How to Automatically Prove Every First-Order Theorem of the Reals (The Gory Details)

Jason Rute

Graduate Student Seminar
Thursday, September 10, 2009

Propositional Logic

The language of proposition logic is composed of the following connectives (and their usual interpretations): $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), $\leftrightarrow$ (iff). The connectives connect propositional variables, $P, Q, R, \ldots$, which are usually interpreted to range over "true" and "false". Last, it's convenient to add the following constants: $\perp$ (false), $\top$ (true). If a sentence "makes sense" we call it *well-formed*. (The exact definition is not given.) Here are examples of well formed sentences in propositional logic that use all the connectives: $P \wedge (Q \vee R)$, $(P \rightarrow Q) \rightarrow (Q \rightarrow P)$, $\neg\perp \leftrightarrow \top$.

If we insert true and false (abbreviated T and F) for the variables then we can calculate the truth value of any sentence. For example, $\mathrm{T} \wedge (\mathrm{F} \vee \mathrm{T}) = \mathrm{T}$.

This gives, us an easy (although not necessarily fast) method to evaluate any propositional sentence to see if it is always true (a *tautology*), sometimes true, or never true. This method is called a *truth table*. Here is an example.

P	Q	$(P \rightarrow Q) \rightarrow (Q \rightarrow P)$
T	T	T
T	F	T
F	T	F
F	F	T

We say two sentences are *equivalent* if they have the same truth tables. Given any sentence, we can use the truth table to create an equivalent sentence that only contains the connectives $\wedge$, $\vee$, and $\neg$. For the example sentence $(P \rightarrow Q) \rightarrow (Q \rightarrow P)$, here is it's equivalent sentence and it's truth table.

P	Q	$(P \wedge Q) \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$
T	T	T
T	F	T
F	T	F
F	F	T

Notice this equivalent sentence is in a special form. It is a conjunction of disjunctions ($\bigvee \bigwedge$) and each disjunction is made up of variables or negations of variables. This special form is called *disjunctive normal form* (*DNF*) and will be important later.

Exercise 1. How does one get a disjunctive normal form sentence from a truth table.

The Language of First-Order Logic

Propositional logic isn't very expressive. We can expand the language of propositional logic by replacing the propositional variables by mathematical equations and relations. To do this we add a few symbols to our language. We always add the equal symbol, =. There are three more symbols we could add depending on our needs: functions, constants, and relations. A *function* takes inputs and has a value. In the real numbers $+$ and $\cdot$ are functions each taking two inputs. A *constant* is a special kind of function with no inputs. In the real numbers 0 and 1 are constants. A *relation* takes inputs like a function, but evaluates to true or false. For example, $<$ would be a relation taking two inputs. The symbol $=$ can also be viewed as special relation symbol as well.

Because we have values, it's also helpful to have variables to range over the possible values. We use the variables $x, y, z, \ldots$. But, for example, the equation $x = y$ doesn't make much sense without a *quantifier*. We use two quantifiers $\forall$ (for all) and $\exists$ (their exists).

An example well-formed sentence in a language with $+, \cdot$, 0, 1, $<$ would be $\forall x(x > 0 \rightarrow \exists y(y \cdot y = x))$.

Interpretations

You should recognize the above example as asserting the existence of a square root. The problem, however, is that I never told you how to understand $>$, $\cdot$, or 0. What if I was devious and intended $\cdot$ to represent addition? And did I even intend the variables to range over the real numbers? This sentence wouldn't be true for the rationals.

So we must give our symbols meaning and specify a domain we are working in. There are two ways to do this, *syntactically* and *semantically.*

The syntactic way is to just to do everything in terms of symbol manipulation. We specify a bunch of sentences that define the logical symbols, the quantifiers, and equals. These are called the *logical axioms.* Also there are rules on how to go from one set of sentences to another sentence. These are called *inference rules*. A *proof* is a trail from the logical axioms to a statement using inference rules. If Γ is a set of sentences (possibly infinite) and ϕ is a sentence, we write $\Gamma \vdash \phi$ to mean there is a proof of ϕ using Γ as a set of *assumptions*.

