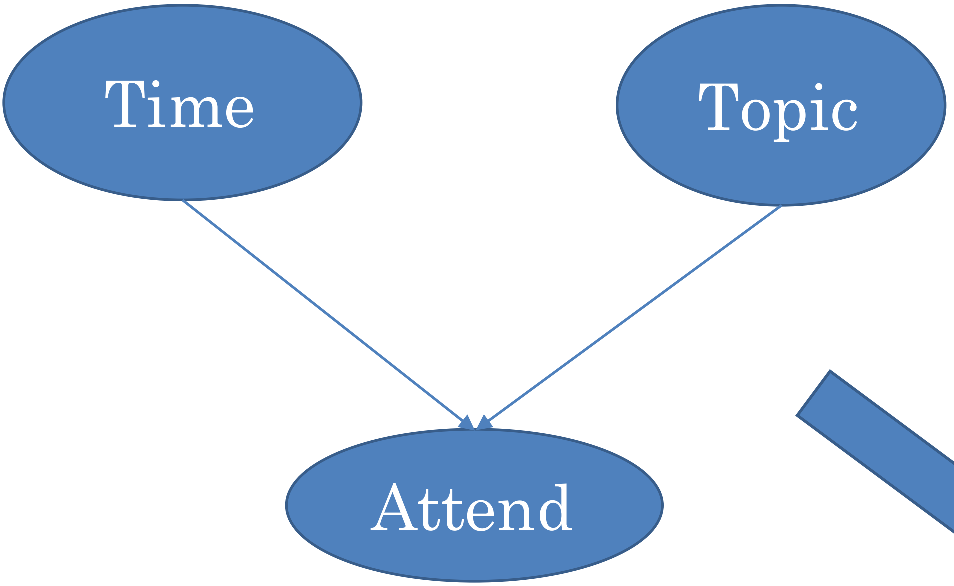
# Approximate Probabilistic Inference via Word-Level Counting

Supratik Chakraborty[1], ***Kuldeep S. Meel***[2], Rakesh Mistry[1], Moshe Y. Vardi[2]

[1] Indian Institute of Technology (IIT) – Bombay

[2] Department of Computer Science, Rice University

# Probabilistic Inference to Model Counting




Event ↓ Evidence ↓

Pr [Attend = Yes | Topic=GM]

Roth, 1996

**Model Counting**

Model counting as the "assembly language" for inference

# Model Counting

- Variables with Domain:
  - Time; {Morning, Afternoon, Evening}
  - Topic; {NLP, GM, Other}

Attend; {Yes, No}

- Constraints:
  - (Topic = GM → Attend = Yes) ∧ (Time = Afternoon → Attend = Yes)
- Models:
  - (Time = Afternoon, Topic = GM, Attend = Yes)
  - (Time = Evening, Topic = Other, Attend = No)
  - ..........
- Model Counting: Count the number of models (#P complete)

