in \( F \).

Let \( F_t \) be the subfield \( \mathbb{Q}[x_0, \ldots, x_{t-1}] \subseteq F \) (or \( \mathbb{F}_m[x_0, \ldots, x_{t-1}] \subseteq F \)). So every \( F_t \) has a splitting algorithm.

For a given \( p(X) \), find a \( t \) with \( p \in F_t[X] \). Check first whether \( p \) splits there. If so, pick its \( q(X) \) to be a linear polynomial. If not, find the splitting field \( K_t \) of \( p(X) \) over \( F_t \), and the roots \( r_1, \ldots, r_d \) of \( p(X) \) in \( K_t \).

**Proposition**

For \( F_t \subseteq L \subseteq K_t \): \( p(X) \) factors in \( L[X] \) \iff there is an \( S \) with \( \emptyset \subset S \subset \{ r_1, \ldots, r_d \} \) such that \( L \) contains all elementary symmetric polynomials in \( S \).

Proof: If \( p = p_0 \cdot p_1 \), let \( S = \{ r_i : p_0(r_i) = 0 \} \), and conversely.
$p(X)$ factors in $F[X] \iff q(X)$ has a root in $F$.

Let $F_t$ be the subfield $\mathbb{Q}[x_0, \ldots, x_{t-1}] \subseteq F$ (or $\mathbb{F}_m[x_0, \ldots, x_{t-1}] \subseteq F$). So every $F_t$ has a splitting algorithm.

For a given $p(X)$, find a $t$ with $p \in F_t[X]$. Check first whether $p$ splits there. If so, pick its $q(X)$ to be a linear polynomial. If not, find the splitting field $K_t$ of $p(X)$ over $F_t$, and the roots $r_1, \ldots, r_d$ of $p(X)$ in $K_t$.

**Proposition**

For $F_t \subseteq L \subseteq K_t$: $p(X)$ factors in $L[X] \iff$ there is an $S$ with $\emptyset \subset S \subset \{r_1, \ldots, r_d\}$ such that $L$ contains all elementary symmetric polynomials in $S$.

**Proof:** If $p = p_0 \cdot p_1$, let $S = \{r_i : p_0(r_i) = 0\}$, and conversely.

**Effective Theorem of the Primitive Element**

Each finite algebraic field extension is generated by a single element, and there is an algorithm for finding such a generator.
$p(X)$ factors in $F[X] \iff q(X)$ has a root in $F$.

For each intermediate field $F_t \subset L_S \subset K_t$ generated by the elementary symmetric polynomials in $S$, let $x_S$ be a primitive generator. Let $q(X)$ be the product of the minimal polynomials $q_S(X) \in F_t[X]$ of each $x_S$. 
\( p(X) \) factors in \( F[X] \) \iff \( q(X) \) has a root in \( F \).

For each intermediate field \( F_t \subsetneq L_S \subsetneq K_t \) generated by the elementary symmetric polynomials in \( S \), let \( x_S \) be a primitive generator. Let \( q(X) \) be the product of the minimal polynomials \( q_S(X) \in F_t[X] \) of each \( x_S \).

\( \Rightarrow \): If \( p(X) \) factors in \( F[X] \), then \( F \) contains some \( L_S \). But then \( x_S \in F \), and \( q(x_S) = 0 \).
$p(X)$ factors in $F[X] \iff q(X)$ has a root in $F$.

For each intermediate field $F_t \subsetneq L_S \subsetneq K_t$ generated by the elementary symmetric polynomials in $S$, let $x_S$ be a primitive generator. Let $q(X)$ be the product of the minimal polynomials $q_S(X) \in F_t[X]$ of each $x_S$.

$\Rightarrow$: If $p(X)$ factors in $F[X]$, then $F$ contains some $L_S$. But then $x_S \in F$, and $q(x_S) = 0$.