The semantic way is to talk about *structures*. A structure is exactly what you think it is (I hope). The real numbers and the natural numbers are structures, as well as things like groups, rings, vector spaces, and a set of two elements. Rather than using sentences to define functions and relations, we just *interpret* them. For example, we write $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$ to represent the structure of the real numbers where the symbols $0, 1, +, \cdot, <$ are interpreted as usual. Then the sentence $\phi = \forall x (x > 0 \rightarrow \exists y (x > y \wedge y \cdot y = x)$ would be true in the structure $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$. But it is not in $\langle \mathbb{N}, 0, 1, +, \cdot, < \rangle$. We say $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$ *satisfies* ϕ while $\langle \mathbb{N}, 0, 1, +, \cdot, < \rangle$ does not. Also we write $\Gamma \Vdash \phi$ to mean every structure that satisfies all the sentences in Γ also satisfies ϕ. For example, if Γ is the group axioms, then every group satisfies Γ and further if $\phi = \forall x \exists y (x \cdot y = x)$ then $\Gamma \Vdash \phi$, because ϕ is true of every group.

An important result is the following.

Theorem 2 (Gödel's Completeness Theorem, 1929)**.** *For any set of sentences Γ and any sentence ϕ, we have that $\Gamma \Vdash \phi$ iff $\Gamma \vdash \phi$.*

This implies that if a sentence ϕ is true of *every* structure satisfying Γ we can prove it.

Theories

We say a set of sentences Γ is a *theory* if it is closed under logical consequence. In other words, if we can prove it from Γ it is already in Γ. Theories generally arise in two ways. One way is to take a structure you know and love, for example, $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$, and you make let $\Gamma = Th(\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle) = \{\phi : \langle \mathbb{R}, 0, 1, +, \cdot, < \rangle \text{ satisfies } \phi\}$. In other words, Γ is every sentence true of $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$. In this method, we have a particular structure in mind. (Although it may turn out that there is another non-isomorphic structure that satisfies the same theory.)

The other way to create theories is to take a set of sentences Λ, and close them under logical deduction: $\Gamma = \{\phi : \Lambda \vdash \phi\}$. We call Λ an *axiomization* of Γ. As an example, if Λ were the group axioms, then Γ would be the theory of groups. In this method, we are using our theory to describe a class of structures. (Although, in some cases, it may turn out this class of structures is empty.)

We say a structure *models* a theory if it satisfies every sentence in it. So $\langle \mathbb{R}, 0, + \rangle$ would model the theory of groups.

There are four important properties of a theory.

1. A theory is *consistent* if it doesn't contain a set of contradictory sentences. Since a theory is closed under logical deduction, it's enough to check that it doesn't contain both ϕ and $\neg\phi$. The theory $Th(\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle)$ is

trivially consistent because it can only contain true sentences of the reals. By the Completeness Theorem (above), we know inconsistant theories have no models. The theory of groups is then consistent because we know there is a group (for example, $\langle \mathbb{R}, 0, +\rangle$).

2. A theory is *complete* if for every sentence ϕ, either ϕ is in the theory or $\neg\phi$ is. In other words, it's as big as it could be, but not necessarily inconsistent. The theory $Th\left(\langle \mathbb{R}, 0, 1, +, \cdot, <\rangle\right)$ is complete because it contains every true sentence. If ϕ is false, then $\neg\phi$ is true.

3. A theory is *finitely axiomatizable* if there is an finite axiomatization Λ of the theory. The theory of groups, of course, has a finite axiomatization, namely the group axioms. A slightly weaker condition is *computably axiomatizable* (also known as *recursively axiomatizable*). We say an axiomatization Λ is *computable* if there is a computer program that could check whether any sentence is an axiom or not. If you can describe the axiomatization, it is probably computable. (The property of being *axiomatizable* is trivial. Every theory Γ has an axiomatization, namely Γ.)

