Algebraic Fields and Computable Categoricity

Russell Miller & Alexandra Shlapentokh

Queens College & CUNY Graduate Center
New York, NY

East Carolina University
Greenville, NC.

George Washington University Logic Seminar

19 November 2010

(Some work joint with Denis Hirschfeldt, University of Chicago, and Ken Kramer, CUNY.)

Slides available at
`qc.edu/~rmiller/slides.html`

Computable Categoricity

Defn.

A computable structure $\mathcal{A}$ is *computably categorical* if for each computable $\mathcal{B} \cong \mathcal{A}$ there is a computable isomorphism from $\mathcal{A}$ onto $\mathcal{B}$.

Examples: (Dzgoev, Goncharov; Remmel; Lempp, McCoy, M., Solomon)

- A linear order is computably categorical iff it has only finitely many adjacencies.
- A Boolean algebra is computably categorical iff it has only finitely many atoms.
- An ordered Abelian group is computably categorical iff it has finite rank ($\equiv$ basis as $\mathbb{Z}$-module).
- For trees, the known criterion is recursive in the height and not easily stated!

Computably Categorical Fields

The following fields are all computably categorical:

- $\mathbb{Q}$.
- All finitely generated extensions of $\mathbb{Q}$ or $\mathbb{F}_p$.
- Every algebraically closed field of finite transcendence degree over $\mathbb{Q}$ or $\mathbb{F}_p$.
- All normal algebraic extensions of $\mathbb{Q}$ or $\mathbb{F}_p$.
- Some (but not all) non-normal algebraic extensions of $\mathbb{Q}$ or $\mathbb{F}_p$.
- Certain fields (but not very many!) of infinite transcendence degree over $\mathbb{Q}$. (Miller-Schoutens.)

Relative Computable Categoricity

Defn.

A computable structure $\mathcal{A}$ is *relatively computably categorical* if for each $\mathcal{B} \cong \mathcal{A}$ with domain ω, there is an isomorphism from $\mathcal{A}$ onto $\mathcal{B}$ which is computable from an oracle for $\mathcal{B}$.

Clearly this implies computable categoricity – but the converse is false! Certain computably categorical structures are not relatively computably categorical.

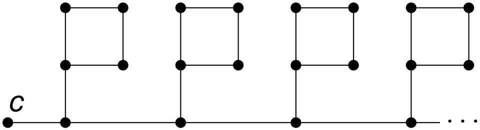
Scott Families

Defn.

A *Scott family* for a structure $\mathcal{A}$ is a set Σ of formulas $\psi(x_0, \ldots, x_n, \vec{c})$, over a fixed finite tuple $\vec{c}$ of parameters from $\mathcal{A}$, such that

- For all $\vec{a} \in \mathcal{A}^{<\omega}$, some $\psi \in \Sigma$ has $\models_{\mathcal{A}} \psi(\vec{a}, \vec{c})$.
- If $\vec{a}, \vec{b} \in \mathcal{A}^n$ satisfy the same $\psi \in \Sigma$, then some $\alpha \in \mathrm{Aut}(\mathcal{A})$ has $\alpha(a_i) = b_i$ for all $i \leq n$.

Example:



Thm. (Ash-Knight-Manasse-Slaman; Chisholm)

A computable structure $\mathcal{A}$ is relatively computably categorical iff $\mathcal{A}$ has a computably enumerable Scott family of existential formulas.

Algebraic Fields with Splitting Algorithms

Definitions

A field is *algebraic* if it is an algebraic extension of its prime subfield (either $\mathbb{Q}$ or $\mathbb{F}_p$).
A computable field F has a *splitting algorithm* if its *splitting set* S_F (or equivalently its *root set* R_F) is computable:

$$S_F = \{p \in F[X] : p \text{ factors properly in } F[X]\}$$
$$R_F = \{p \in F[X] : (\exists a \in F)\ p(a) = 0\}$$

Algebraic Fields with Splitting Algorithms

Definitions

A field is *algebraic* if it is an algebraic extension of its prime subfield (either $\mathbb{Q}$ or $\mathbb{F}_p$).
A computable field F has a *splitting algorithm* if its *splitting set* S_F (or equivalently its *root set* R_F) is computable:

$$S_F = \{p \in F[X] : p \text{ factors properly in } F[X]\}$$
$$R_F = \{p \in F[X] : (\exists a \in F)\ p(a) = 0\}$$

Facts:

- All finite algebraic extensions of $\mathbb{Q}$ and $\mathbb{F}_p$ have splitting algorithms, uniformly in their generators.
- An algebraic field F has a splitting algorithm iff all computable fields isomorphic to F have splitting algorithms.

