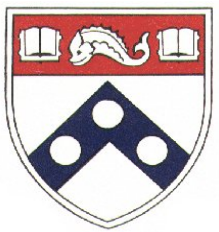
From Saturated Embeddings to Explicit Algorithms

Henry Towsner



University of Pennsylvania

May 20, 2023

Definition

A theory T admits *quantifier elimination* if whenever $\phi(\vec{x})$ is a formula in the language of T, there is a quantifier-free formula $\psi(\vec{x})$ such that $T \vdash \forall \vec{x}\, \phi(\vec{x}) \leftrightarrow \psi(\vec{x})$.

Definition

A theory T admits *quantifier elimination* if whenever $\phi(\vec{x})$ is a formula in the language of T, there is a quantifier-free formula $\psi(\vec{x})$ such that $T \vdash \forall \vec{x}\, \phi(\vec{x}) \leftrightarrow \psi(\vec{x})$.

Examples:

- Dense linear orders without endpoints (i.e. the theory of $(\mathbb{Q}, <)$),
- Random graph,
- Torsion-free divisible abelian groups,
- Algebraically closed fields,
- etc.

Definition

A theory T admits *quantifier elimination* if whenever $\phi(\vec{x})$ is a formula in the language of T, there is a quantifier-free formula $\psi(\vec{x})$ such that $T \vdash \forall \vec{x}\, \phi(\vec{x}) \leftrightarrow \psi(\vec{x})$.

Examples:

- Dense linear orders without endpoints (i.e. the theory of $(\mathbb{Q}, <)$),
- Random graph,
- Torsion-free divisible abelian groups,
- Algebraically closed fields,
- etc.

A well-known non-example is Presburger arithmetic, the theory of $(\mathbb{N}, <, 0, S, +)$: the formula $\exists y\, y + y = x$ defines even numbers and is not equivalent to a combination of equalities and inequalities. (The expansion of Presburger arithmetic by new predicates "divisible by n" for each fixed n does admit quantifier elimination.)

All these examples (and many others) are computable theories in countable languages, and there are proofs of quantifier elimination which are algorithmic: given a formula ϕ, they provide an explicit algorithm which takes $\phi(\vec{x})$ and produces the formula $\psi(\vec{x})$.

All these examples (and many others) are computable theories in countable languages, and there are proofs of quantifier elimination which are algorithmic: given a formula ϕ, they provide an explicit algorithm which takes $\phi(\vec{x})$ and produces the formula $\psi(\vec{x})$.

Many modern proofs of quantifier-elimination in model theory, however, do not give an explicit algorithm. They relay on *saturated embedding tests*: theorems that derive quantifier elimination from facts about embeddings into saturated models.

Theorem (Proposition 4.3.28 of Marker)

T has quantifier elimination if and only if whenever $\mathcal{M} \models T$, $A \subseteq M$, $\mathcal{N} \models T$ is $|M|^+$-saturated, and $f : A \to \mathcal{N}$ is a homomorphism, f extends to an embedding of $\mathcal{M}$ into $\mathcal{N}$.

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Proof.

To show quantifier elimination, it suffices to show that whenever $\phi(x, \vec{y})$ is a a quantifier-free formula with the displayed free variables, there is a quantifier-free formula $\psi(\vec{y})$ so that $T \vdash \psi(\vec{y}) \leftrightarrow \exists x\, \phi(x, \vec{y})$.

□

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Proof.

So let $\phi(x, \vec{y})$ be given. We work in an extension of the language of T with some fresh constant symbols for $\vec{d}$. If $T \cup \{\forall x \neg\phi(x, \vec{d})\}$ is inconsistent then $\forall x \neg\phi(x, \vec{d})$ is equivalent to $\bot$.

Otherwise, $T \cup \{\forall x \neg\phi(x, \vec{d})\}$ is consistent, so there let $\mathcal{N}$ be an ω-saturated model.

□

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Proof.

Otherwise, let Σ be the set of quantifier-free sentences true in $\mathcal{N}$.