4. A theory is *decidable* if there is an algorithm (a computer program) that decides whether a given sentence is in the theory or not. Even if a theory is computably axiomatizable, it may not be decidable

These properties can combine non-trivially. For years, mathematicians had been trying to find a complete, consistent, decidable, computable axiomatization for $Th(\langle \mathbb{N}, 0, 1, +, \cdot, <\rangle)$. It turns out this is impossible.

Theorem 3 (Gödel's Incompleteness Theorem, 1931)**.** *Given a computably axiomatizable theory* Γ *that contains certain sentences that are true of* $\langle \mathbb{N}, 0, 1, +, \cdot, <\rangle$ *(basically enough to define addition, multiplication, and some induction) then* Γ *can not be both complete and consistent.*

Theorem 4 (Tarski's Undefinability of Truth, 1936; Turing's Halting Problem, 1937)**.** *Given a computably axiomatizable theory* Γ *as in the last theorem,* Γ *is not decidable.*

Real Closed Fields

Now, what about $Th(\langle \mathbb{R}, 0, 1, +, \cdot, <\rangle)$? Can it be computably axiomatized? Is it decidable? First, let us look at the typical axioms of the real numbers.

1. *Field axioms.* These say that $\langle \mathbb{R}, 0, 1, +, \cdot\rangle$ is a *field.* There are finitely many of them, and they are all first order sentences, for example $\forall x(x \neq 0 \rightarrow \exists y(x \cdot y = 1)$.

2. *Ordering axioms.* Together with the field axioms, these show that $\langle \mathbb{R}, 0, 1, +, \cdot, <\rangle$ is an *ordered field.* Again, there are finitely many and they are all first-order sentences.

3. *Completeness axiom.* This axiom says that for *any* set S, if S has an upper bound, then it has a least upper bound. This property is *not* a first order property. The problem is that first order sentences can only talk about points, not sets. The completeness axiom is a property of sets and therefore needs *second-order logic* to express it. (You could also, define the real number in set theory and keep using first-order logic, but that isn't our goal here.)

This brings me to my first important point:

First-order logic does not describe everything.

However, there are still partial versions of the completeness theorem that are describable in first order logic. If a set is definable in first-order logic, then we can express it's least upper bound in first order logic as well. For example,

$$\exists x(\forall y(y \cdot y < 1 + 1 \to y < x) \wedge \forall z(z < x \to \exists w(z < w < x \wedge w \cdot w < 1 + 1))).$$

This (in a round about way) asserts that 2 has a square root. (This is the typical way to prove it using the completeness axiom.) Of course, we could just write $\exists x(x \cdot x = 1 + 1)$, or even more generally $\forall x(x > 0 \to \exists y(y \cdot y = x)$.

This brings us to real closed fields. A *real closed field* (*RCF*) is a structure satisfying these axioms.

1. Field axioms.

2. Ordering axioms.

3. Square root axiom: Every positive number is the square of another.

4. Odd degree polynomial axiom: Every polynomial of odd degree has a root.

The first two sets are the same as for the real numbers. The third axiom is a single axiom in first order logic. The fourth however cannot be expressed as a single axiom, or even a finite set of axioms. But you can express it as an *axiom scheme,* which just means an infinite set of axioms that follow a pattern. For example here is axiom (3). (There are only odd number axioms.)

$$\forall a \forall b \forall c \exists x(ax^3 + bx^2 + cx = 0)$$

(Note, we dropped the "$\cdot$"s and abbreviated $x \cdot x \cdot x$ as x^3.)

The theory of real closed fields is computably axiomatizable since all the groups of axioms are either finite or follow a nice pattern. Also the theory of real closed fields is consistent, because there is at least one structure that models it, namely $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$.

There are however other useful RCFs. The smallest is the *real algebraic numbers.* (Note the similarity between algebraically closed fields and real closed fields.) Another is the *computable real numbers.* These are numbers whose decimal expansions are computable (except for the $.999\ldots$ / $.000\ldots$ issue). So

π and e are computable real numbers. The computable numbers are important to certain logicians. Also both sets are important to computer scientists; unlike all real numbers, these numbers can be calculated with a computer. (Who needs floating point numbers, when you can have the real thing?)