Orbit Relations for Fields

Definition

For a computable field F, the *full orbit relation* A_F for F is the set:

$$\{\langle a_1, \ldots, a_n; b_1, \ldots, b_n\rangle : (\exists \sigma \in \mathrm{Aut}(F))(\forall i)\ \sigma(a_i) = b_i\} \subseteq \cup_n F^{2n}.$$

For algebraic F, by the Effective Theorem of the Primitive Element, A_F is computably isomorphic to the *orbit relation* B_F of F, defined by the action of $\mathrm{Aut}(F)$:

$$B_F = \{\langle a; b\rangle \in F^2 : (\exists \sigma \in \mathrm{Aut}(F))\ \sigma(a) = b\}.$$

Orbit Relations for Fields

Definition

For a computable field F, the *full orbit relation* A_F for F is the set:

$$\{\langle a_1,\ldots,a_n; b_1,\ldots,b_n\rangle : (\exists\sigma \in \mathrm{Aut}(F))(\forall i)\ \sigma(a_i) = b_i\} \subseteq \cup_n F^{2n}.$$

For algebraic F, by the Effective Theorem of the Primitive Element, A_F is computably isomorphic to the *orbit relation* B_F of F, defined by the action of $\mathrm{Aut}(F)$:

$$B_F = \{\langle a; b\rangle \in F^2 : (\exists\sigma \in \mathrm{Aut}(F))\ \sigma(a) = b\}.$$

For algebraic $F \supseteq \mathbb{Q}$ in general, $\mathcal{B}_F$ is Π^0_2:

$$\langle a; b\rangle \in B_F \text{ iff } (\forall q \in \mathbb{Q}[X,Y])\ [q(a,Y) \in R_F \iff q(b,Y) \in R_F].$$

However, when F has a splitting algorithm, B_F becomes Π^0_1. (And when $F \supseteq \mathbb{Q}$ is a *normal* algebraic extension, B_F is computable.)

Computable Categoricity

Theorem (MS 2010)

Let F be a computable algebraic field with a splitting algorithm. Then F is computably categorical iff B_F is computable.

Since F has a splitting algorithm, B_F is Π_1^0, so the complexity of this condition is Σ_3^0 in indices for F and its splitting algorithm.

Computable Categoricity

Theorem (MS 2010)

Let F be a computable algebraic field with a splitting algorithm. Then F is computably categorical iff B_F is computable.

Since F has a splitting algorithm, B_F is Π^0_1, so the complexity of this condition is Σ^0_3 in indices for F and its splitting algorithm.

Corollary

A computable algebraic field with a splitting algorithm is computably categorical iff it is relatively computably categorical.

(The proof below relativizes easily.)

B_F Computable $\implies$ F Computably Categorical

Sketch of Proof: Suppose we have defined $f_s : F_s = \mathbb{Q}(x_0, \ldots, x_s) \to E$, where $F = \{x_0, x_1, \ldots\}$, and $F_s \subseteq F_{s+1}$ are both normal within F.
Assume f_s extends to an isomorphism $\psi : F \to E$.
Find a primitive generator $a \in F$ of F_{s+1}, and find its minimal polynomial $p(X) \in F_s[X]$. Let $a = y_1, y_2, \ldots, y_d$ be all its roots in F.

B_F Computable $\implies$ F Computably Categorical

Sketch of Proof: Suppose we have defined $f_s : F_s = \mathbb{Q}(x_0, \ldots, x_s) \to E$, where $F = \{x_0, x_1, \ldots\}$, and $F_s \subseteq F_{s+1}$ are both normal within F.
Assume f_s extends to an isomorphism $\psi : F \to E$.
Find a primitive generator $a \in F$ of F_{s+1}, and find its minimal polynomial $p(X) \in F_s[X]$. Let $a = y_1, y_2, \ldots, y_d$ be all its roots in F.
For each $j \leq d$ with $\langle a, y_j \rangle \notin B_F$, find some $q_j \in F_s[X, Y]$ with