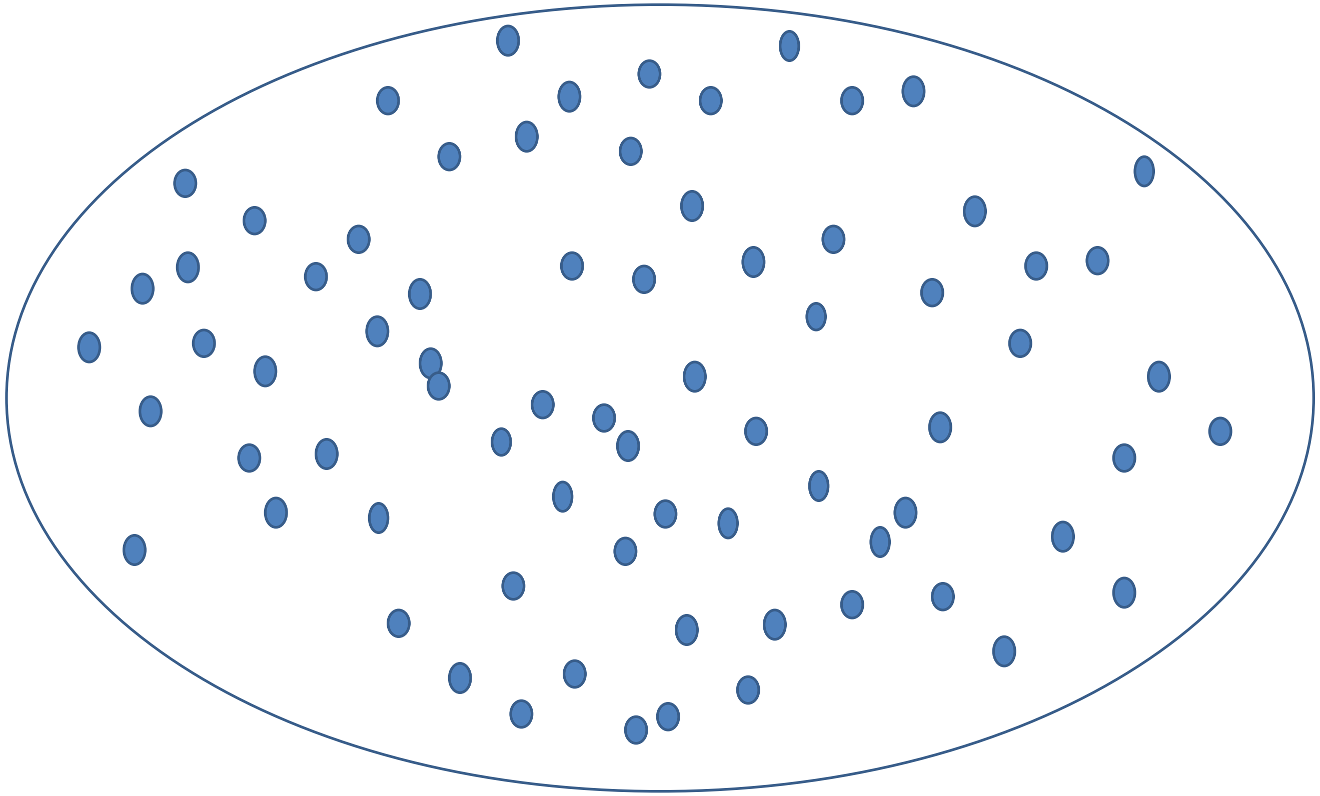
# Approximate Model Counting

- Approximate Model Counting

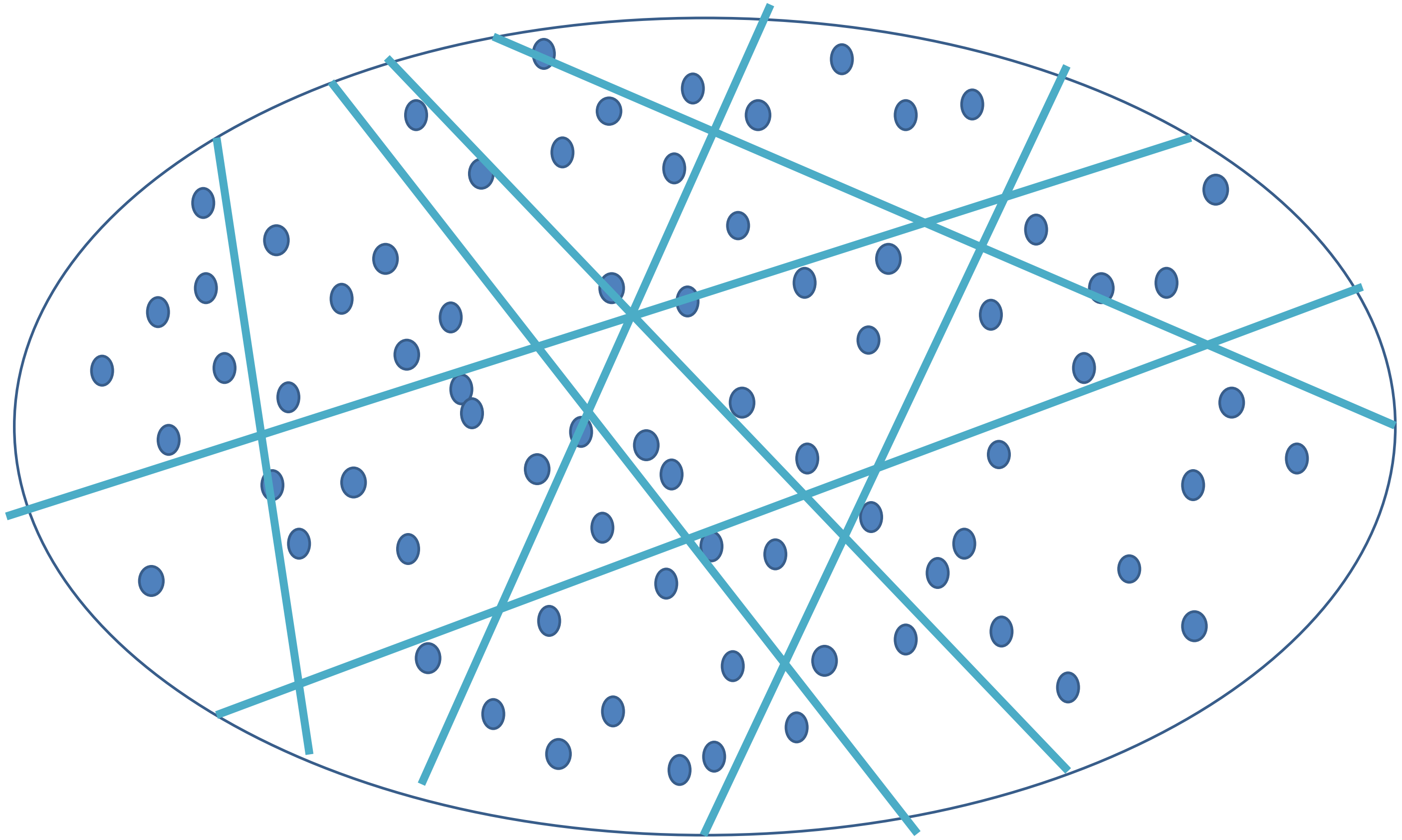
$$\Pr\left[\frac{|R_F|}{1+\varepsilon} \leq \text{ApproxMC}(F,\varepsilon,\delta) \leq (1+\varepsilon)|R_F|\right] \geq 1-\delta$$

- Hashing-based Approaches
  - CAV 2013
  - CP 2013
  - UAI 2013
  - NIPS 2013
  - DAC 2014
  - ICML 2014
  - AAAI 2014
  - TACAS 2015
  - IJCAI 2015
  - ICML 2015
  - UAI 2015
  - AAAI 2016
  - AISTATS 2016

