Decision Procedures
1: Survey of decision procedures

John Harrison

Intel Corporation

TYPES summer school 2005, Göteborg

Fri 19th August 2005 (09:00 – 09:45)

Summary

- Interesting and uninteresting proofs
- Theory and practice
- Beyond our scope
- Logic and theories
- Pure logic
- Decidable theories

Interesting and uninteresting proofs

Much of this summer school emphasizes how interesting and useful proofs themselves are. But they aren't always!

$$\begin{aligned}
&6(x_1^2 + x_2^2 + x_3^2 + x_4^2)^2 = \\
&(x_1 + x_2)^4 + (x_1 + x_3)^4 + (x_1 + x_4)^4 + \\
&(x_2 + x_3)^4 + (x_2 + x_4)^4 + (x_3 + x_4)^4 + \\
&(x_1 - x_2)^4 + (x_1 - x_3)^4 + (x_1 - x_4)^4 + \\
&(x_2 - x_3)^4 + (x_2 - x_4)^4 + (x_3 - x_4)^4
\end{aligned}$$

We'd like to concentrate on interesting parts, automating parts with

- No interesting computational content
- No intellectual interest in the proof method

Theory and practice

We may ask what problems are decidable

- In principle
- In a feasible time bound
- On real problems of interest

Not always the same! Consider propositional logic.

- Trivial
- Infeasible
- Very useful

What we’ll cover

We’ll consider only theories in classical first-order logic.

- Key decidability results for first order theories
- Focus on pure logic and arithmetical theories

What we won’t cover

We miss out several key related areas:

- Decision procedures for constructive/intuitionistic theories
- Decision procedures for fragments of higher-order logic
- Decision procedures for modal or other nonclassical logics.

For example:

- First-order validity semidecidable, but higher-order validity subsumes arithmetic truth, so not even semidecidable
- Example: first order theories of real and algebraically closed fields are decidable classically (Tarski 1930) but not intuitionistically (Gabbay 1973).

First-order logic

English	Standard	Other
false	$\bot$	0, F
true	$\top$	1, T
not p	$\neg p$	$\overline{p}$, $-p$, $\sim p$
p and q	$p \land q$	pq, $p\&q$, $p \cdot q$
p or q	$p \lor q$	$p + q$, $p \mid q$, $p\ or\ q$
p implies q	$p \Rightarrow q$	$p \leq q$, $p \rightarrow q$, $p \supset q$
p iff q	$p \Leftrightarrow q$	$p = q$, $p \equiv q$, $p \sim q$
For all x, p	$\forall x.\ p$	$(x)p$, Axp
Exists x s.t. p	$\exists x.\ p$	$(\exists x.\ )p$, Exp

Semantics

Key semantic notion is $A \models p$: in any model where all formulas in A hold, then p holds.

Crucial distinction between

- Logical validity — holds whatever the interpretation of symbols
- Truth in a particular theory

For example, $x + y = y + x$ holds in most arithmetical models, but not for *any* interpretation of '$+$', so $\not\models x + y = y + x$.

Theories

A theory is a set of formulas T closed under logical validity, i.e. $T \models p$ iff $p \in T$. A theory T is:

- *Consistent* if we never have $p \in T$ and $(\neg p) \in T$.
- *Complete* if for closed p we have $p \in T$ or $(\neg p) \in T$.
- *Decidable* if there's an algorithm to tell us whether a given closed p is in T

Note that a complete theory generated by an r.e. axiom set is also decidable.

Pure first-order logic

Not decidable but at least *semidecidable*: there is a complete proof search procedure to decide if $\models p$ for any given p.

- Can search for proofs in any of the standard calculi
- Tends to be easier using ‘cut-free’ systems like sequent calculus
- More convenient, though not necessary, to Skolemize first.
- Exploit unification to instantiate intelligently

A significant distinction

A significant characteristic is whether unifiers are global, applying everywhere, or just local:

- Top-down, global methods (tableaux, model elimination)
- Bottom-up, local methods (resolution, inverse method)

These proof methods tend to have corresponding characteristics.

Decidable problems

Although first order validity is undecidable, there are special cases where it is decidable, e.g.