$$q_j(a, Y) \in R_F \;\; \& \;\; q_j(y_j, Y) \notin R_F.$$

B_F Computable $\implies$ F Computably Categorical

Sketch of Proof: Suppose we have defined $f_s : F_s = \mathbb{Q}(x_0, \ldots, x_s) \to E$, where $F = \{x_0, x_1, \ldots\}$, and $F_s \subseteq F_{s+1}$ are both normal within F.
Assume f_s extends to an isomorphism $\psi : F \to E$.
Find a primitive generator $a \in F$ of F_{s+1}, and find its minimal polynomial $p(X) \in F_s[X]$. Let $a = y_1, y_2, \ldots, y_d$ be all its roots in F.
For each $j \leq d$ with $\langle a, y_j \rangle \notin B_F$, find some $q_j \in F_s[X, Y]$ with

$$q_j(a, Y) \in R_F \;\;\&\;\; q_j(y_j, Y) \notin R_F.$$

Then find all roots $z_1, \ldots, z_d \in E$ of the image $\overline{p}(X)$ of $p(X)$ under f_s.
Define $f_{s+1}(a)$ to be any z_k for which all the polynomials $\overline{q_j}(z_k, Y)$ have roots in E. Then $f_s \subseteq f_{s+1}$ and $\langle a, \psi^{-1}(z_k) \rangle \in B_F$, so f_{s+1} must extend to the isomorphism $\psi \circ \sigma : F \to E$, where $\sigma \in \mathrm{Aut}(F)$ has $\sigma(a) = \psi^{-1}(z_k)$ and $(\forall i)\sigma(x_i) = x_i$. By iterating, we get a computable isomorphism.

F Computably Categorical $\implies B_F$ Computable

Proof: Here we assume that F is computably categorical, and build a computable $E \cong F$. In doing so, whenever possible, we build E so that φ_e will *not* be an isomorphism. (This uses a priority construction, based on the values e.) For the least e such that φ_e defies all our attempts, the isomorphism φ_e will allow us to compute B_F.

F Computably Categorical $\Longrightarrow$ B_F Computable

Proof: Here we assume that F is computably categorical, and build a computable $E \cong F$. In doing so, whenever possible, we build E so that φ_e will *not* be an isomorphism. (This uses a priority construction, based on the values e.) For the least e such that φ_e defies all our attempts, the isomorphism φ_e will allow us to compute B_F.

At each stage $s+1$, we look for the least e such that for some $a, b \in F_s$, $\varphi_{e,s}(a)\downarrow$ and $\langle a, b\rangle \in B_{F_s}$, yet $\langle a, b\rangle \notin B_{F_{s+1}}$. (Essentially we search for $q \in \mathbb{Q}[X, Y]$ such that $q(b, Y)$ has a root in F and $q(a, Y)$ does not.) Then, when building the extension E_{s+1} of E_s, we add a root of $q(\varphi_e(a), Y)$, so that our isomorphism $F_{s+1} \to E_{s+1}$ has $b \mapsto \varphi_e(a)$, and no isomorphism $F \to E$ has $a \mapsto \varphi_e(a)$.

F Computably Categorical $\implies$ B_F Computable

Proof: Here we assume that F is computably categorical, and build a computable $E \cong F$. In doing so, whenever possible, we build E so that φ_e will *not* be an isomorphism. (This uses a priority construction, based on the values e.) For the least e such that φ_e defies all our attempts, the isomorphism φ_e will allow us to compute B_F.

At each stage $s+1$, we look for the least e such that for some $a, b \in F_s$, $\varphi_{e,s}(a)\downarrow$ and $\langle a, b\rangle \in B_{F_s}$, yet $\langle a, b\rangle \notin B_{F_{s+1}}$. (Essentially we search for $q \in \mathbb{Q}[X, Y]$ such that $q(b, Y)$ has a root in F and $q(a, Y)$ does not.) Then, when building the extension E_{s+1} of E_s, we add a root of $q(\varphi_e(a), Y)$, so that our isomorphism $F_{s+1} \to E_{s+1}$ has $b \mapsto \varphi_e(a)$, and no isomorphism $F \to E$ has $a \mapsto \varphi_e(a)$.

If $\varphi_e : F \to E$ is an isomorphism, with e minimal, then

$$(\forall s \geq s_0)\ [\varphi_{e,s}(a)\downarrow \implies (\forall b)[\langle a, b\rangle \in B_{F_s} \implies \langle a, b\rangle \in B_F]].$$

So B_F is c.e., as well as Π^0_1.