It's also possible to find real closed fields that are much larger than the set of reals. These are called the non-standard real numbers. A non-standard real closed field contains all the real numbers plus *infinite* and *infinitesimal* numbers. This allows one to do calculus without limits, and analysis without ϵ's and δ's. A notable amount of non-trivial work is done in analysis using non-standard real numbers. (See Terrence Tao's blog article "Ultrafilters, nonstandard analysis, and epsilon management". Disclaimer: Terrence Tao is not a logician.)

The Main Result

The question is how close is the theory of RCF to $Th(\langle\mathbb{R}, 0, 1, +, \cdot, <\rangle$. The answer is they are the same. So we have an complete, computable axiomatization of $Th(\langle\mathbb{R}, 0, 1, +, \cdot, <\rangle$. Here is the theorem.

Theorem 5 (Tarski, 1951)**.** *The theory of RCF is the same as* $Th(\langle\mathbb{R}, 0, 1, +, \cdot, <\rangle$. *It is therefore complete, consistent, and computably axiomatizable. Further, the theory of RCF is decidable, and the algorithm is known.*

This implies that the the real numbers have many nice properties with respect to first order logic. Also, it implies that any first order theorem of the real numbers has an elementary proof. It means that non-standard real numbers are easier to work with because they have *all* the same first-order properties as real numbers. It means there can be (and actually are) computer solvers for first-order problems in the real numbers. (They are slow, but can quickly handle certain classes of sentences.) In summary, I have my second important point:

> *It's important for a mathematician to know what a first order statement is, because first order logic has many nice properties.*

Quantifier Elimination

The proof of the main theorem is a bit long. It's not difficult. It just involves logic and calculus, and a tedious amount of bookkeeping.

All the logic comes at the beginning and generalizes to a number of similar problems. We will work out the main method, called quantifier elimination, and then prove similar but weaker results.

The method of *quantifier elimination* works basically as follows. For any sentence ϕ we can construct an equivalent sentence ψ without quantifiers. Since ψ has no quantifiers, it also has no variables. Therefore, it's just a matter of checking a sentence like $1 + 1 < 1 + 1 + 1 \wedge (1 + 1) \cdot (1 + 1) = 1 + 1 + 1 + 1$.

Also, if a theory admits quantifier elimination then it is complete.

Here are the details for quantifier elimination.[1] Assume we have a sentence ϕ we want to do quantifier elimination on. We proceed in steps as follows:

1. First, we move the quantifiers to the front. This is called *prenex normal form*. While, we are not going to go through the details, there are only a few cases to check, such as $(\theta \wedge \exists x \xi(x)) \leftrightarrow \exists x(\theta \wedge \xi(x))$. It's basically the same idea in programming that you can declare a variable at the beginning, even if you never use it. We now have a sentence in this form (or similar), where θ is quantifier free.

$$\forall x \exists y \forall z \theta(x, y, z)$$

2. Now we take the last quantifier. If it is a $\forall$, we convert it into a $\exists$ using the following logical identity, $\forall x \xi(x) \leftrightarrow \neg\exists x \neg\xi(x)$. (This is clear if you think about it.) This turns our example into

$$\forall x \exists y \neg\exists z \neg\theta(x, y, z).$$

3. We will attempt to eliminate every quantifier from the inside out. First we start with $\exists z$ in our example. To do this, we zoom in on that part of the formula and we treat the other variables like parameters (or constants). So our focus becomes

$$\exists z \neg\theta(a, b, z).$$

(I switched x to a, and y to b to emphasis they are parameters.)

4. We then convert the quantifier free part into disjunctive normal form (see Propositional Logic section). This is possible, since we have no quantifiers. Now our example is something like

$$\exists z \bigvee_n (\theta_n(a, b, z)$$

where we will zoom in on θ_n next.

