On Satisfiability of Polynomial Equations over Large (Finite) Prime Fields

Lucas Clement Vella
Leonardo Alt

Motivation: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARKs)

ZK program in ZoKrates:

```
def main(private field input)
    -> field
{
    field hash =
      /* some complicated hash
         function on "input" */;
    return hash;
}
```

Given this ZK program and a hash value, using zkSNARKs ***I can prove that I know some input that generates the hash value, without disclosing any information about such input***.

Example Arithmetization of a Trivial Program

```
def main(private u8 a):
    assert(a != 0)
    return
```

$$x3^2 - x3 = 0$$

$$x4^2 - x4 = 0$$

$$x5^2 - x5 = 0$$

$$x6^2 - x6 = 0$$

$$x7^2 - x7 = 0$$

$$x8^2 - x8 = 0$$

$$x9^2 - x9 = 0$$

$$x10^2 - x10 = 0$$

$$x10 + 2*x9 + 4*x8 + 8*x7 + 16*x6 + 32*x5 + 64*x4 + 128*x3 - a = 0$$

$$a*x0 - x1 = 0$$

$$-a*x1 + a = 0$$

$$-x1 + 1 = 0$$

An SMT Theory for Prime Field Polynomials

A conjunction of literals

$$
\begin{aligned}
(\quad & 0 = 4x^3 + 2x^2y^2 + 12x - y - 5 \\
\wedge \quad & 0 = 7xyz - 12 \\
\wedge \quad & 0 \neq \mathbf{y^3 - 3x^2 - 7xy^3 - 1} \\
\wedge \quad & 0 \neq \mathbf{x - y} \quad)
\end{aligned}
$$

Is equivalent to a system of polynomial equations:

$$
\left\{
\begin{aligned}
4x^3 + 2x^2y^2 + 12x - y - 5 &= 0 \\
7xyz - 12 &= 0 \\
w(\mathbf{y^3 - 3x^2 - 7xy^3 - 1})(\mathbf{x - y}) - 1 &= 0
\end{aligned}
\right.
$$

By introducing a new variable “w” we can turn a set of negatives into a positive, which then we can handle with algebraic tools.

Get easy cases out of the way

- Handle empty set and constant polynomials
- Check if (0, 0, 0, …) is a solution (just look if there is a constant term)
- Handle linear case
- Handle single variable case
  - Extract a root with Cantor-Zassenhaus algorithm
- Maybe handle single polynomial case?
  - Factorize and handle each factor independently, possibly recursively
  - If absolutely irreducible, it is easy to find a solution via algebraic geometry
  - If not, $\{f = 0 \wedge df/dx_i = 0\}$ have the same roots, can recurse

Fast Algebraic Geometry Method

- If ideal $\langle f_1, f_2, f_3 \ldots \rangle$ is prime, defines an absolutely irreducible variety, and some other hard to test properties, there is a probabilistic polynomial algorithm to find a common root.
  - *"Fast computation of a rational point of a variety over a finite field"* A. Cafure & G. Matera
- We don't know what happens if properties are not met.
- We don't know if instances of our problem usually meet those properties.
- The algorithm does not help with the general case.

Robust Algebraic Geometry Method

- If $p > d^{n^n}$, there is an algorithm that decides if polynomial system has solution.
  - *"Solvability of systems of polynomial congruences modulo a large prime"* Huang & Wong
- Field size of $\sim 2^{254}$ is too low for the algorithm to be useful.
- Worst case complexity of $O(d^{n^n})$
  - Does not seems to be practical
  - Actually has this explicit triple `for` in the algorithm:
    - `for s in 0 to n:`
      - `for i in 0 to O(d`$^{s-1}$`):`
        - `for j in 0 to s:`
          - …
  - On the bright side, is parallelizable.

Methods based on Gröbner Basis

- The Gröbner Basis of $\{f_1, f_2, \dots\}$ is an "equivalent" set $\{g_1, g_2, \dots\}$
  - They have the same set of zeros ($f_1 = 0 \land f2 = 0 \land \dots \leftrightarrow g_1 = 0 \land g_2 = 0 \land \dots$)
  - Makes evident many algebraic properties of the system
  - Immediately tells if the system has any solutions over the algebraic closure of the field:
    - If some $g_i = 1$, the system has no solutions.
    - If no $g_i = 1$, the system maybe (probably?) has solutions
- Calculating it is EXPSPACE-hard
  - Complexity depends on the choice of total ordering among the monomials
  - Often works in practice
    - It took a little more than 1s to compute the GB of the trivial program shown before

Triangular System

- Gröbner Basis is the first step to place the system in triangular form
  - Or the only step, depending on the order you use
  - Univariate polynomial can be solved one at a time
    - Cantor-Zassenhaus
- Once in triangular form, literature considers the problem solved!
  - Not quite!
  - Univariate polynomial might not have roots in GF(p)
  - Might have to backtrack
  - Feels like a typical NP-complete problem
- I may be missing some fundamental algebraic tool here, must investigate

Example over $\mathbb{C}$

Original:

- $x^2 + y + z - 1$
- $x + y^2 + z - 1$
- $x + y + z^2 - 1$
- $x^2 + y^2 + z^2 - 1$

Reduced Gröbner Basis in lexicographical ordering:

- $z^2 - z$
- $2yz + z^4 + z^2 - 2z$
- $y^2 - y - z^2 + z$
- $x + y + z^2 - 1$

Primary decomposition

- Triangular system is the first step for prime decomposition of the problem
  - "Every ideal is a unique intersection of primary ideals" (all commutative algebra books)
- If any one of them has a solution, it is a solution to the original problem
  - Each primary ideal of the decomposition can be considered independently, in parallel
- If any one of them happens to define an absolutely irreducible variety, a solution can be found using *"Fast Algebraic Geometry Method"*
  - The problem remains if none of them happens to be.
  - What to do next? Proceed to the general method of NP-complete search?

Ground field restriction

- By Fermat’s Little Theorem, $x^p - x = 0$
- If you include polynomials $x_i^p - x_i$ for every variable x_i in the Gröbner Basis computation, you remove all roots outside of the ground field.
- A plain Gröbner Basis computation will immediately decide if the system has solutions or not.
- It will also will never end before the age of mankind.
  - For large p, polynomials will grow proportionally large.
- Not sure if this idea can be salvaged.

Local Search Method

- If there are a large number of solution, this might work
- Not sure what kind of algorithm to use
  - I don’t see a clear generalization from local search SAT solvers
  - I don’t see a clear generalization from minimization/optimization methods
- Can’t prove unsolvability
  - Unless you exhaustively test every possible solution

Model Constructing Satisfiability Calculus

- Thomas Hader just explained it less than 20 min ago
- The technique seems very promising for very large prime fields, but:
  - Must not use field polynomials $x^p - x = 0$
  - Must figure out some other way to deal with roots outside of the field

Conclusions

- Field polynomials (x^p - x) are the bane of large prime fields
  - They prevent the generalization of techniques that works for small prime fields.
- Most UNSAT cases can be (hopefully) handled by plain Gröbner Basis
- Otherwise, we still lack some algebraic tool to help filtering out non-field solutions.

Questions?