Algebraic Fields Without Splitting Algorithms

Theorem (easy corollary of Ash-Knight-Manasse-Slaman)

Let F be a computable, relatively computably categorical, algebraic field. Then the orbit relation B_F is computably enumerable.

This generalizes our theorem on fields with splitting algorithms, since for those fields, B_F is automatically Π^0_1.

Algebraic Fields Without Splitting Algorithms

Theorem (easy corollary of Ash-Knight-Manasse-Slaman)

Let F be a computable, relatively computably categorical, algebraic field. Then the orbit relation B_F is computably enumerable.

This generalizes our theorem on fields with splitting algorithms, since for those fields, B_F is automatically Π^0_1.
However, if F has no splitting algorithm, then B_F can be c.e., or even computable, with F *not* computably categorical.

Example: Begin to build $E = F = \mathbb{Q}(\theta_0)$ with $\theta_0^3 = 2$. If $\varphi_e(\theta_0)\downarrow = \theta_0$, then adjoin to E and F two more cube roots θ_1, θ_2 of 2. Also adjoin to E a square root of θ_0, and to F a square root of θ_1. Then $\varphi_e : E \to F$ is not an isomorphism, yet B_E and B_F remain computable.

Full Construction: $B_F \leq_T \emptyset$, but F not C.C.

Lemma

For every Galois extension $\mathbb{Q} \subseteq E$ and every $d > 1$, there exists a monic $f(X) \in \mathbb{Z}[X]$ of degree d such that $\mathrm{Gal}(K/\mathbb{Q}) \cong S_d$ and the splitting field K of $f(X)$ over $\mathbb{Q}$ is linearly disjoint from E.

Corollary

There is a computable sequence $f_0, f_1, \ldots$ in $\mathbb{Z}[X]$ whose splitting fields K_i each have Galois group S_7 over $\mathbb{Q}$ and such that each K_i is linearly disjoint from the compositum of all K_j ($j \neq i$).

Use this sequence to build F and $\tilde{F}$. For distinct roots r_1, r_2, r_3, r_4 of K_i, first adjoin $(r_1 + r_2)$ to $\mathbb{Q}$ in both F and $\tilde{F}$. If $\varphi_i(r_1 + r_2)\downarrow = (r_1 + r_2)$, then adjoin $(r_3 + r_4)$ to both fields, r_1 to F, and r_3 to $\tilde{F}$. Then $F \cong \tilde{F}$, but not via φ_i. However, F is rigid (except for interchanging r_1 with r_2, if they entered F). Thus B_F is computable.

The Isomorphism Tree

Let $F \cong \tilde{F}$ be computable algebraic fields, and let $z_1, z_2, z_3, \ldots$ be a sequence of elements generating F. For simplicity, assume $\mathbb{Q} = \mathbb{Q}(z_0) \subseteq \mathbb{Q}(z_1) \subseteq \mathbb{Q}(z_2) \subseteq \cdots$, with $z_0 = 1$.
Compute polynomials $f_{i+1} \in \mathbb{Q}[Y, Z]$ s.t. $f_{i+1}(z_i, Z)$ is the minimal polynomial of z_{i+1} over $\mathbb{Q}(z_i)$, for each i.

Defn.

The *isomorphism tree* $I_{F\tilde{F}}$ is the following subtree of $\tilde{F}^{<\omega}$:

$$\{\langle \tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_m \rangle : (\forall i < m)\ \tilde{f}_{i+1}(\tilde{z}_i, \tilde{z}_{i+1}) = 0\}.$$

Here $\tilde{f}_i(Y, Z)$ is the image of $f_i(Y, Z)$ under the isomorphism of the prime subfields.

So $I_{F\tilde{F}}$ is a finite-branching tree, and paths through it correspond to isomorphisms from F onto $\tilde{F}$, with each path Turing-equivalent to its isomorphism.

Scott Families for Algebraic Fields

To enumerate a Scott family Σ for F, we need to give an $\exists$-formula ψ_i for each z_i such that, for all $z \in F$,

$$\psi_i(z) \text{ holds in } F \iff \langle z_i, z \rangle \in B_F.$$

For the finitely many roots of $f_i(z_{i-1}, Z)$ in F, ψ_i needs to know some level m of I_{FF} such that all nodes at level i with extensions to level m are extendible (i.e. lie on paths).