5. We can move our $\exists z$ inside the $\bigvee$ since $\exists x(\xi(x) \vee \zeta(x)) \leftrightarrow \exists x \xi(x) \vee \exists x \zeta(x)$ is also a logical identity. (This is clear if you think about it.) Now our example becomes,

$$\bigvee_n \exists z \theta_n(a, b, z).$$

6. Further it is enough to eliminate each disjunct on it's own. So we are only concerned with

$$\exists z \theta_n(a, b, z)$$

where θ_n is a conjunction of atomic formulas and negations of an atomic formulas. An *atomic formula* is a formula that contains no logical symbols. The only atomic formulas are equations or relations.

[1] Note, that there are two approaches to quantifier elimination. The one here is the syntactic approach (symbol manipulation and formal proof). The other is the semantic approach, concerning the structures involved.

7. The remaining task is to find some $\psi(a,b)$ without quantifiers such that $\psi(a,b) \leftrightarrow \exists z \theta_n(a,b,z)$. This depends on the interpretation of the symbols, or on the axioms. In the case of real closed fields, it's a matter of calculus facts that can be proved in RCF and bookkeeping.

8. After eliminating z, we start our algorithm over and eliminate the next variable.

Notice that all the rearranging and recursion makes this method slow. (The proposition logic alone is at least NP-complete, and the whole thing can get up to PSPACE-complete (I believe).) However, if the formula is simple and the quantifiers small, it can be very practical for software applications (not so much for research math).

Dense Linear Orders Without Endpoints

A simpler example of quantifier elimination is $\langle \mathbb{R}, < \rangle$, in other words, the theory of the reals only using the relation $<$. We prove it starting at step (7) above. (Note quantifier elimination involves a lot of checking cases, and I may miss or skip a few.)

The task at hand is to eliminate the x in an existential formula made of disjuncts of atomic sentences and negations of atomic sentences. Again, we proceed by steps. But before we begin, lets set two examples to work on.

$$\exists x (x = b \wedge x < a \wedge b < x)$$

$$\exists x (a < x \wedge c < x \wedge x \neq b)$$

1. First, we eliminate the negations. This can be done, using the following identities of linear orders: $a \not< b \leftrightarrow (a = b \vee b < a)$ and $a \neq b \leftrightarrow (a < b \vee b < a)$. Our second example then becomes

$$\exists x (a < x \wedge c < x \wedge (x < b \vee b < x)).$$

   Of course this is no long a conjunction. Putting it back in DNF, and moving the $\exists x$ in we get two sub-goals.

$$\exists x (a < x \wedge c < x \wedge b < x)$$

$$\exists x (a < x \wedge c < x \wedge x < b).$$

2. Next, we get rid of any $x = \dots$ equations. This is simply done by substituting in whatever x equals for x. (There of course is the possible case that we have $x = x$, in which case we can replace it with $\top$ (true).) Our very first example becomes,

$$b < a \wedge b < b.$$

   Since we know longer have an x, we no longer need a quantifier. We've eliminated the quantifier!

3. Now we no longer have any equations containing x. But we still have inequalities containing x. In our two remaining examples, one has x bounded on both sides: $\exists x(a, c < x < b)$. Because $\mathbb{R}$ is a dense linear order, this is true iff $a, c < b$. Again, we've eliminated a quantifier! For the remaining case, $\exists x(a, b, c < x)$, it is trivially true because $\mathbb{R}$ has no endpoints. We just replace the formula with $\top$. No more quantifiers! (Also, if we run across $x < x$, we replace it with $\bot$ (false).)

4. Because there are no constants in $\langle \mathbb{R}, < \rangle$, the final quantifier free sentence will consist only of $\top$, $\bot$ and propositional connectives. This sentence is easily evaluated.

We just showed $Th(\langle \mathbb{R}, < \rangle)$ is decidable. Further, since the only facts we used are those of *dense linear orders without endpoints* (*DLOWOE*), we know that this holds for the theory of DLOWOE, which must be complete. Hence it and $Th(\langle \mathbb{R}, < \rangle)$ are the same theory, as is $Th(\langle \mathbb{Q}, < \rangle)$, since $\langle \mathbb{Q}, < \rangle$ is a DLOWOE. Since the theory of DLOWOE is finitely axiomatizable, then so is $Th(\langle \mathbb{R}, < \rangle)$.