Our goal is to show that Σ implies $\forall x \neg\phi(x, \vec{d})$. Then, by compactness, there will be some finite subset of Σ which is equivalent to $\forall x \neg\phi(x, \vec{d})$.

□

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Proof.

$\mathcal{N}$ is an ω-saturated model of $T \cup \{\forall x \neg\phi(x, \vec{d})\}$. Σ is the quantifier-free sentences of $\mathcal{N}$.

Suppose towards a contradiction that $T \cup \Sigma \cup \{\exists x \, \phi(x, \vec{d})\}$ is consistent. Let $\mathcal{M}$ be a model of $T \cup \Sigma \cup \{\exists x \, \phi(x, \vec{d})\}$.

Let $A = \vec{d}$ and f map the copy of $\vec{d}$ in $\mathcal{M}$ to the copy in $\mathcal{N}$.

□

Theorem

Let T be a theory. Suppose that whenever $\mathcal{M}$ and $\mathcal{N}$ are models of T, $\mathcal{N}$ is ω-saturated, $A \subseteq \mathcal{M}$ is finite, $f : A \to \mathcal{N}$ is a homomorphism, and $a \in |\mathcal{M}| \setminus A$, there is a homomorphism $g : A \cup \{a\} \to \mathcal{N}$ extending f.
Then T has quantifier elimination.

Proof.

$\mathcal{N}$ is an ω-saturated model of $T \cup \{\forall x \, \neg\phi(x, \vec{d})\}$. Σ is the quantifier-free sentences of $\mathcal{N}$. $\mathcal{M}$ is a model of $T \cup \Sigma \cup \{\exists x \, \phi(x, \vec{d})\}$.

Pick $a \in |\mathcal{M}|$ so that $\mathcal{M} \vDash \phi(a, \vec{d})$. The assumption gives us a homomorphism g mapping $\vec{d}^{\mathcal{M}}$ to $\vec{d}^{\mathcal{N}}$ and a to some element $g(a)$ witnessing that $\mathcal{N} \vDash \phi(a, \vec{d})$, contradicting the construction of $\mathcal{N}$.

□

Many quantifier elimination theorems have been proven using this result (or various technical modifications) in suitable languages:

- many theories extending algebraically closed fields: differentially closed fields, "p-adically" closed fields, algebraically closed valued fields, and so on.
- the reals, or o-minimal structures more generally, augmented by various predicates
- The culminating result of Aschenbrenner–van den Dries–van der Hoeven's book Asymptotic Differential Algebra and Model Theory of Transseries is a quantifier elimination result for a certain theory of transseries using precisely the theorem above.

In some cases, algorithms were subsequently found, but in many cases no algorithms are known.

When T is a computable (or computably enumerable) theory, the statement that T admits quantifier elimination can be encoded as a Π^0_2 statement in the language of arithmetic: it says that for every (code for) a formula ϕ, there exists (a code for) a quantifier-free formula ψ together with (a code for) a deduction from T showing the equivalence.

General meta-theorems show that when we have a proof of a Π^0_2 statement, it is generally possible to extract an algorithm from it.

When T is a computable (or computably enumerable) theory, the statement that T admits quantifier elimination can be encoded as a Π^0_2 statement in the language of arithmetic: it says that for every (code for) a formula ϕ, there exists (a code for) a quantifier-free formula ψ together with (a code for) a deduction from T showing the equivalence.

General meta-theorems show that when we have a proof of a Π^0_2 statement, it is generally possible to extract an algorithm from it.

These methods would apply straightforwardly if our proofs were, say, given as formal deductions in Peano arithmetic. There has been extensive work the methods of proof mining to proofs which, at least superficially, go beyond arithmetic reasoning, like arguments about analytic spaces or proofs which use ultraproducts or nonstandard analysis.

But none of these methods directly apply to reasoning about infinite models, especially uncountable ones.

The solution is to replace reasoning about models with computational reasoning. What is a computational notion we can substitute for a model?

The solution is to replace reasoning about models with computational reasoning. What is a computational notion we can substitute for a model?