# Partitioning into equal “small” cells

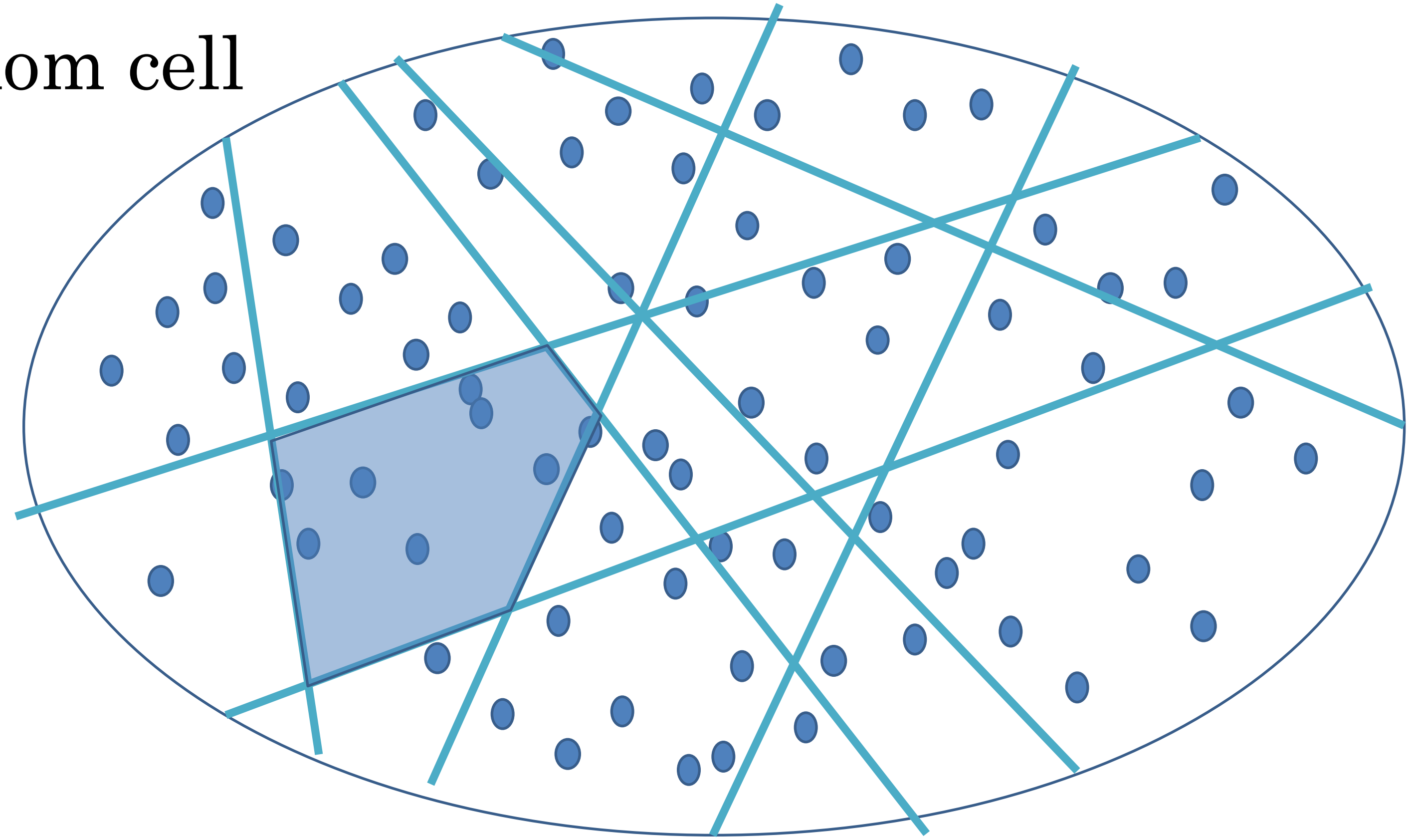


# Partitioning into equal "small" cells



# Partitioning into equal “small” cells

Pick a random cell



Estimate = # of models in cell * # of cells

# How to Partition?

How to partition into *roughly equal small cells* of solutions *without knowing the distribution* of solutions?