Divisible Ordered Abelian Groups

We can easily extend the previous results to $\langle \mathbb{R}, 0, 1, +, < \rangle$. First, however, we'd like to extend the language without damaging anything. We can express any natural number n as $1 + \ldots + 1$ in our original language. We can also do the same with integer multiples of variables. nx becomes $x + \ldots + x$. Further, any equations containing rational numbers, rational multiples of variables, and addition can be expressed as follows. We combine the terms on each side, to get say, $\frac{m}{n} + \frac{a}{b}x = \frac{k}{l}x$ which is equivalent to $mbl + (nal)x = (nbk)x$. If any side is negative, we just add to both sides until positive. So without loss of generality, we can add rational constants and rational scalars of variables to our language.

Then we proceed as follows. For each inequality and equality (we can assume all negations are gone as before) we solve for x. This is where, having rationals is important. Now we have a conjunction of equalities and inequalities with x on one side, for example

$$\exists x(x = \frac{3}{4} \wedge x < \frac{a}{b} \wedge -1 < x).$$

We proceed exactly as before, eliminating x. But this time, at the end, we will have rational constants left in our quantifier free sentence. These however, are easy to evaluate.

We've proved the decidability of $Th(\langle \mathbb{R}, 0, 1, +, < \rangle)$. Also, as before, we really only used a small part of the theory, namely that of *divisible ordered Abelian groups*, of which $\langle \mathbb{Q}, 0, 1, +, < \rangle$ is the smallest, so $Th(\langle \mathbb{Q}, 0, 1, +, < \rangle$ is the same theory. (*Divisible* refers to each number being divisible by every positive integer n.) This is a computably axiomatizable theory and hence so is $Th(\langle \mathbb{R}, 0, 1, +, < \rangle) = Th(\langle \mathbb{Q}, 0, 1, +, < \rangle)$.

Other Similar Results

There are also other decidable theories of interest. For example, $Th(\langle \mathbb{C}, 0, 1, +, \cdot \rangle$ is decidable. It is also the theory of *algebraically closed fields of characteristic 0* (a field with all the integers where the all polynomials have roots), which is computably axiomatizable. The also implies the structure of *algebraically numbers* has the same theory. The proof of this result is done with quantifier elimination and the fact that every polynomial has a solution.

Also, the theory of addition (but not addition and multiplication) of the natural number and integers are both decidable, i.e. $Th(\langle \mathbb{N}, 0, 1, +, < \rangle$ and $Th(\langle \mathbb{Z}, 0, 1, +, < \rangle$. (Note, the two structure have different theories, as one doesn't have negatives.) The proof relies on quantifier elimination and the the Chinese Remainder Theorem.

Also, a recent theorem is that the first order theory (in a two sorted logic) of vector spaces with an inner product are decidable, but the theory of normed vector spaces is undecidable.

In general, for any class of yes/no mathematical questions, we can ask if there is an algorithm to decide if the answer is yes or no. The problem of finding the algorithm or showing no such algorithm exists is known as an *Entscheidungsproblem* (decision problem). There are many unsolved decision problems and they are an active area of interest in math and computer science.

References

- Harrison, J., *Handbook of Practical Logic and Automated Reasoning,* Cambridge University Press, 2009

  - This is a great book. Chapter 5 is all about various algorithms to solve problems including real closed fields, and the other ones I've mentioned. He even includes executable computer code (in OCaml) for each algorithm.

- Marker, D., *Model Theory : An Introduction*, Springer, 2002

  - This is a book also gives quantifier elimination proofs, but of a model-theoretic (or semantic) variety.

- Tao, T., "Ultrafilters, Nonstandard Analysis, and Epsilon Management", (http://terrytao.wordpress.com/2007/06/25/ultrafilters-nonstandard-analysis-and-epsilon-management/)

  - A good short exposition of non-standard analysis by a non-logician for non-logicians. I am very much a fan of Terry Tao and his blog.