One answer is a computable model with universe $\mathbb{N}$. This is a bit too narrow: recall from the proof that a typical example of a model for us is "a model of $T \cup \Sigma \cup \{\exists x\, \phi(a, \vec{d})\}$, if this is consistent". There may be no computable model of this theory.

The solution is to replace reasoning about models with computational reasoning. What is a computational notion we can substitute for a model?

One answer is a computable model with universe $\mathbb{N}$. This is a bit too narrow: recall from the proof that a typical example of a model for us is "a model of $T \cup \Sigma \cup \{\exists x\, \phi(a, \vec{d})\}$, if this is consistent". There may be no computable model of this theory.

Instead, we will work with a function which attempts to enumerate the facts about the model, but is permitted to be wrong for a while, as long as it eventually gets it right.

Definition

An *approximation of a model of T* is a function h such that

- for each n, $h(n)$ is a finite set of sentences in $\mathcal{L}_{\mathbb{N}}$,
- for every ϕ, either ϕ or $\neg\phi$ is in $h(n)$ for all but finitely many n,
- every axiom σ of T is eventually in $f(n)$,
- if σ is eventually in $h(n)$ and $\sigma \vdash \tau$ then τ is eventually in $h(n)$.

Then we can take $\lim_n h(n)$ to be those σ which are eventually in $h(n)$, and $\lim_n h(n)$ will always be a complete (not necessarily consistent) theory extending T. We say h is consistent if $\bot \notin \lim_n h(n)$.

For instance, we can construct a computable approximation to a model of $T \cup \Sigma \cup \{\exists x\, \phi(a, \vec{d})\}$:

- let $\{\psi_1, \ldots, \psi_m, \ldots\}$ be a computable enumeration of formulas,
- $h(n)$ will be a complete subset of $S_n = \{\psi_1, \ldots, \psi_n, \neg\psi_1, \ldots, \neg\psi_n\}$ for all n,
- we determine whether $\psi_i \in h(n)$ with $i < n$ by:
  - if $\psi_i \in T$ or $\neg\psi_i \in T$, the corresponding formula is in $h(n)$,
  - if there is a deduction with code $\leq n$ putting ψ_i or $\neg\psi_i$ in Σ, the corresponding formula is in $h(n)$,
  - for those formulas not settled in this way, we place them in $h(n)$ in order unless there is a deduction with code $\leq n$ showing that this would lead to an inconsistent set, in which case we put the negation in $h(n)$.

How can we make sense of saturated models in this context?

A typical pattern is that when we try to push uncountable objects down so we can deal with them computably, we end up with "higher order" objects.

How can we make sense of saturated models in this context?

A typical pattern is that when we try to push uncountable objects down so we can deal with them computably, we end up with "higher order" objects.

For instance, a set S is uncountable if there is no surjection $f : \mathbb{N} \to S$. Sometimes the right analog is to work with functionals: we functionals F and S so that $S(F)$ is a set and $F(S(F)) : \mathbb{N} \to S(F)$, and S is uncountable if $F(S(F))$ is never surjective.

Our analog of embedding a model in a saturated model is to have interrelated functions ρ and h, a finite set A and an $a \notin A$ where:

- ρ and h are both approximations of models,
- h has to agree with ρ about quantifier-free statements about A,
- ρ has to agree with h about quantifier-free statements about $A \cup \{a\}$.

Our analog of embedding a model in a saturated model is to have interrelated functions ρ and h, a finite set A and an $a \notin A$ where:

- ρ and h are both approximations of models,
- h has to agree with ρ about quantifier-free statements about A,
- ρ has to agree with h about quantifier-free statements about $A \cup \{a\}$.

The two functions are interrelated, and we can think of them as competing: they're trying to force the other to either be inconsistent or non-convergent.

Theorem

Let T be a theory. Suppose that whenever we have h and ρ as above, if ρ converges for all quantifier-free formulas about A, h converges, and h is consistent, then ρ is consistent.

Then T has quantifier elimination.

Proof.