**Universal Hashing [Carter-Wegman 1979]**

# Bit-level reasoning

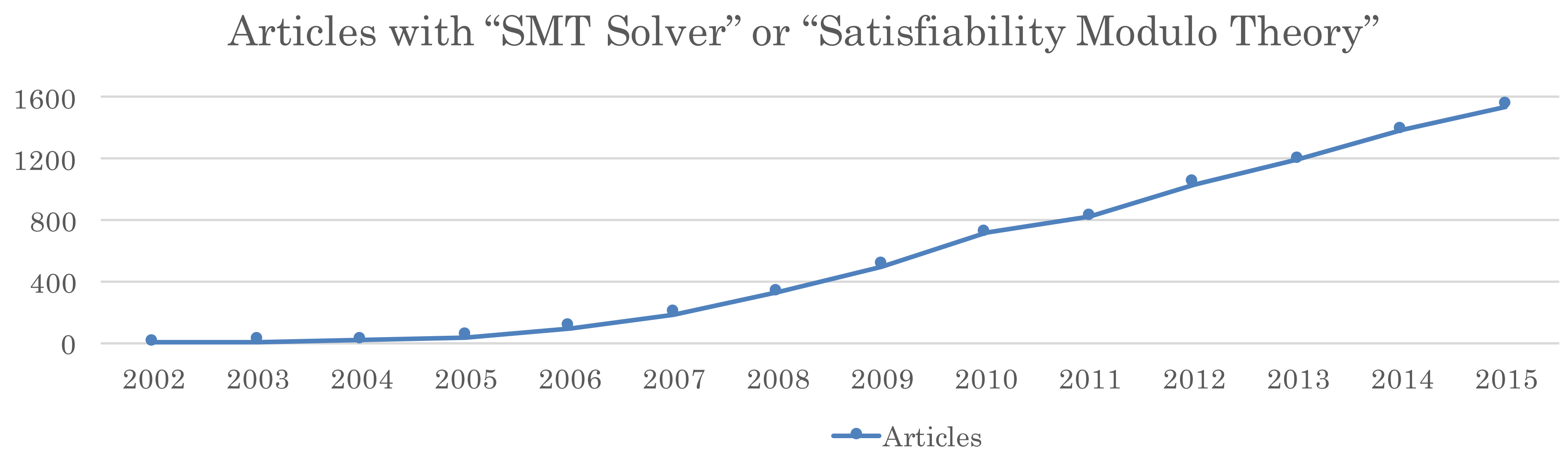
- XOR-based (mod 2) hash functions in **all** prior works

- Variables in Graphical Models are not binary
- Approach: Perform "bit-blasting"
  - $Dom(X) = \{0, 1, 2, 3\}$
  - X can be represented using two bits $(y_1, y_2)$ such that $X = y_1 y_2$
  - XOR constraints over $y_i$ variables

- Require solvers to perform bit-level reasoning

# Word-level Revolution

- Development of SMT Solvers to reason directly at the level of words (No need for “bit-blasting”)
- The biggest advance in formal methods in last 25 years (John Rushby, 2011)



# Our Contributions

- $H_{SMT}$: Efficient word-level Hash Function
- SMTApproxMC: Efficient word-level counter

# Towards Efficient word-level Hashing

- Lifting hashing from (mod 2) to (mod p) constraints
  - **p**: smallest prime greater than largest domain of variables

- Linear (mod p) constraints to partition into p cells
  - Amenable to Gaussian Elimination

- Number of cells (N) = $p^c$ ,
  - **Challenge**: Larger p does not give finer control on number of cells
  - Few cells → Too many solutions in a cell
  - Many cells → No solutions in most of the cells