If we have a computable function which gives such a level m for every m, then F has a c.e. Scott family, hence is relatively computably categorical. This function gives a computable bound on the height of the tree I_{FF} above nonextendible nodes.

Back to Computable Categoricity

Theorem (HKMS 2010)

There exists a computable algebraic field F which is computably categorical, but not relatively c.c. In particular, B_F is not Σ^0_2.

Proof: a tree construction of F.
A node ρ at level $2e$ has two outcomes: $\cong$ and $\ncong$. It tries to ensure that if the structure computed by the e-th Turing program is a field K_e isomorphic to F, then some program P_ρ computes an isomorphism between them.
Each time larger initial fragments of F and K_e are found to embed into each other, ρ makes its stronger outcome $\cong$ eligible. This outcome does not allow lower-priority nodes to do anything until F and K_e match up well enough for P_ρ to be sure how to build its isomorphism.

Back to Computable Categoricity

Theorem (HKMS 2010)

There exists a computable algebraic field F which is computably categorical, but not relatively c.c. In particular, B_F is not Σ^0_2.

Proof: a tree construction of F.
A node ρ at level $2e$ has two outcomes: $\cong$ and $\ncong$. It tries to ensure that if the structure computed by the e-th Turing program is a field K_e isomorphic to F, then some program P_ρ computes an isomorphism between them.
Each time larger initial fragments of F and K_e are found to embed into each other, ρ makes its stronger outcome $\cong$ eligible. This outcome does not allow lower-priority nodes to do anything until F and K_e match up well enough for P_ρ to be sure how to build its isomorphism.

Suppose ρ is on the true path. If $F \cong K_e$, then $\rho\hat{}\langle\cong\rangle$ will also be on the true path, and P_ρ will compute an isomorphism. (Finitely much information is needed: ρ, and the last stage at which ρ is initialized.)

Computable Categoricity, continued

Nodes τ at levels $2e + 1$ ensure that the e-th partial computable function φ_e does not compute an m-reduction from B_F to $\emptyset''$. (If this holds for every e, then $B_F \not\leq_m \emptyset''$, hence cannot be Σ^0_2.)

τ adds to F two $\mathbb{Q}$-conjugates x_τ and y_τ. At all stages, there will be two distinct z_s and z_t already in F such that the minimal polynomials of each over x_τ have no root over y_τ. Whenever $W_{\varphi_e(\langle x_\tau, y_\tau \rangle)}$ gets a new element, we add a root over y_τ of the minimal polynomial of z_s over x_τ (where $s < t$), but also add a new $u > t$ for which F has no root over y_τ of the minimal polynomial of z_u over x_τ. Therefore, if τ is never injured, then $\langle x_\tau, y_\tau \rangle \in B_F$ iff $W_{\varphi_e(\langle x_\tau, y_\tau \rangle)}$ is infinite.

B_F and the Galois Group of F over L

Extend the definition to field extensions F/L with F computable:

$$B_{F/L} = \{\langle a; b\rangle \in F^2 : (\exists \sigma \in \mathrm{Gal}(F/L))\ \sigma(a) = b\}.$$

For algebraic F/L, if L is a c.e. subfield of F, $B_{F/L}$ is always Π^0_2.

B_F and the Galois Group of F over L

Extend the definition to field extensions F/L with F computable:

$$B_{F/L} = \{\langle a; b\rangle \in F^2 : (\exists \sigma \in \mathrm{Gal}(F/L))\ \sigma(a) = b\}.$$

For algebraic F/L, if L is a c.e. subfield of F, $B_{F/L}$ is always Π^0_2.

Let $B_{F/L}$ be c.e. Then given $\langle a, b\rangle \in B_{F/L}$, we can compute some $\sigma \in \mathrm{Gal}(F/L)$ with $\sigma(a) = b$. Let $F = \{x_0, x_1, \ldots\}$, and let $\sigma(x_s)$ be the first x_t with $\langle a, x_0, \ldots, x_s; b, \sigma(x_0), \ldots, \sigma(x_{s-1}), x_t\rangle \in A_{F/L}$.

