Automatic Generator of Floating-Point Inputs by Constraint Solving

Gene Der Su

Jian (Kevin) Xu

Introduction

US$370 million loss is caused by the self-destruction of European Space Agency’s Ariane 5 rocket, due to a floating-point overflow.

Car brake system

Remote surgery system

4 Kinds Of Floating-Point Problems

1. Overflow
2. Underflow
3. Invalid number
4. Inexact number

Examples of Problems - Floating-Point Error Accumulation

```c
float t = 0.0f;
int i;
for (i = 0; i < 20000; i++)
      t += 0.1f;
printf("%f\n",t);
```

$$(0.1)_{10} = (0.000\overline{1100})_2$$

t = 1999.658813

rather than

2000

Examples of Problems - Round Off Error

$$f(x) = \frac{1 - cos(x)}{x^2}$$ for $x = 1.2\times10^{-5}$

In 10-digit Floating-Point,

$$\frac{10^{-10}}{1.44 \times 18^{-10}} = 0.6944$$

In Real,

$$f(x) = \frac{1}{2}\left(\frac{sin(x/2)}{x/2}\right)^2 \approx 0.5$$

Relative error 0.39

Motivation - Problem in Klee

Klee was not able to solve all floating-point constraints due to the **multivariate** and **nonlinear** properties of the constraints.

There is a need of constraint solver for floating-point inputs

Technical Approach

- Randomly generate polynomial constraint
- Concretization
- Tell the parity of maximum exponent

Technical Approach

- **Randomly generate polynomial constraint**
- Concretization
- Tell the parity of maximum exponent

Example Of Constraints

constraint	source
$(1.5 - x1 * (1 - x2)) == 0$	Beale
$(-13 + x1 + ((5 - x2) * x2 - 2) * x2) + (-29 + x1 + ((x2 + 1) * x2 - 14) * x2) == 0$	Freudenstein and Roth
$pow((1 - x1), 2) + 100 * (pow((x2 - x1 * x1), 2)) == 0$	Rosenbrock
$((pow(((x * (sin((((y * 0.017) - (z * 0.017)) + (((((((pow(w, 2.0))/((sin((t* 0.017)))/(cos((t*0.017)))))/68443.0)*0.0)/w)*-1.0)*x)/(((pow(x, 2.0))/((sin((t* 0.017)))/(cos((t * 0.017)))))/68443.0)))))) - (w * 0.0)), 2.0)) + (pow(((x* (cos((((y * 0.017) - (z * 0.017)) + (((((((pow(w, 2.0))/((sin((t * 0.017)))/(cos((t* 0.017)))))/68443.0)*0.0)/w)*-1.0)*x)/(((pow(x, 2.0))/((sin((t*0.017)))/(cos((t* 0.017)))))/68443.0)))))) - (w * 1.0)), 2.0))) == 0.0$	TSAFE
$((exp(x) - exp((x * -1.0)))/(exp(x) + exp((x * -1.0)))) > (((exp(x) + exp((x * -1.0))) * 0.5)/((exp(x) - exp((x * -1.0))) * 0.5))$	PISCES
$x^{tan(y)} + z < x * atan(z) \wedge sin(y) + cos(y) + tan(y) >= x - z \wedge atan(x) + atan(y) > y$	manual

As a start, we focus on polynomial constraints

$$\sum_{j=1}^{m} \sum_{i=1}^{n} c_{i,j} x_i^j \oplus k$$

j is the exponent, x_i is the i^{th} variable

$c_{i,j}$ is a constant coefficient between -1 and 1

$$\oplus \in (<, =, >, \leq, \geq)$$

$$n, m \in [1, 5]$$

Random Generation Of Polynomial Constraint

$$\sum_{j=1}^{m} \sum_{i=1}^{n} c_{i,j} x_i^j \oplus k$$

$$\begin{aligned}
& c_{1,1} x_1 + c_{1,2} x_1^2 + c_{1,3} x_1^3 + c_{1,4} x_1^4 + c_{1,5} x_1^5 \\
& + c_{2,1} x_2 + c_{2,2} x_2^2 + c_{2,3} x_2^3 + c_{2,4} x_2^4 + c_{2,5} x_2^5 \\
& + c_{3,1} x_3 + c_{3,2} x_3^2 + c_{3,3} x_3^3 + c_{3,4} x_3^4 + c_{3,5} x_3^5 \\
& + c_{4,1} x_4 + c_{4,2} x_4^2 + c_{4,3} x_4^3 + c_{4,4} x_4^4 + c_{4,5} x_4^5 \\
& + c_{5,1} x_5 + c_{5,2} x_5^2 + c_{5,3} x_5^3 + c_{5,4} x_5^4 + c_{5,5} x_5^5
\end{aligned}$$