- AE formulas: no function symbols, universal quantifiers before existentials in prenex form (so finite Herbrand base).
- Monadic formulas: no function symbols, only unary predicates

These are not particularly useful in practice, though they can be used to automate syllogistic reasoning.

> If all M are P, and all S are M, then all S are P

can be expressed as the monadic formula:

$$(\forall x.\ M(x) \Rightarrow P(x)) \wedge (\forall x.\ S(x) \Rightarrow M(x)) \Rightarrow (\forall x.\ S(x) \Rightarrow P(x))$$

The theory of equality

A simple but useful decidable theory is the universal theory of equality with function symbols, e.g.

$$\forall x.\ f(f(f(x)) = x \land f(f(f(f(f(x))))) = x \Rightarrow f(x) = x$$

after negating and Skolemizing we need to test a ground formula for satisfiability:

$$f(f(f(c)) = c \land f(f(f(f(f(c))))) = c \land \neg(f(c) = c)$$

Two well-known algorithms:

- Put the formula in DNF and test each disjunct using one of the classic ‘congruence closure’ algorithms.
- Reduce to SAT by introducing a propositional variable for each equation between subterms and adding constraints.

Decidable theories

More useful in practical applications are cases not of *pure* validity, but validity in special (classes of) models, or consequence from useful axioms, e.g.

- Does a formula hold over all rings (Boolean rings, non-nilpotent rings, integral domains, fields, algebraically closed fields, . . .)
- Does a formula hold in the natural numbers or the integers?
- Does a formula hold over the real numbers?
- Does a formula hold in all real-closed fields?
- . . .

Because arithmetic comes up in practice all the time, there's particular interest in theories of arithmetic.

Quantifier elimination

Often, a quantified formula is T-equivalent to a quantifier-free one:

- $\mathbb{C} \models (\exists x.\ x^2 + 1 = 0) \Leftrightarrow \top$
- $\mathbb{R} \models (\exists x. ax^2{+}bx{+}c = 0) \Leftrightarrow a \neq 0 {\wedge} b^2 \geq 4ac {\vee} a = 0 {\wedge} (b \neq 0 {\vee} c = 0)$
- $\mathbb{Q} \models (\forall x.\ x < a \Rightarrow x < b) \Leftrightarrow a \leq b$
- $\mathbb{Z} \models (\exists k\ x\ y.\ ax = (5k + 2)y + 1) \Leftrightarrow \neg(a = 0)$

We say a theory T admits *quantifier elimination* if *every* formula has this property.

Assuming we can decide variable-free formulas, quantifier elimination implies completeness.

And then an *algorithm* for quantifier elimination gives a decision method.

Important arithmetical examples

- Presburger arithmetic: arithmetic equations and inequalities with addition but *not multiplication*, interpreted over $\mathbb{Z}$ or $\mathbb{N}$.
- Tarski arithmetic: arithmetic equations and inequalities with addition and multiplication, interpreted over $\mathbb{R}$ (or any real-closed field)
- General algebra: arithmetic equations with addition and multiplication interpreted over $\mathbb{C}$ (or other algebraically closed field).

However, arithmetic with multiplication over $\mathbb{Z}$ is not even semidecidable, by Gödel's theorem.

Nor is arithmetic over $\mathbb{Q}$ (Julia Robinson), nor just solvability of equations over $\mathbb{Z}$ (Matiyasevich). Equations over $\mathbb{Q}$ unknown.

Pick ’n mix

There are some known cases of quantifier elimination for combined theories

- BAPA — Boolean algebra of finite sets plus Presburger arithmetic (Feferman/Vaught, Kuncac/Nguyen/Rinard)
- Mixed real-integer linear arithmetic with floor function (Weispfenning)

In lecture 3 we’ll examine more systemtic and modular ways of combining theories.

Summary

- We’d like to be able to automate boring routine proofs
- Well-established repertoire of decidable theories
- Theory/practice distinction can make a dramatic difference
- Many decision methods are based on more general quantifier elimination
- It is possible, but not routine, to find decidable mixtures.