B_F and the Galois Group of F over L

Extend the definition to field extensions F/L with F computable:

$$B_{F/L} = \{\langle a; b\rangle \in F^2 : (\exists \sigma \in \mathrm{Gal}(F/L))\ \sigma(a) = b\}.$$

For algebraic F/L, if L is a c.e. subfield of F, $B_{F/L}$ is always Π^0_2.

Let $B_{F/L}$ be c.e. Then given $\langle a, b\rangle \in B_{F/L}$, we can compute some $\sigma \in \mathrm{Gal}(F/L)$ with $\sigma(a) = b$. Let $F = \{x_0, x_1, \ldots\}$, and let $\sigma(x_s)$ be the first x_t with $\langle a, x_0, \ldots, x_s; b, \sigma(x_0), \ldots, \sigma(x_{s-1}), x_t\rangle \in A_{F/L}$. We suggest:

Definition

A computable algebraic extension F/L has *computably approximable Galois group* $\mathrm{Gal}(F/L)$ if $B_{F/L}$ is computably enumerable.

$\mathrm{Gal}(F/L)$ is essentially a type-2 computable object, in the sense of computable analysis. (It may have 2^ω-many elements!)

Automorphism Groups in General

Defn.

For any structure $\mathcal{M}$ with domain ω in which all orbits are finite, say that Aut($\mathcal{M}$) is ***d**-computably approximable* if the following set is computably enumerable in the Turing degree ***d***:

$$A_{\mathcal{M}} = \{\langle \vec{a}; \vec{b} \rangle \in \bigcup_n \left(\omega^{2n}\right) : (\exists \sigma \in \mathrm{Aut}(\mathcal{M}))(\forall i < n) \sigma(a_i) = b_i\}.$$

In general $A_{\mathcal{M}}$ is Σ^1_1. For relatively computably categorical structures $\mathcal{M}$, Aut($\mathcal{M}$) is $\mathcal{M}$-computably approximable: enumerate $\langle \vec{a}; \vec{b} \rangle$ into $A_{\mathcal{M}}$ whenever some ψ in a (computably enumerable) Scott family for $\mathcal{M}$ is found to be satisfied by both $\vec{a}$ and $\vec{b}$.

Automorphism Groups in General

Defn.

For any structure $\mathcal{M}$ with domain ω in which all orbits are finite, say that $\text{Aut}(\mathcal{M})$ is ***d**-computably approximable* if the following set is computably enumerable in the Turing degree ***d***:

$$A_{\mathcal{M}} = \{\langle \vec{a}; \vec{b} \rangle \in \bigcup_n \left(\omega^{2n}\right) : (\exists \sigma \in \text{Aut}(\mathcal{M}))(\forall i < n) \sigma(a_i) = b_i\}.$$

In general $A_{\mathcal{M}}$ is Σ^1_1. For relatively computably categorical structures $\mathcal{M}$, $\text{Aut}(\mathcal{M})$ is $\mathcal{M}$-computably approximable: enumerate $\langle \vec{a}; \vec{b} \rangle$ into $A_{\mathcal{M}}$ whenever some ψ in a (computably enumerable) Scott family for $\mathcal{M}$ is found to be satisfied by both $\vec{a}$ and $\vec{b}$.

However, Steiner has found computable structures $\mathcal{M}$ with all orbits finite and $A_{\mathcal{M}}$ computable, such that $\mathcal{M}$ is not computably categorical. For instance, let $\mathcal{M}$ be an equivalence relation with exactly one equivalence class of each finite size. We saw the same above for a computable algebraic field.

Standard References on Computable Fields

- A. Frohlich & J.C. Shepherdson; Effective procedures in field theory, *Phil. Trans. Royal Soc. London, Series A* **248** (1956) 950, 407-432.
- M. Rabin; Computable algebra, general theory, and theory of computable fields, *Transactions of the American Mathematical Society* **95** (1960), 341-360.
- Yu.L. Ershov; Theorie der Numerierungen III, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **23** (1977) 4, 289-371.
- G. Metakides & A. Nerode; Effective content of field theory, *Annals of Mathematical Logic* **17** (1979), 289-320.
- M.D. Fried & M. Jarden, *Field Arithmetic* (Berlin: Springer-Verlag, 1986).
- V. Stoltenberg-Hansen & J.V. Tucker; Computable rings and fields, in *Handbook of Computability Theory*, ed. E.R. Griffor (Amsterdam: Elsevier, 1999), 363-447.