$(= , >, >=, <, <=)$ k

Technical Approach

- Randomly generate polynomial constraint
- **Concretization**
- Tell the parity of maximum exponent

Concretization - Simple Example

Concretization: Conversion from multivariate to univariate

$$x_1{}^2 + x_2{}^3 + x_3{}^5 < 10$$

$$\text{make } x_2 = 1 \text{ and } x_3 = 1$$

Then Solve $x_1{}^2 < 8$

Let $x_1{}^2 = 7$ so $x_1 = 2.646$

Concretization - How To Solve The Constraints

When the maximum exponent is odd,

Make $x_2 = x_3 = x_4 = x_5 = 1$

Then solve for x_1

About Concretization

- Concretization is effective due to the fact that the variables in our numerical constraints in practice exist solutions.

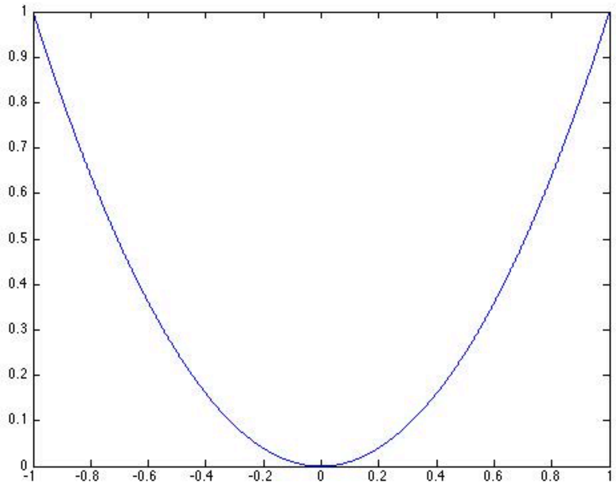
- Usually, the more complex variables in the constraint are concretized, and the simple one is left to be solved.

Technical Approach

- Randomly generate polynomial constraint
- Concretization
- **Tell the parity of maximum exponent**

Methods We Used To Cover Even Constraints

When exponent = 2 or 4, there is problem.



$y = x^2$

When the limits of the variable goes to positive and negative infinity, the constraint will result the same extreme (i.e. the constraint in only covering around half of the real space).

In other words, the solution may spin complex space naturally or if we did not pay attention in choosing the concertized variables.

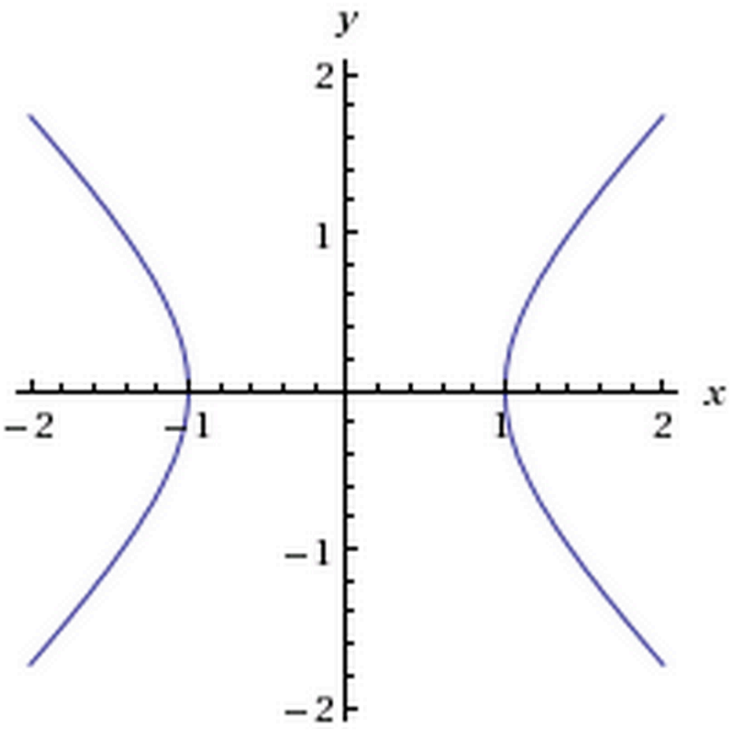
Problems With Even Exponent

For $- y^2 + x^2 = 1$

In this example, if we set x=0, then the equation becomes unsatisfiable in real space.

