MSR 2018

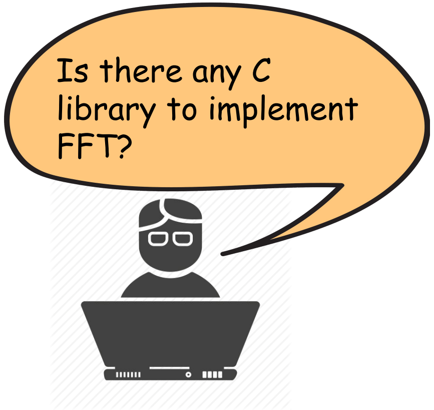
A Search System for Mathematical Expressions on Software Binaries

Ridhi Jain
Sai Prathik
Venkatesh Vinayakarao
Rahul Purandare
Indraprastha Institute of Information Technology, Delhi

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

Acknowledgements: Microsoft Research, FormaliSE

Introduction



Quora

Fast Fourier Transform (FFT) | C (programming language) | +1

Are there any libraries in C to implement FFT's?

Answer | Request | Follow 2 | Comment | Downvote

C++ Math Library | search

4,450 results | relevance | newest | votes | active

Motivation

Developers search for Math Expressions

Inverse Chi Square Function in C#

1

this is the equation of what I want to implement:

$$H = C^{-1}(-2 \ln \prod_{w} f(w), 2n)$$

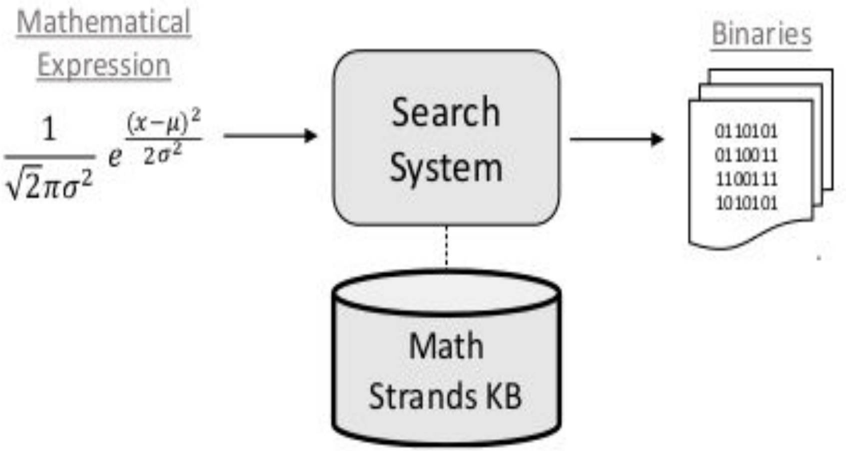
I have already made the inside up to comma. But I have really no idea how to implement C^-1 with 2n degrees of freedom, is there any function in Math class?

Looking at the description on the wiki I am still confused like a lil kid. wikipedia inverse chi function

I have found the Python implementation: Python implementation of chi function

No search systems for binaries!

System Overview



An Overview of the System

Research Question

Can we locate a given Mathematical Expression in a given Software Binary?

Challenges in Handling Binaries

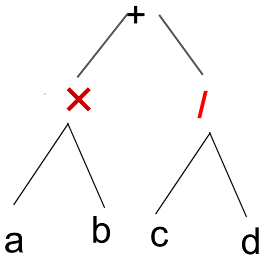
x^2

Optimization -O0

```
callq  4006c0 <pow@plt>
```

Optimization -O2 and -Os

```
mulsd  %xmm0,%xmm0
```



Variants — **Ghost Ops** — **Evaluation Ordering** — **Operand Resolution**

```
add    $0x18,%rsp
```

$b^2 - 4ac$

$=$

$a^2 - 4ac$

Challenges in Working with ME

x×y = x*y = xy

Content MathML normalizes it

Specification of Operation

```
<apply>
        <times/>
                <ci>x</ci>
                <ci>y</ci>
</apply>
```

Types of Operations

log(x)

Optimization -O0

```
callq   4006c0 <log@plt>
```

Optimization -O2 and -Os

```
callq   400680 <_ZNSo9_M_insertIdEERSoT_@plt>
```

Related Work

Tracelet-based Code Search in Executables
PLDI' 14

Retrieving Documents with Mathematical Content
SIGIR' 13

Detecting Code Clones in Binary Executables
ISSTA' 09

Work for search in software binaries

Retrieves ME in documents

No comprehensive solution for searching ME in software binaries!!