Decision Procedures
2: Real quantifier elimination

John Harrison

Intel Corporation

TYPES summer school 2005, Göteborg

Fri 19th August 2005 (09:55 – 10:40)

Summary

- What we’ll prove
- History
- Sign matrices
- The key recursion
- Parametrization
- Real-closed fields

What we'll prove

Take a first-order language:

- All rational constants p/q
- Operators of negation, addition, subtraction and multiplication
- Relations '$=$', '$<$', '$\leq$', '$>$', '$\geq$'

We'll prove that every formula in the language has a quantifier-free equivalent, and will give a systematic algorithm for finding it.

Applications

In principle, this method can be used to solve many non-trivial problems.

> Kissing problem: how many disjoint n-dimensional spheres can be packed into space so that they touch a given unit sphere?

Pretty much *any* geometrical assertion can be expressed in this theory.

If theorem holds for *complex* values of the coordinates, and then simpler methods are available (Gröbner bases, Wu-Ritt triangulation. . .).

History

- 1930: Tarski discovers quantifier elimination procedure for this theory.
- 1948: Tarski's algorithm published by RAND
- 1954: Seidenberg publishes simpler algorithm
- 1975: Collins develops and *implements* cylindrical algebraic decomposition (CAD) algorithm
- 1983: Hörmander publishes very simple algorithm based on ideas by Cohen.
- 1990: Vorobjov improves complexity bound to doubly exponential in number of quantifier *alternations*.

We'll present the Cohen-Hörmander algorithm.

Current implementations

There are quite a few simple versions of real quantifier elimination, even in computer algebra systems like Mathematica.

Among the more heavyweight implementations are:

- `qepcad` — `http://www.cs.usna.edu/~qepcad/B/QEPCAD.html`
- `REDLOG` — `http://www.fmi.uni-passau.de/~redlog/`

One quantifier at a time

For a general quantifier elimination procedure, we just need one for a formula

$$\exists x.\ P[a_1, \ldots, a_n, x]$$

where $P[a_1, \ldots, a_n, x]$ involves no other quantifiers but may involve other variables.

Then we can apply the procedure successively inside to outside, dealing with universal quantifiers via $(\forall x.\ P[x]) \Leftrightarrow (\neg \exists x.\ \neg P[x])$.

Forget parametrization for now

First we’ll ignore the fact that the polynomials contain variables other than the one being eliminated.

This keeps the technicalities a bit simpler and shows the main ideas clearly.

The generalization to the parametrized case will then be very easy:

- Replace polynomial division by pseudo-division
- Perform case-splits to determine signs of coefficients

Sign matrices

Take a set of univariate polynomials $p_1(x)$, . . . , $p_n(x)$.

A *sign matrix* for those polynomials is a division of the real line into alternating points and intervals:

$$(-\infty, x_1), x_1, (x_1, x_2), x_2, \ldots, x_{m-1}, (x_{m-1}, x_m), x_m, (x_m, +\infty)$$

and a matrix giving the sign of each polynomial on each interval:

- Positive ($+$)
- Negative ($-$)
- Zero (0)

Sign matrix example

The polynomials $p_1(x) = x^2 - 3x + 2$ and $p_2(x) = 2x - 3$ have the following sign matrix:

Point/Interval	p_1	p_2
$(-\infty, x_1)$	$+$	$-$
x_1	0	$-$
(x_1, x_2)	$-$	$-$
x_2	$-$	0
(x_2, x_3)	$-$	$+$
x_3	0	$+$
$(x_3, +\infty)$	$+$	$+$

Using the sign matrix

Using the sign matrix for all polynomials appearing in $P[x]$ we can answer any quantifier elimination problem: $\exists x.\ P[x]$

- Look to see if any row of the matrix satisfies the formula (hence dealing with existential)
- For each row, just see if the corresponding set of signs satisfies the formula.

We have replaced the quantifier elimination problem with sign matrix determination

Finding the sign matrix

For constant polynomials, the sign matrix is trivial (2 has sign '$+$' etc.)

To find a sign matrix for $p, p_1, \ldots, p_n$ it suffices to find one for $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$, where

- $p_0 \equiv p'$ is the derivative of p
- $r_i = \textsf{rem}(p, p_i)$

(Remaindering means we have some q_i so $p = q_i \cdot p_i + r_i$.)