Let ρ the computable approximation of a saturated model of $T \cup \{\forall x \neg\phi(x, \vec{d})\}$ and h the computable approximation to a model of $T \cup \Sigma \cup \{\exists x \, \phi(x, \vec{d})\}$ where Σ is the quantifier-free statements about $\vec{d}$.

As soon as h enumerates $\phi(a, \vec{d})$, this formula gets copied into ρ, leading to a contradiction. Since ρ^h is inconsistent, h is inconsistent. □

So instead of working with saturated models, we can interpret model theoretic arguments as proofs of:

> *Suppose that we have an embedding of computable approximations h into ρ (as described above), if h is consistent then ρ^h is consistent.*

This is a perfectly decent arithmetic statement of the kind we can hope to proof mine.

Theorem

Suppose M is an algebraically closed field of characteristic 0, N is an ω-saturated algebraically closed field of characteristic 0, $A \subseteq M$ is finite, $f : A \to N$ is a local isomorphism, and $b \in M$. Then there is a homomorphism $g : A \cup \{b\} \to N$ extending f.

Theorem

Suppose M is an algebraically closed field of characteristic 0, N is an ω-saturated algebraically closed field of characteristic 0, $A \subseteq M$ is finite, $f : A \to N$ is a local isomorphism, and $b \in M$. Then there is a homomorphism $g : A \cup \{b\} \to N$ extending f.

Proof.

If b is in the field generated by A then $b = \frac{\sum_i q_i a_i}{\sum_i q'_i a'_i}$ where the q_i, q'_i are in $\mathbb{Q}$ and the a_i, a'_i are in A. Therefore we take $g(b) = \frac{\sum_i q_i f(a_i)}{\sum_i q'_i f(a'_i)}$.

□

Theorem

Suppose M is an algebraically closed field of characteristic 0, N is an ω-saturated algebraically closed field of characteristic 0, $A \subseteq M$ is finite, $f : A \to N$ is a local isomorphism, and $b \in M$. Then there is a homomorphism $g : A \cup \{b\} \to N$ extending f.

Proof.

If not, suppose there is a polynomial $p(x) = \sum_i c_i x^i$ where each c_i is a rational sum from A and $p(b) = 0$. We may choose p to have minimal degree, and then take $g(b)$ to be any root of $\sum_i f(c_i)x^i$. (It requires some non-trivial field theory to establish that this is really a valid choice.) $\square$

Theorem

Suppose M is an algebraically closed field of characteristic 0, N is an ω-saturated algebraically closed field of characteristic 0, $A \subseteq M$ is finite, $f : A \to N$ is a local isomorphism, and $b \in M$. Then there is a homomorphism $g : A \cup \{b\} \to N$ extending f.

Proof.

If not, b is transcendental over the field generated by A. Choose $g(b)$ to be any element transcendental over $g(A)$. □

So suppose we have a quantifier-free statement $\phi(x, \vec{y})$ and we wish to find a quantifier-free equivalent for $\exists x\, \phi(x, \vec{y})$.

We have an approximation h to a model of $\mathrm{ACF}_0 \cup \Sigma \cup \{\phi(a, \vec{d})\}$ which is trying to enumerate a consistent model. The model theoretic arguments usually start with cases where the witness is "close" to $\vec{y}$ and work towards cases where the witness is far away, but when we're trying to imagine how h behaves, it's more natural to work in the other direction.

So suppose we have a quantifier-free statement $\phi(x, \vec{y})$ and we wish to find a quantifier-free equivalent for $\exists x\, \phi(x, \vec{y})$.

We have an approximation h to a model of $\mathrm{ACF}_0 \cup \Sigma \cup \{\phi(a, \vec{d})\}$ which is trying to enumerate a consistent model. The model theoretic arguments usually start with cases where the witness is "close" to $\vec{y}$ and work towards cases where the witness is far away, but when we're trying to imagine how h behaves, it's more natural to work in the other direction.