# $H_{SMT}$: Efficient word-level Hash Function

- Use different primes to control the number of cells

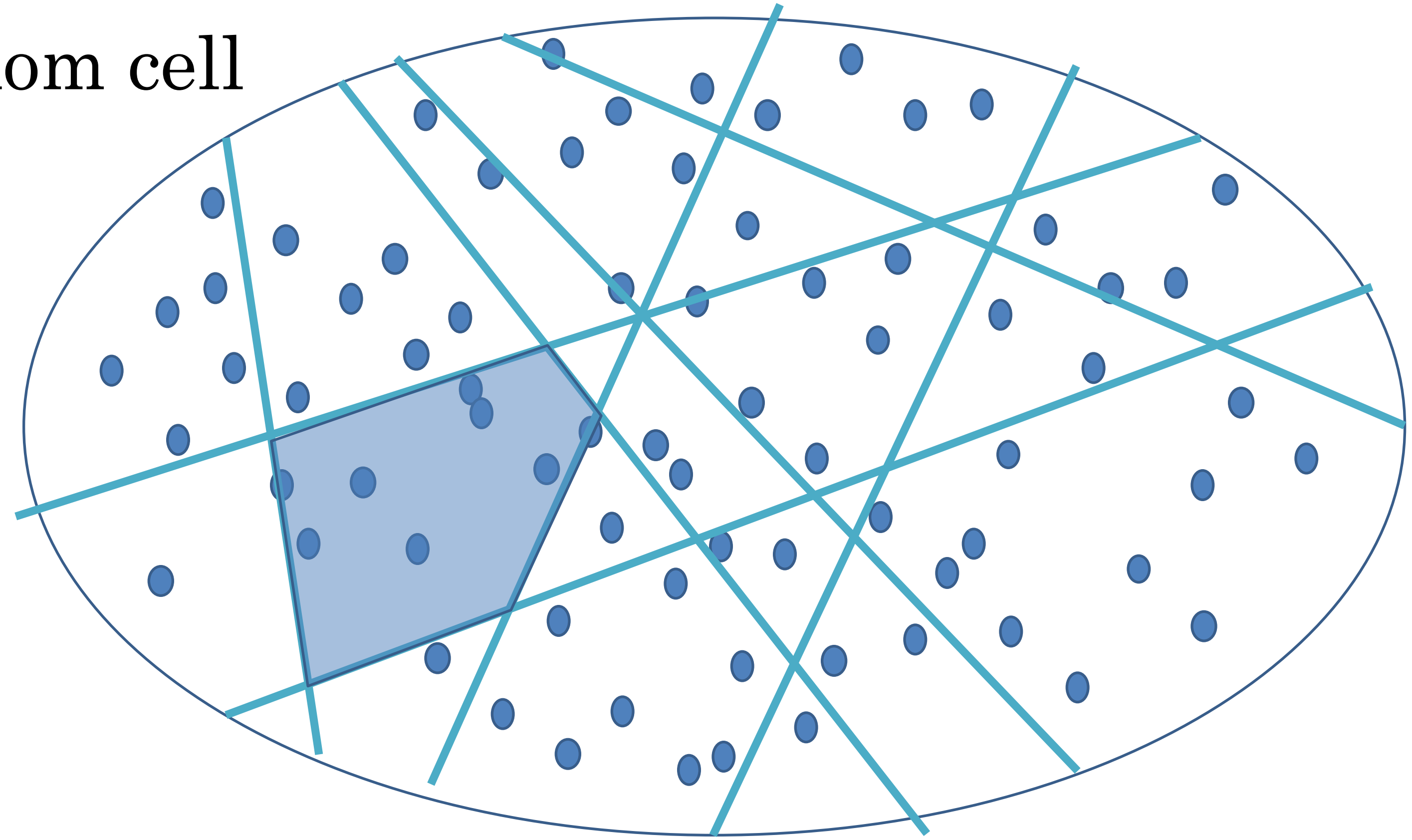
- Choose appropriate N and express as product of *preferred* primes, i.e. $N = p_1^{c_1} p_2^{c_2} p_3^{c_3} \ldots\ldots p_n^{c_n}$

- $H_{SMT}$:
  - $c_1$ (mod $p_1$) constraints
  - $c_2$ (mod $p_2$) constraints
  - ……..