Key Contributions

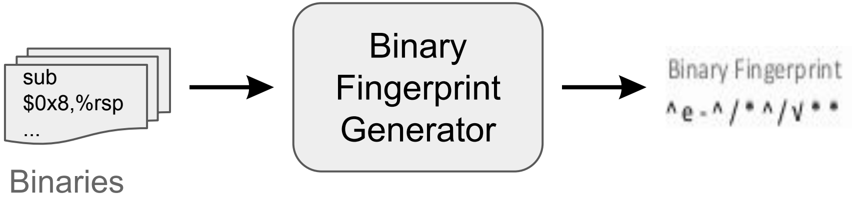
- A search system to search for ME in binaries.
- An approach to compare binaries and ME.
- A knowledge base of math operators mapped to their assembly opcodes.

Components of the System

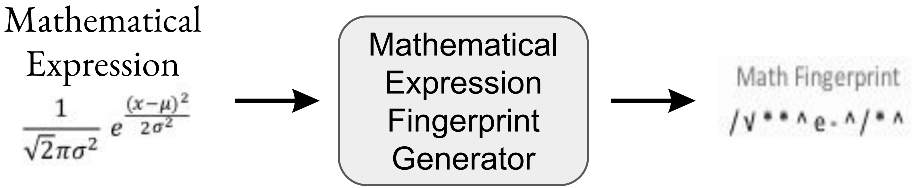
Math Strands Knowledge Base (M_S KB):

x^2 pow, mulsd
e^x exp
√x sqrt
...

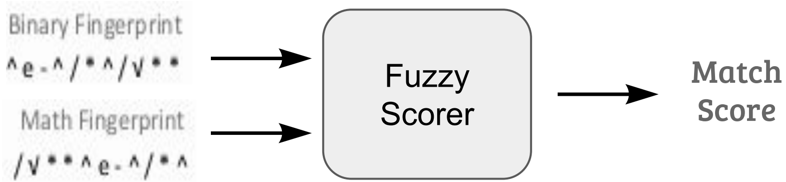
Binary Fingerprint Generator (B_{fp}):



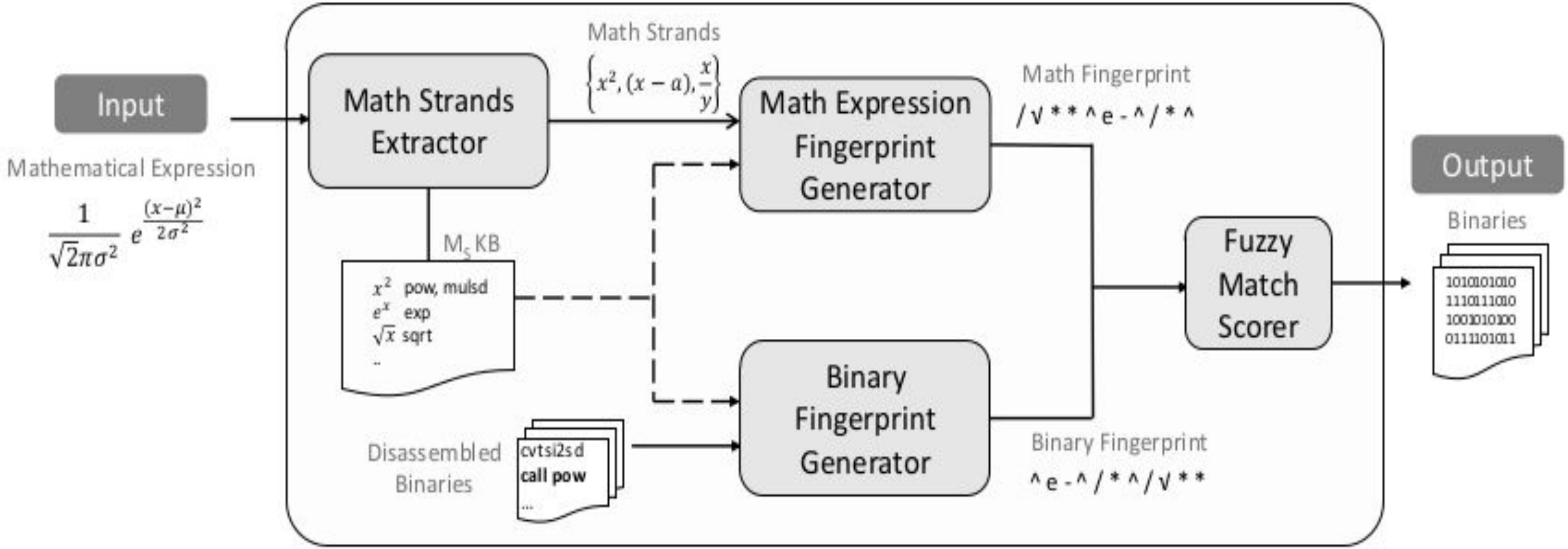
Mathematical Expression Fingerprint Generator (M_{fp}):