Initially, we can imagine h trying to make a transcendental over $\vec{d}$; this is easy, since we just say that each non-trivial polynomial involving a is non-zero. The model theoretic argument must tell us how to enumerate formulas into Σ which contradict this: that is, we must discover that there are finitely many polynomials p_i and a quantifier-free formula ψ so that

$$\bigwedge_{i \leq n} p_i(a, \vec{d}) \neq 0 \to \phi(a, \vec{d}) \leftrightarrow \psi(\vec{d}).$$

This is typically what happens with proof mining. We start with a case split between a Σ_1 case and a Π_1 case. In this case the Σ_1 case is that there exists a polynomial $p(a, \vec{d}) = 0$, while the Π_1 case is that there is no such polynomial.

To get effective information out, we need to replace this with a split between the Σ_1 case happening *with a small witness* versus the Π_1 case "almost" happening.

This is typically what happens with proof mining. We start with a case split between a Σ_1 case and a Π_1 case. In this case the Σ_1 case is that there exists a polynomial $p(a, \vec{d}) = 0$, while the Π_1 case is that there is no such polynomial.

To get effective information out, we need to replace this with a split between the Σ_1 case happening *with a small witness* versus the Π_1 case "almost" happening.

To finish the proof of quantifier elimination, we then consider have to consider a disjunction of cases, one for each $p_i(a, \vec{d}) = 0$. In this case the model theory depends on some basic field theory which in turn depends on a lot of calculations involving things like polynomial division which show up in the quantifier elimination algorithm.

Van den Dries and Günaydin showed quantifier elimination-like results for algebraically or real closed fields "small" distinguished subgroups.

If K is a field of characteristic 0 and $G \subseteq K^\times$ is a multiplicative subgroup, G has *the Mann property* if each equation of the form

$$q_1 x_1 + \cdots + q_n x_n = 1$$

where the q_i are rational has only finitely many non-degenerate solutions in G. A solution $g_1, \ldots, g_n$ is degenerate if $\sum_{i \in I} q_i g_i = 0$ for some non-empty set I.

Van den Dries and Günaydin showed quantifier elimination-like results for algebraically or real closed fields "small" distinguished subgroups.

If K is a field of characteristic 0 and $G \subseteq K^\times$ is a multiplicative subgroup, G has *the Mann property* if each equation of the form

$$q_1 x_1 + \cdots + q_n x_n = 1$$

where the q_i are rational has only finitely many non-degenerate solutions in G. A solution $g_1, \ldots, g_n$ is degenerate if $\sum_{i \in I} q_i g_i = 0$ for some non-empty set I.

Motivating examples are $2^{\mathbb{Z}}$, $2^{\mathbb{Q}}$, $2^{\mathbb{Z}}3^{\mathbb{Z}}$ as subsets of $\mathbb{R}$.

Let Γ be a dense subgroup of $\mathbb{R}^{>0}$ with the Mann property. Work in the language of real closed fields together with:

- a predicate U (intended to name the dense subgroup),
- constant symbols for each element of Γ.

Let Γ be a dense subgroup of $\mathbb{R}^{>0}$ with the Mann property. Work in the language of real closed fields together with:

- a predicate U (intended to name the dense subgroup),
- constant symbols for each element of Γ.

Theorem (Van den Dries–Günaydin)

Any formula $\phi(\vec{x})$ in this language is equivalent (in the corresponding extension of RCF*) to a Boolean combination of formula of the form*

$$\exists \vec{y}(U(\vec{y}) \wedge \theta(\vec{y}) \wedge \psi(\vec{x}))$$

where:

- *θ is a formula where addition only appears in atomic formulas of the form $\sum_i t_i = \sum_i t'_i$,*
- *all quantifiers in θ are restricted to U,*
- *ψ is quantifier-free.*

Van den Dries and Günaydin prove quantifier elimination by a similar saturated embedding type argument: they show that if we have a finite partial homomorphism $f : (M, U) \to (N, V)$ with N ω-saturated and $a \in M$ and $U \equiv V$ and $a \in M$ then the homomorphism can be extended to include a.