- $H_{SMT}$ satisfies guarantees of 2-universality

# SMTApproxMC

Pick a random cell



Estimate = # of models in cell * # of cells

# Theoretical Guarantees

- $F$: Formula over bounded domain variables;
- $R_F$ : Solution Space of $F$
- SMTApproxMC
  - Input: $F$, $\varepsilon$, $\delta$ Output: $C$

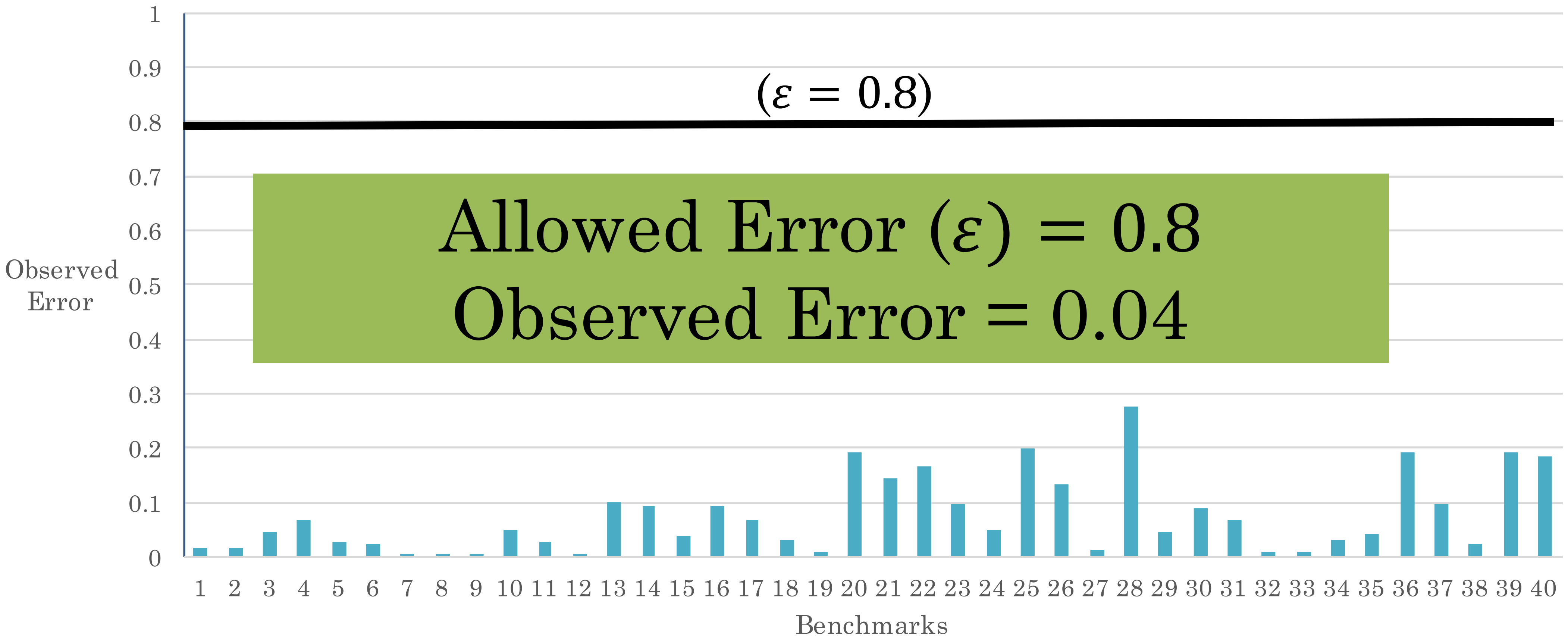
$$\Pr\left[\frac{|R_F|}{(1+\varepsilon)} \leq C \leq |R_F|(1+\varepsilon)\right] \geq 1-\delta$$

- Polynomial in $F, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)$ relative to word-level oracle