Taking p to be the polynomial of highest degree we get a simple recursive algorithm for sign matrix determination.

Details of recursive step

So, suppose we have a sign matrix for $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$.

We need to construct a sign matrix for $p, p_1, \ldots, p_n$.

- May need to add more points and hence intervals for roots of p
- Need to determine signs of $p_1, \ldots, p_n$ at the new points and intervals
- Need the sign of p itself everywhere.

Step 1

Split the given sign matrix into two parts, but keep all the points for now:

- M for $p', p_1, \ldots, p_n$
- M' for $r_0, r_1, \ldots, r_n$

We can infer the sign of p at all the ‘significant’ *points* of M as follows:

$$p = q_i p_i + r_i$$

and for each of our points, one of the p_i is zero, so $p = r_i$ there and we can read off p’s sign from r_i’s.

Step 2

Now we’re done with M' and we can throw it away.

We also ‘condense’ M by eliminating points that are not roots of one of the $p', p_1, \ldots, p_n$.

Note that the sign of any of these polynomials is stable on the condensed intervals, since they have no roots there.

- We know the sign of p at all the points of this matrix.
- However, p itself may have additional roots, and we don’t know anything about the intervals yet.

Step 3

There can be at most one root of p in each of the existing intervals, because otherwise p' would have a root there.

We can tell whether there is a root by checking the signs of p (determined in Step 1) at the two endpoints of the interval.

Insert a new point precisely if p has strictly opposite signs at the two endpoints (simple variant for the two end intervals).

None of the other polynomials change sign over the original interval, so just copy the values to the point and subintervals.

Throw away p' and we're done!

Multivariate generalization

In the multivariate context, we can't simply divide polynomials. Instead of

$$p = p_i \cdot q_i + r_i$$

we get

$$a^k p = p_i \cdot q_i + r_i$$

where a is the leading coefficient of p_i.

The same logic works, but we need case splits to fix the sign of a.

Real-closed fields

With more effort, all the ‘analytical’ facts can be deduced from the axioms for *real-closed fields*.

- Usual ordered field axioms
- Existence of square roots: $\forall x.\ x \geq 0 \Rightarrow \exists y.\ x = y^2$
- Solvability of odd-degree equations:
  $\forall a_0, \ldots, a_n.\ a_n \neq 0 \Rightarrow \exists x.\ a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$

Examples include computable reals and algebraic reals. So this already gives a complete theory, without a stronger completeness axiom.

Summary

- Real quantifier elimination one of the most significant logical decidability results known.
- Original result due to Tarski, for general real closed fields.
- A half-century of research has resulted in simpler and more efficient algorithms (not always at the same time).
- The Cohen-Hörmander algorithm is remarkably simple (relatively speaking).
- The complexity, both theoretical and practical, is still bad, so there’s limited success on non-trivial problems.

Decision Procedures
3: Combination and certification of decision procedures

John Harrison

Intel Corporation

TYPES summer school 2005, Göteborg

Sat 20th August 2005 (12:05 – 12:50)

Summary

- Need to combine multiple decision procedures
- Basics of Nelson-Oppen method
- Proof-producing decision procedures
- Separate certification
- LCF-style implementation and reflection

Need for combinations

In applications we often need to combine decision methods from different domains.

$$x - 1 < n \wedge \neg(x < n) \Rightarrow a[x] = a[n]$$

An arithmetic decision procedure could easily prove

$$x - 1 < n \wedge \neg(x < n) \Rightarrow x = n$$

but could not make the additional final step, even though it looks trivial.

Most combinations are undecidable

Adding almost any additions, especially uninterpreted, to the usual decidable arithmetic theories destroys decidability.

Some exceptions like BAPA (‘Boolean algebra + Presburger arithmetic’).

This formula over the reals constrains P to define the integers:

$$(\forall n.\ P(n+1) \Leftrightarrow P(n)) \wedge (\forall n.\ 0 \leq n \wedge n < 1 \Rightarrow (P(n) \Leftrightarrow n = 0))$$

and this one in Presburger arithmetic defines squaring:

$$\begin{aligned}&(\forall n.\ f(-n) = f(n)) \wedge (f(0) = 0) \wedge \\ &(\forall n.\ 0 \leq n \Rightarrow f(n+1) = f(n) + n + n + 1)\end{aligned}$$

and so we can define multiplication.