$\Leftarrow$: If $q(X)$ has a root $x \in F$, then some $q_S(x) = 0$, so $x$ is $F_t$-conjugate to some $x_S$. Then some $\sigma \in \text{Gal}(K_t/F_t)$ maps $x_S$ to $x$. But $\sigma$ permutes the set $\{r_1, \ldots, r_d\}$, so $x$ generates the subfield containing all elementary symmetric polynomials in $\sigma(S)$. Then $F$ contains the subfield $L_{\sigma(S)}$, so $p(X)$ factors in $F[X]$.

Thus $S_F \leq_m R_F$. 

Building an $F$ with $R_F \not\preceq_m S_F$

Strategy to show that a single $\varphi_e$ is not an $m$-reduction from $R_F$ to $S_F$: have a witness polynomial $q_e(X) = X^5 - X - 1$, say, of degree 5, with splitting field $K_e$ over $\mathbb{Q}$ for which $\text{Gal}(K_e/\mathbb{Q})$ is the symmetric group $S_5$ on the five roots (all irrational) of $q_e$. We wish to make

$$q_e \in R_F \iff \varphi_e(q_e) \not\in S_F.$$ 

If $\varphi_e(q_e)$ halts and equals some polynomial $p_e(X) \in \mathbb{Q}[X]$, then either keep $F = \mathbb{Q}$ (if $p_e$ is reducible there), or add a root of $q_e$ to $\mathbb{Q}$ to form $F$ (if $\deg(p_e) < 2$), or . . .
$q_e$ has no root in $F$ $\iff$ $p_e$ factors over $F$

Let $L$ be the splitting field of $p_e(X)$ over $\mathbb{Q}$, containing all roots $x_1, \ldots, x_n$ of $p_e$. If $\mathbb{Q}[x_1]$ contains no root $r_i$ of $q_e(X)$, then let $F = \mathbb{Q}[x_1]$. Else say (WLOG) $r_1 = h(x_1)$ for some $h(X) \in \mathbb{Q}[X]$. Then each $h(x_j) \in \{r_1, \ldots, r_5\}$, and each $r_i$ is $h(x_j)$ for some $j$. Let $F$ be the fixed field of the subgroup $G_{12}$:

$$G_{12} = \{ \sigma \in \text{Gal}(L/\mathbb{Q}) : \{ \sigma(r_1), \sigma(r_2) \} = \{ r_1, r_2 \} \}.$$ 

Then each $\sigma \in G_{12}$ fixes $I = \{ x_j : h(x_j) \in \{ r_1, r_2 \} \}$ setwise. So $F$ contains all polynomials symmetric in $I$, and $p_e(X)$ splits in $F$. But there is a $\tau \in G_{12}$ which fixes no $r_i$. So $q_e(X)$ has no root in $F$. 

Russell Miller (CUNY)    Factoring and Finding Roots    MathFest 2014 13 / 15
Defeating all $\varphi_e$ at once

The foregoing argument built a computable algebraic field $F$ for which a given $\varphi_e$ was not an $m$-reduction from $R_F$ to $S_F$. This shows that there is no uniform $m$-reduction that works across all such fields.

To see that there is a single such field $F$ with $R_F \nleq_m S_F$, we need to execute the same procedure as above for every possible $m$-reduction $\varphi_e$. The danger here is that, in adding the fixed field of $G_{12}$ to $F$ for one polynomial $p_e$, to satisfy $\varphi_e$, we might add elements which would upset the strategy for defeating other functions $\varphi_{e'}$.

Solution: use a priority argument, in which each $\varphi_e$ is assigned a natural number (in fact, $e$) as its priority. When two strategies clash, the one with higher priority (≡ with smaller $e$) decides what to do, and the other one is injured and starts over with a new polynomial $q_e$. Each individual strategy will be re-started only finitely many times, and will eventually ensure that $\varphi_e$ is not an $m$-reduction.
Standard References on Computable Fields


These slides will be available soon at

qcpages.qc.cuny.edu/~rmiller/slides.html