Van den Dries and Günaydin prove quantifier elimination by a similar saturated embedding type argument: they show that if we have a finite partial homomorphism $f : (M, U) \to (N, V)$ with N ω-saturated and $a \in M$ and $U \equiv V$ and $a \in M$ then the homomorphism can be extended to include a.

As usual, the proof breaks into cases:

- if a is in U,
- if there is an elementary extension $U' \succ U$ so that a is in the real closure of $\mathrm{dom}(f) \cup U'$,
- if a is not in the real closure of any $\mathrm{dom}(f) \cup U'$.

Further, the proof actually depends on M itself being real closed.

These complications in the model theoretic proof lead to corresponding complications in the algorithm.

We start in the case where a is not in the real closure of any $\mathrm{dom}(f) \cup U'$. This means that our model h keeps enumerating sentences like $\forall \vec{u}\, U(\vec{u}) \to p(a, \vec{d}, \vec{u}) \neq 0$.

These complications in the model theoretic proof lead to corresponding complications in the algorithm.

We start in the case where a is not in the real closure of any $\mathrm{dom}(f) \cup U'$. This means that our model h keeps enumerating sentences like $\forall \vec{u}\, U(\vec{u}) \to p(a, \vec{d}, \vec{u}) \neq 0$. When we use the real closure of M, we end up adding formulas depending on other elements. This means we actually need to enumerate sentences like $\forall \vec{u}\, U(\vec{u}) \to \forall w\, q(\vec{d}, \vec{u}, w) = 0 \to p(a, \vec{d}, \vec{u}, w) \neq 0$, and similarly with sequences $w_1, \ldots, w_n$.

These complications in the model theoretic proof lead to corresponding complications in the algorithm.

We start in the case where a is not in the real closure of any $\mathrm{dom}(f) \cup U'$. This means that our model h keeps enumerating sentences like $\forall \vec{u}\, U(\vec{u}) \to p(a, \vec{d}, \vec{u}) \neq 0$. When we use the real closure of M, we end up adding formulas depending on other elements. This means we actually need to enumerate sentences like $\forall \vec{u}\, U(\vec{u}) \to \forall w\, q(\vec{d}, \vec{u}, w) = 0 \to p(a, \vec{d}, \vec{u}, w) \neq 0$, and similarly with sequences $w_1, \ldots, w_n$.

Evenutually the model theory tells us how to find a quantifier-free equivalent for ϕ under these assumptions. That is, we get an algorithm to find formulas so that

$$\forall \vec{u}, \vec{w}\, U(\vec{u}) \wedge \eta(\vec{d}, \vec{u}, \vec{w}) \to \exists a(\phi(a, \vec{d}) \wedge \bigwedge_i p_i(a, \vec{d}, \vec{u}, \vec{w}) \neq 0) \leftrightarrow \psi(\vec{x}, \vec{u}, \vec{w})$$

where η says that each element of $\vec{w}$ belongs to the real closure of $\vec{d}, \vec{u}$.

- We can reinterpret statements about uncountable sets as statements about functionals.

- We can reinterpret statements about uncountable sets as statements about functionals.
- We can use this, plus proof mining, to give a systematic method for turning model theoretic quantifier elimination proofs into algoirithms.

- We can reinterpret statements about uncountable sets as statements about functionals.
- We can use this, plus proof mining, to give a systematic method for turning model theoretic quantifier elimination proofs into algoirithms.
- We can use this in practice to produce new algorithms:
  - Forthcoming: quantifier elimination algorithm for real and algebraically closed fields with small subgroups.
  - Forthcoming, but more slowly: quantifier elimination algorithm for valued D-fields.

- We can reinterpret statements about uncountable sets as statements about functionals.
- We can use this, plus proof mining, to give a systematic method for turning model theoretic quantifier elimination proofs into algoirithms.
- We can use this in practice to produce new algorithms:
  - Forthcoming: quantifier elimination algorithm for real and algebraically closed fields with small subgroups.
  - Forthcoming, but more slowly: quantifier elimination algorithm for valued D-fields.

The end.