Quantifier-free theories

However, if we stick to so-called ‘quantifier-free’ theories, i.e. deciding universal formulas, things are better.

Two well-known methods for combining such decision procedures:

- Nelson-Oppen
- Shostak

Nelson-Oppen is more general and conceptually simpler.

Shostak seems more efficient where it does work, and only recently has it really been understood.

Nelson-Oppen basics

Key idea is to combine theories $T_1, \ldots, T_n$ with *disjoint signatures*. For instance

- T_1: numerical constants, arithmetic operations
- T_2: list operations like cons, head and tail.
- T_3: other uninterpreted function symbols.

The only common function or relation symbol is '$=$'.

This means that we only need to share formulas built from equations among the component decision procedure, thanks to the *Craig interpolation theorem*.

The interpolation theorem

Several slightly different forms; we'll use this one (by compactness, generalizes to theories):

> If $\models \phi_1 \wedge \phi_2 \Rightarrow \bot$ then there is an 'interpolant' ψ, whose only free variables and function and predicate symbols are those occurring in *both* ϕ_1 and ϕ_2, such that $\models \phi_1 \Rightarrow \psi$ and $\models \phi_2 \Rightarrow \neg\psi$.

This is used to assure us that the Nelson-Oppen method is complete, though we don't need to produce general interpolants in the method.

In fact, interpolants can be found quite easily from proofs, including Herbrand-type proofs produced by resolution etc.

Nelson-Oppen I

Proof by example: refute the following formula in a mixture of Presburger arithmetic and uninterpreted functions:

$$f(v-1)-1 = v+1 \land f(u)+1 = u-1 \land u+1 = v$$

First step is to *homogenize*, i.e. get rid of atomic formulas involving a mix of signatures:

$$u+1 = v \land v_1+1 = u-1 \land v_2-1 = v+1 \land v_2 = f(v_3) \land v_1 = f(u) \land v_3 = v-1$$

so now we can split the conjuncts according to signature:

$$(u+1 = v \land v_1+1 = u-1 \land v_2-1 = v+1 \land v_3 = v-1)\land$$
$$(v_2 = f(v_3) \land v_1 = f(u))$$

Nelson-Oppen II

If the entire formula is contradictory, then there's an interpolant ψ such that in Presburger arithmetic:

$$\mathbb{Z} \models u + 1 = v \land v_1 + 1 = u - 1 \land v_2 - 1 = v + 1 \land v_3 = v - 1 \Rightarrow \psi$$

and in pure logic:

$$\models v_2 = f(v_3) \land v_1 = f(u) \land \psi \Rightarrow \bot$$

We can assume it only involves variables and equality, by the interpolant property and disjointness of signatures.

Subject to a technical condition about finite models, the pure equality theory admits quantifier elimination.

So we can assume ψ is a propositional combination of equations between variables.

Nelson-Oppen III

In our running example, $u = v_3 \land \neg(v_1 = v_2)$ is one suitable interpolant, so

$$\mathbb{Z} \models u + 1 = v \land v_1 + 1 = u - 1 \land v_2 - 1 = v + 1 \land v_3 = v - 1 \Rightarrow u = v_3 \land \neg(v_1 = v_2)$$

in Presburger arithmetic, and in pure logic:

$$\models v_2 = f(v_3) \land v_1 = f(u) \Rightarrow u = v_3 \land \neg(v_1 = v_2) \Rightarrow \bot$$

The component decision procedures can deal with those, and the result is proved.

Nelson-Oppen IV

Could enumerate all significanctly different potential interpolants.

Better: case-split the original problem over all possible equivalence relations between the variables (5 in our example).

$$T_1, \ldots, T_n \models \phi_1 \wedge \cdots \wedge \phi_n \wedge ar(P) \Rightarrow \bot$$

So by interpolation there's a C with

$$T_1 \models \phi_1 \wedge ar(P) \Rightarrow C$$
$$T_2, \ldots, T_n \models \phi_2 \wedge \cdots \wedge \phi_n \wedge ar(P) \Rightarrow \neg C$$

Since $ar(P) \Rightarrow C$ or $ar(P) \Rightarrow \neg C$, we must have one theory with $T_i \models \phi_i \wedge ar(P) \Rightarrow \bot$.