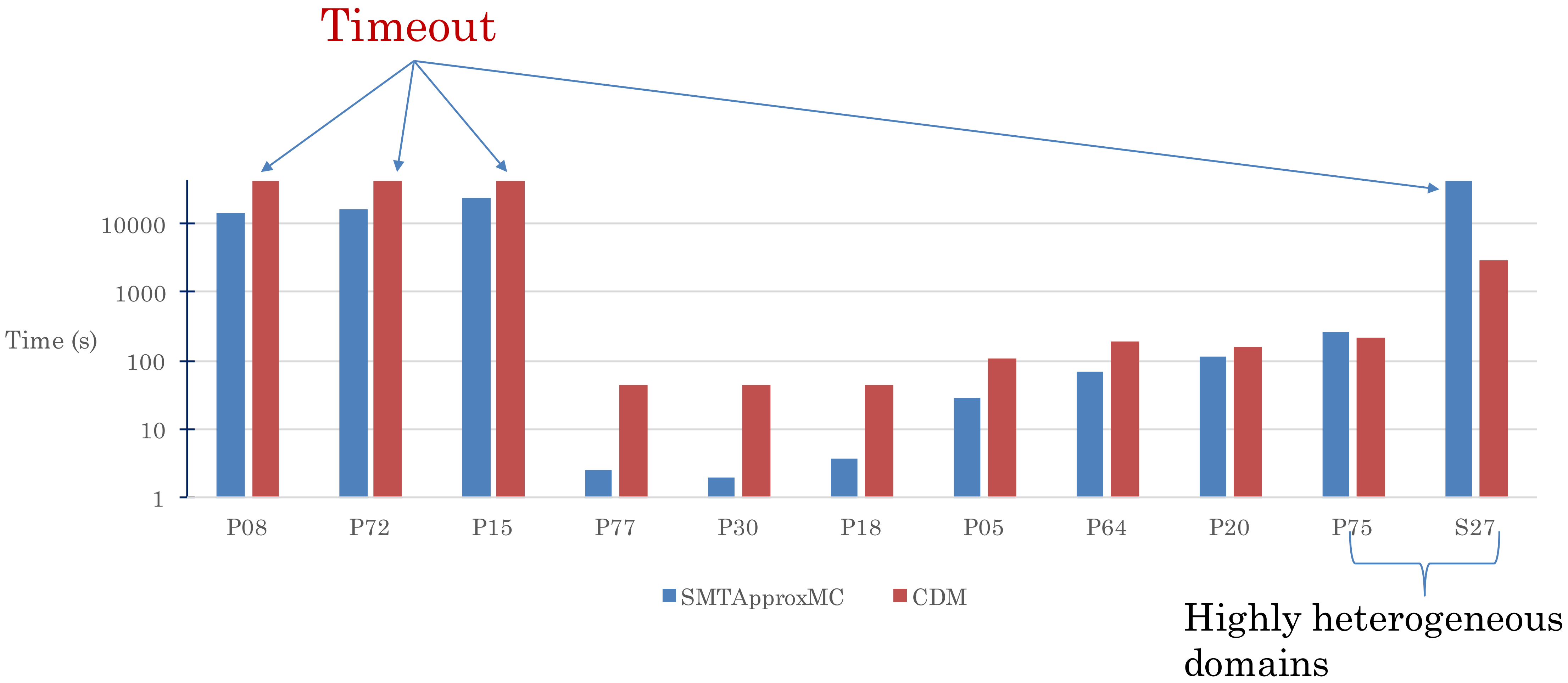
# Experimental Evaluations

- Over 150 benchmarks from:
  - Ising Models
  - ISCAS89 Circuits
  - Program Synthesis
- Comparison with state of the art tool: CDM
  - Based on Chistikov, Dimitrova, and Majumdar 2015
    - Similar to Ermon et al, Chakraborty et al etc..
    - Uses XOR-based hash functions
- Objectives:
  - Runtime performance comparison
  - Quality of estimates

# Quality Comparison



# Runtime Performance Comparison




SMTApproxMC is 2-10 times faster than CDM

# Summary

- Model Counting as "assembly language" for inference
- Recent model counting techniques rely on bit-level reasoning
- SMTApproxMC: The first efficient word-level model counter
  - Outperforms existing state of the art tool by 2-10 times in runtime
  - Observed error is far less than theoretical guarantees
- Source code available online at tinyurl.com/smtapproxmc