Fuzzy Scorer:



Approach



Input
Mathematical Expression
$\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$
Math Strands Extractor
Math Strands
$\{x^2, (x-a), \frac{x}{y}\}$
M_S KB
x^2 pow, mulsd
e^x exp
$\sqrt{x}$ sqrt
Math Expression Fingerprint Generator
Math Fingerprint
/ √ * * ^ e - ^ / * ^
Binary Fingerprint Generator
Binary Fingerprint
^ e - ^ / * ^ / √ * *
Disassembled Binaries
cvtsi2sd
call pow
Fuzzy Match Scorer
Output
Binaries
1010101010
1110111010
1001010100
0111101011

Evaluation: Selection of Expressions

- We focus on expressions whose fundamental building blocks are generally available in the in-built language features of high-level languages such as C, C++ and Java.
- We select 4 ME listed as follows:

Name	Expression	Type
Heron's Formula	$\sqrt{(s(s-a)(s-b)(s-c))}$	Algebraic
Sigmoid Function	$1/(1+e^{-x})$	Transcendental
Compound Interest	$K_n = K_0 * (1+(P/100))^n$	Algebraic
D1-Black-Scholes	$(\log(s_0/x)+t(r-q+(\sigma^2/2)))/(\sigma\sqrt{t})$	Transcendental

- We convert the selected **ME** to **ContentML** using Visual Math Editor.

Evaluation: Selection of Projects

- We pick 20 projects from GitHub from distinct domains such as Finance, Algebra, and Machine Learning.
- For each of the ME of interest, we pick 5 **C/C++** projects.
- Next, we compile all 20 programs with **gcc/g++** optimization levels: **-O0**, **-O2**, **-Os**
- As a result we had 60 binaries in which we searched for all the expressions.

Heuristics

- We use **Longest Common Subsequence** algorithm to find generate the match score for Binary Fingerprint (B_{fp}) and Mathematical Expression Fingerprint (M_{fp}), $LCS(M_{fp}, B_{fp})$.

- $LCS(M_{fp}, B_{fp}) > \sigma$ implies that ME was found in the binary.

- To improve the precision we use two heuristics:
  - **Length Heuristics for short ME**: If $|M_{fp}| < \alpha |B_{fp}|$ then ME $\notin$ binary.
  - **Relevance Heuristics for Ghost Ops**: $(Irr(B_{fp}, M_{fp})) = |s|$ such that strand $s \in B_{fp}$ and $s \notin M_{fp}$. If $(Irr(B_{fp}, M_{fp})) > \beta |B_{fp}|$ then ME $\notin$ binary.

