



# Approximate Model Counting and its applications in Quantitative Information Flow

Tejas Anand<sup>1</sup> Kohei Suenaga<sup>2</sup>

IIT Delhi<sup>1</sup> Kyoto University<sup>2</sup>



## What is Model Counting ?

- Model counting is the problem of counting the number of solutions to a given set of constraints.
- The problem of Model Counting (#SAT) is #P-complete. In very simple words, finding the **Exact Count** would take an abnormally high amount of time.
- Therefore, we work with an  $(\epsilon, \delta)$  approximation algorithm  $\mathcal{A}$ , whose output  $\mathbf{n}$  over a problem instance  $\mathcal{F}$  satisfies,

$$\Pr[\mathbf{n} \leftarrow \mathcal{A}(\mathcal{F}) : \frac{\#\mathcal{F}}{1+\epsilon} \leq \mathbf{n} \leq \#\mathcal{F}(1+\epsilon)] \geq 1 - \delta$$

- In simple words, it gives a good enough number with high probability, for small values of  $\epsilon$  and  $\delta$ .
- For instance, we might want to count the number of equivalence classes of the given relation

$$x \sim y \Leftrightarrow x \equiv y \equiv 0 \pmod{8} \vee x = y \quad (1)$$

## Research Problem

- Recently, a scalable approximation algorithm for model counting over boolean constraints was proposed by Chakraborty et al.
- We want to generalize this algorithm to simple arithmetic constraints like modulo, addition, subtraction, etc. over integers (finite fields like  $Z_n$ ) and lists, using SMT solvers (SAT modulo theory) like Z3.
- This has applications in computer security, it would be the main ingredient to quantify the sensitive information leaked by a computer programme.

## KU-STAR Internship

- As of now, we have worked on the correctness of the algorithm for  $Z_k$
- Future work would involve implementing the algorithm and experimenting on relevant examples.

## Example

**Question:** How many equivalence classes does the following relation have, where  $x, y$  are 32 bit integers ?

$$x \sim y \Leftrightarrow x \equiv y \equiv 0 \pmod{8} \text{ or } x = y \quad (2)$$

**Answer:**  $7 \cdot 2^{29} + 1$ . All of the multiples of 8 form 1 equivalence class, and the remaining  $7/8^{th}$ s of the total  $2^{32}$  integers form singleton equivalence classes of their own.

$$(x \equiv y \equiv 0 \pmod{8}) \vee (x = y) \rightarrow \text{Our Algorithm} \rightsquigarrow 7 \cdot 2^{29}$$

## Polynomially Many Queries to the oracle SMT

- For simplicity, we have an oracle **SMT**, which on invocation over an arithmetic formula  $\mathcal{F}$ , gives us any one solution to it or tells if it is **UNSAT**.
- The model count,  $\mathcal{F}$  can be exponentially large, so the naive approach of invoking the oracle till we get no new solutions is infeasible.
- A better way is to keep adding the negation of all the models we have obtained so far to the formula  $\mathcal{F}$  as  $\mathcal{F} \leftarrow \mathcal{F} \wedge (x \neq x_0) \wedge (x \neq x_1) \cdots (x \neq x_i)$ , where  $x_0, x_1, \dots, x_i$  are the models obtained so far. This approach is better than the first one, but it is still exponential.

## How ApproxMC works, intuitively

- We must take advantage of the fact that we are not concerned with the exact model count.
- So we divide the set of models uniformly into **cells**, using a randomly sampled  $r$ -wise independent hash function.
- All models having a given hash-value belong to a given cell.
- The representative count of a given cell is then multiplied by the number of cells to get an approximate count of the total number of models of the formula.
- We try to make the number of cells such that it is "fine grained" enough and has  $\leq pivot$  models in it.
- This *pivot* is a function of the threshold  $\epsilon$  around the correct model that we desire. It is of the order  $\mathcal{O}(1/\epsilon^2)$ .
- Hence, we only need to make a bounded number of invocations to the oracle **SMT**.

## Algorithm 1 ApproxMCCore( $F, pivot$ )

```

1:  $S \leftarrow \text{BoundedSMT}(F, pivot + 1) \triangleright$  Assume  $x_1, x_2 \cdots x_q$  are the variables of  $F$ 
2: if  $|S| \leq pivot$  then
3:   return  $|S|$ ;
4: else
5:    $l \leftarrow \lfloor \log_k(pivot) \rfloor - 1; i \leftarrow l - 1$ 
6:   repeat
7:      $i \leftarrow i + 1$ ;
8:     Choose  $h \leftarrow \mathcal{H}_k(n, i - l, 3)$  uniformly at random;
9:     Choose  $\alpha \leftarrow Z_k^{i-l}$  uniformly at random;
10:     $S \leftarrow \text{BoundedSMT}(F \wedge h(x_1, x_2 \cdots x_q) = \alpha, pivot + 1)$ 
11:  until  $(1 \leq |S| \leq pivot)$  or  $(i = n)$ ;
12: end if
13: if  $(|S| > pivot \text{ or } |S| = 0)$  then
14:   return  $\perp$ ;
15: else
16:   return  $|S| \cdot k^{i-l}$ ;
17: end if
```

$\mathcal{H}_k(n, m, r)$  = Set of hash functions from  $Z_k^n \rightarrow Z_k^m$  which are  $r$ -wise independent. The no. of cells is equal to the number of unique hash values possible which is equal to  $|Z_k^m| = k^m$

## Quantitative Information Flow

- We often require the number of equivalence classes to calculate the **Bayes Vulnerability** which is an Adversary's maximum probability of guessing the correct "secret" given he knows the encryption scheme.
- For some information theoretic measures, we even require the size of each equivalence class.
- Our work can be used to quantify the sensitive information leaked by a computer program.

## References

[1] S. Chakraborty, K. S. Meel, and M. Y. Vardi. A scalable approximate model counter, 2013.