Nelson-Oppen V

Still, there are quite a lot of possible equivalence relations ($\text{bell}(5) = 52$), leading to large case-splits.

An alternative formulation is to repeatedly let each theory deduce new disjunctions of equations, and case-split over them.

$$T_i \models \phi_i \Rightarrow x_1 = y_1 \vee \cdots \vee x_n = y_n$$

This allows two imporant optimizations:

- If theories are *convex*, need only consider pure equations, no disjunctions.
- Component procedures can actually produce equational consequences rather than waiting passively for formulas to test.

Shostak’s method

Can be seen as an optimization of Nelson-Oppen method for common special cases. Instead of just a decision method each component theory has a

- Canonizer — puts a term in a T-canonical form
- Solver — solves systems of equations

Shostak’s original procedure worked well, but the theory was flawed on many levels. In general his procedure was incomplete and potentially nonterminating.

It’s only recently that a full understanding has (apparently) been reached.

See ICS (`http://www.icansolve.com`) for one implementation.

Certification of decision procedures

We might want a decision procedure to produce a ‘proof’ or ‘certificate’

- Doubts over the correctness of the core decision method
- Desire to use the proof in other contexts

This arises in at least two real cases:

- Fully expansive (e.g. ‘LCF-style’) theorem proving.
- Proof-carrying code

Certifiable and non-certifiable

The most desirable situation is that a decision procedure should produce a short certificate that can be checked easily.

Factorization and primality is a good example:

- Certificate that a number is not prime: the factors! (Others are also possible.)
- Certificate that a number is prime: Pratt, Pocklington, Pomerance, . . .

This means that primality checking is in NP $\cap$ co-NP (we now know it's in P).

Certifying universal formulas over $\mathbb{C}$

Use the (weak) *Hilbert Nullstellensatz*:

The polynomial equations $p_1(x_1, \ldots, x_n) = 0$, . . . , $p_k(x_1, \ldots, x_n) = 0$ in an algebraically closed field have *no* common solution iff there are polynomials $q_1(x_1, \ldots, x_n)$, . . . , $q_k(x_1, \ldots, x_n)$ such that the following polynomial identity holds:

$$q_1(x_1, \ldots, x_n) \cdot p_1(x_1, \ldots, x_n) + \cdots + q_k(x_1, \ldots, x_n) \cdot p_k(x_1, \ldots, x_n) = 1$$

All we need to certify the result is the cofactors $q_i(x_1, \ldots, x_n)$, which we can find by an instrumented Gröbner basis algorithm.

The checking process involves just algebraic normalization (maybe still not totally trivial. . .)

Certifying universal formulas over $\mathbb{R}$

There is a similar but more complicated Nullstellensatz (and Positivstellensatz) over $\mathbb{R}$.

The general form is similar, but it's more complicated because of all the different orderings.

It inherently involves sums of squares (SOS), and the certificates can be found efficiently using semidefinite programming (Parillo . . .)

Example: easy to check

$$\forall a\ b\ c\ x.\ ax^2 + bx + c = 0 \Rightarrow b^2 - 4ac \geq 0$$

via the following SOS certificate:

$$b^2 - 4ac = (2ax + b)^2 - 4a(ax^2 + bx + c)$$

Less favourable cases

Unfortunately not all decision procedures seem to admit a nice separation of proof from checking.

Then if a proof is required, there seems no significantly easier way than generating proofs along each step of the algorithm.

Example: Cohen-Hörmander algorithm implemented in HOL Light by McLaughlin (CADE 2005).

Works well, useful for small problems, but about $1000\times$ slowdown relative to non-proof-producing implementation.

Summary

- There is a need for combinations of decision methods
- For general quantifier prefixes, relatively few useful results.
- Nelson-Oppen and Shostak give useful methods for universal formulas.
- We sometimes also want decision procedures to produce proofs
- Some procedures admit efficient separation of search and checking, others do not.
- Interesting research topic: new ways of compactly certifying decision methods.