We will check the upper and lower bound for each variables and determine which one to concertize and which one to solve.

We can choose y=0 and attempt to solve $x^2=1$.



Algorithm for floating-point constraint solver

if *ODD exponent constraint* **then**
    (1) Concretize all variables to 1 except the first variable;
    (2) Solve for the roots of the equation with one variable;
    (3) Use the first real root as the generated value;
else if *EVEN exponent constraint* **then**
    (1) Determine the minimum/maximum of each individual variables;
    (2) Find the global minimum and/or maximum of the function;
    (3) Determine whether the constraint is satisfiable;
    if *Unsatisfiable* **then**
        Output "Unsatisfiable"
    else
        (1) Use the minimum/maximum information to determine which variables to concretize to 0;
        (2) Solve the root of last variable as the generated value;

An Illustrative Example: $X_1^2 - X_2^2 > 4$

1. The algorithm first check the dominant exponent (i.e. 2) and know that this is an even exponent constraint.
2. It determines there is a minimum for $f(x_1)$ at $x_1 = 0$ and gives 0, and a maximum for $f(x_2)$ at $x_2=0$ and gives 0.
3. This information tells us that $x_1^2 – x_2^2$ can form any number from -infinity to infinity and that the constraint is satisfiable.
4. Using that information, we know $f(x_1)$ can be anything greater than 0 and $f(x_2)$ can be anything less than 0. Since we want $f(x_1,x_2)>4$, we will leave x_1 free and assign 0 to x_2.
5. The algorithm will now view the constraint as $x_1^2>4$ and attempt to solve $x_1^2=5$.
6. It will output the solution as $(x_1,x_2)=[2.236, 0]$.

Evaluation

- **Percent of Coverage**
- **Average running time**

Percent of Coverage

$$\text{Coverage} = \frac{Number\ of\ satisfied\ cases\ by\ our\ tool}{Number\ of\ total\ satisfiable\ cases} \times 100\%$$

Evaluation – Coverage

Running 10,000 times for each cell

	m=1	m=2	m=3	m=4	m=5
n=1	100%	100%	100%	100%	100%
n=2	100%	97.04%	100%	96.24%	100%
n=3	100%	97%	100%	97.12%	100%
n=4	100%	97.85%	100%	97.97%	100%
n=5	100%	98.9%	100%	99.43%	100%

$$\sum_{j=1}^{m} \sum_{i=1}^{n} c_{i,j} x_i^j \oplus k$$

Example Of Satisfied Constraint: n=2, m=2

$$- 0.3093\ x_1 - 0.7541\ x_1^2 + 0.6458\ x_2 + 0.3867\ x_2^2 >= -0.7170$$

$f(0.4995,0) = -0.1545$

Generated input:

$x_1 = 0.4995$, $x_2 = 0$, ($x_3 = 0$, $x_4 = 0$, $x_5 = 0$)

Example Of Satisfied Constraint: n=5, m=5

$$
\begin{aligned}
& 0.5447\, x_1 + 0.9129\, x_1^2 - 0.6191\, x_1^3 - 0.4655\, x_1^4 + 0.9946\, x_1^5 \\
& + 0.2500\, x_2 + 0.0132\, x_2^2 + 0.6500\, x_2^3 + 0.9432\, x_2^4 - 0.0404\, x_2^5 \\
& + 0.9473\, x_3 + 0.1055\, x_3^2 + 0.8696\, x_3^3 - 0.4645\, x_3^4 - 0.3716\, x_3^5 \\
& - 0.1780\, x_4 - 0.0363\, x_4^2 - 0.5614\, x_4^3 - 0.5838\, x_4^4 - 0.0558\, x_4^5 \\
& - 0.9398\, x_5 - 0.4217\, x_5^2 + 0.2957\, x_5^3 - 0.7398\, x_5^4 - 0.5765\, x_5^5
\end{aligned}
$$