- We derive the parameters **α** and **β** empirically to be **0.25** and **0.40** respectively

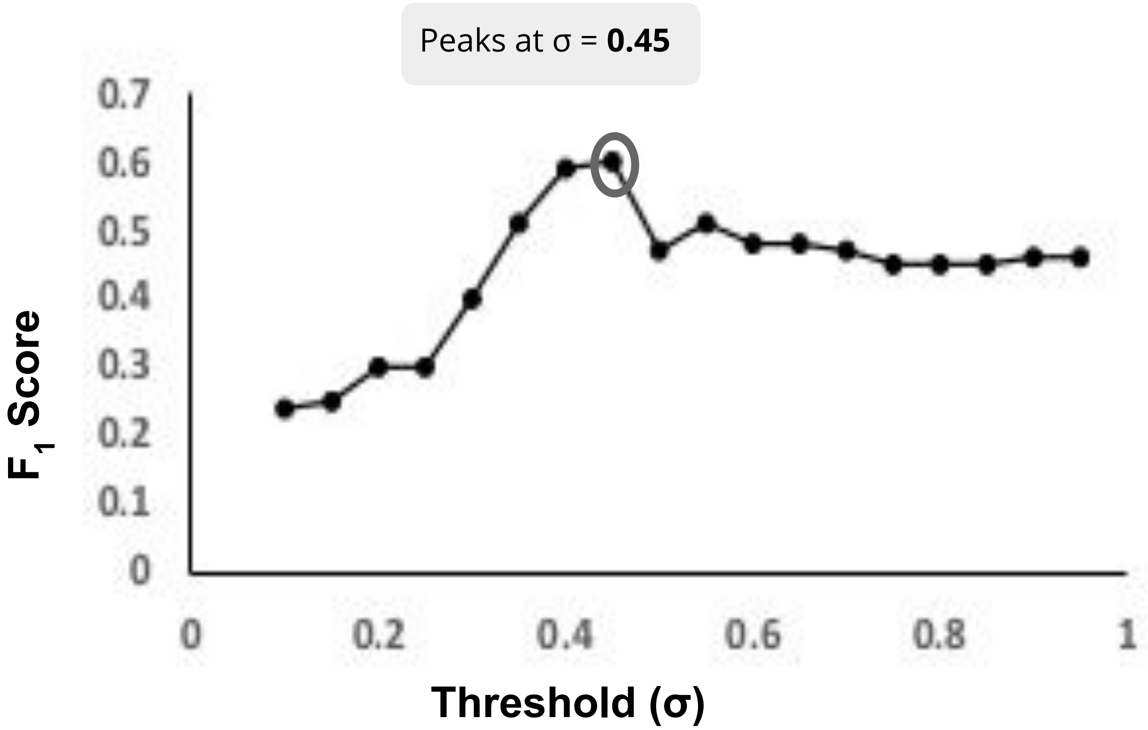
Results

- We compute precision and recall for all the binaries for all ME
- We compare our results against ground truth for various values of threshold
- F_1 score peaks to 0.61 for a threshold value, **σ = 0.45**
- With our approach we were able to identify **ME** in software binaries with an average precision of **80%** and a recall of **53%**.

Results

ME	M_{fp}	Precision	Recall	F_1 Score
Heron's Formula	-*-*-*q	0.69	0.56	0.61
Sigmoid Function	e+/	1.00	0.54	0.70
Compound Interest	/+^*	0.50	0.62	0.56
D1-Black-Scholes	/l^/+-*+q*/	1.00	0.38	0.55
Average		**0.80**	**0.53**	**0.61**

Results



F_1 = 0.61 at

σ = 0.45
α = 0.25
β = 0.40

Conclusion

- Mathematical expressions and software binaries pose different challenges.
- We solve these problems using data driven fingerprinting based approach.
- We show that **ME** can be located in binaries with **61% F_1 score** for algebraic and transcendental expressions.

Future Work

- We envision automating the creation of M_s KB for multiple system architectures.
- Classes of operations that still remain to be explored, such as, logical and relational.
- Iterative operations such as Summation ($\sum$) and Product ($\prod$) require the operation to be applied over a range of values.
- Precision and Recall can be improved by keeping track of the operands in the ME.

Thank You

Questions?