> 0.4450

$f(1.5196,1,1,1,1) = 6.2783$

Generated input:

$x_1 = 1.5196$, $x_2 = 1.0000$, $x_3 = 1.0000$, $x_4 = 1.0000$, $x_5 = 1.0000$

Example Of Unsatisfiable Constraint: n=5, m=4

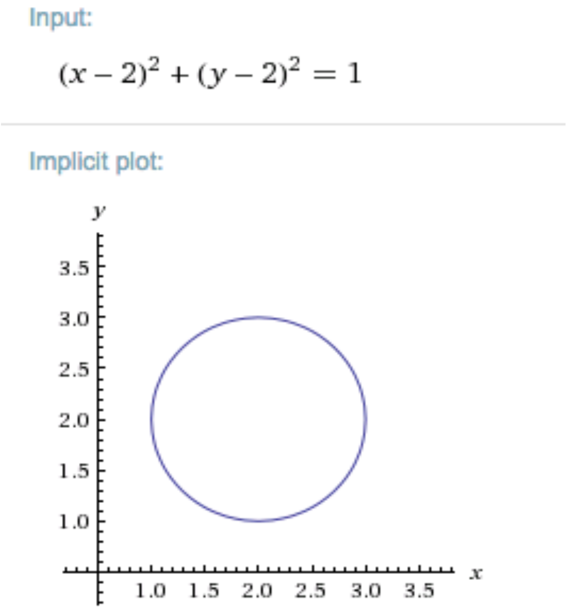
$$
\begin{aligned}
&- 0.4847\, x_1 + 0.8676\, x_1^2 - 0.2709\, x_1^3 + 0.7819\, x_1^4 \\
&+ 0.4646\, x_2 + 0.9364\, x_2^2 - 0.6033\, x_2^3 + 0.6978\, x_2^4 \\
&+ 0.0857\, x_3 - 0.3179\, x_3^2 - 0.1652\, x_3^3 + 0.8966\, x_3^4 \\
&- 0.3374\, x_4 - 0.3033\, x_4^2 + 0.4198\, x_4^3 + 0.2463\, x_4^4 \\
&- 0.4986\, x_5 + 0.4031\, x_5^2 - 0.0648\, x_5^3 + 0.7948\, x_5^4
\end{aligned}
$$

$$<= \quad -0.6933$$

$$
\begin{aligned}
&f(x_1) \in [-0.06885, \infty) \\
&f(x_2) \in [-0.04952, \infty) \\
&f(x_3) \in [-0.03593, \infty) \\
&f(x_4) \in [-0.34766, \infty) \\
&f(x_5) \in [-0.11875, \infty) \\
&f(x_1, x_2, x_3, x_4, x_5) \in [-0.62071, \infty)
\end{aligned}
$$

Example Of Unsatisfied Constraint

$$(x - 2)^2 + (y - 2)^2 = 1$$



We need both variable to solve the constraint. Concertized any variable to zero will lead to the constraint unsatisfied.

Evaluation – Average Running Time

$$\sum_{j=1}^{m} \sum_{i=1}^{n} c_{i,j} x_i^j \oplus k$$

	m=1	m=2	m=3	m=4	m=5
n=1	1.776e-4	4.306e-4	1.504e-4	2.758e-4	1.832e-4
n=2	1.623e-4	3.283e-4	1.741e-4	4.002e-4	1.752e-4
n=3	1.301e-4	4.428e-4	1.803e-4	5.726e-4	1.749e-4
n=4	1.510e-4	5.073e-4	1.472e-4	6.385e-4	1.620e-4
n=5	1.660e-4	6.140e-4	2.097e-4	7.786e-4	1.876e-4

Time in seconds, averaged from 10,000 runs.

Conclusion

1. Odd exponent polynomials are easy to deal with because it spans the entire real number space and always satisfiable.
2. Even exponent polynomials need special treatments to deal with the satisfiability and may need more than one free variable to solve the constraint.
3. Only the dominant exponent in the polynomial matters in the determination of the odd or even exponent.
4. There are much works to be done to deal with all possible constraints.

Future work

- Negative exponents.
- Product and quotient of different variables altogether.
- Other math functions like sin(), exp(), log()...etc.
- Combination of multiple constraints.