



# GANAK: A Scalable Probabilistic Exact Model Counter

Shubham Sharma<sup>†</sup>, Mate Soos<sup>‡</sup>, Subhajit Roy<sup>†</sup> and Kuldeep S. Meel<sup>‡</sup>

<sup>†</sup>Indian Institute of Technology Kanpur, India

<sup>‡</sup>National University of Singapore

Model Counting Workshop 2020

---

<sup>1</sup>This work is published in IJCAI 2019.

# Propositional Model Counting

- Given:
  - Propositional formula  $F$  (CNF) over a set of variables  $X$
- Propositional Model Counting (#SAT):
  - Compute the number of satisfying assignments of  $F$
- #SAT is a #P complete problem

- Probabilistic Exact Model Counting
  - Given a propositional formula  $F$  (CNF) and confidence  $\delta \in (0, 1]$ , counter returns `count` such that:

$$\Pr[|\text{Solutions of } F| = \text{count}] \geq 1 - \delta$$

# Propositional Model Counting

- Probabilistic Exact Model Counting
  - Given a propositional formula  $F$  (CNF) and confidence  $\delta \in (0, 1]$ , counter returns `count` such that:

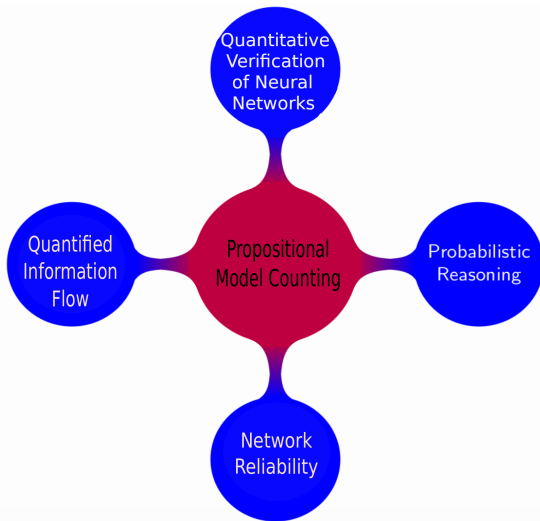
$$\Pr[|\text{Solutions of } F| = \text{count}] \geq 1 - \delta$$

- Probabilistic Exact Model Counting is almost as hard as Exact Model Counting<sup>2</sup>

---

<sup>2</sup>Chakraborty et al., 2019

# Applications of Propositional Model Counting



# Main Ingredients

- Decision Process:

- $(F \wedge I) \vee (F \wedge \neg I)$

- $\#(F) = \#(F \wedge I) + \#(F \wedge \neg I)$

mutually inconsistent

# Main Ingredients

- Decision Process:

- $(F \wedge I) \vee (F \wedge \neg I)$  mutually inconsistent
- $\#(F) = \#(F \wedge I) + \#(F \wedge \neg I)$

- Component Decomposition:

- $F = \Delta_1 \wedge \Delta_2 \cdots \Delta_n$   $\Delta_1 \cdots \Delta_n$  does not share any variables
- $\#(F) = \#(\Delta_1) \times \#(\Delta_2) \cdots \times \#(\Delta_n)$  mutually disjoint

# Main Ingredients

- Decision Process:

- $(F \wedge I) \vee (F \wedge \neg I)$  mutually inconsistent
- $\#(F) = \#(F \wedge I) + \#(F \wedge \neg I)$

- Component Decomposition:

- $F = \Delta_1 \wedge \Delta_2 \cdots \Delta_n$   $\Delta_1 \cdots \Delta_n$  does not share any variables
- $\#(F) = \#(\Delta_1) \times \#(\Delta_2) \cdots \times \#(\Delta_n)$  mutually disjoint

- Component Caching:

Key	Value
$\Delta_1$	$\#(\Delta_1)$
$\Delta_2$	$\#(\Delta_2)$



# Main Ingredients

- Decision Process:

- $(F \wedge I) \vee (F \wedge \neg I)$  mutually inconsistent
- $\#(F) = \#(F \wedge I) + \#(F \wedge \neg I)$

- Component Decomposition:

- $F = \Delta_1 \wedge \Delta_2 \cdots \Delta_n$   $\Delta_1 \cdots \Delta_n$  does not share any variables
- $\#(F) = \#(\Delta_1) \times \#(\Delta_2) \cdots \times \#(\Delta_n)$  mutually disjoint

- Component Caching:

Key	Value
$\Delta_1$	$\#(\Delta_1)$
$\Delta_2$	$\#(\Delta_2)$

- Conflict Driven Clause Learning

## Example

$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

# Example

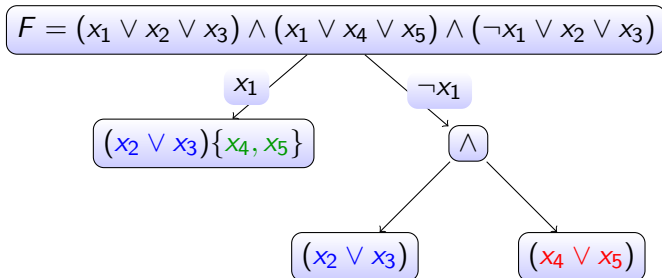
$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

$x_1$

$$(x_2 \vee x_3)\{x_4, x_5\}$$

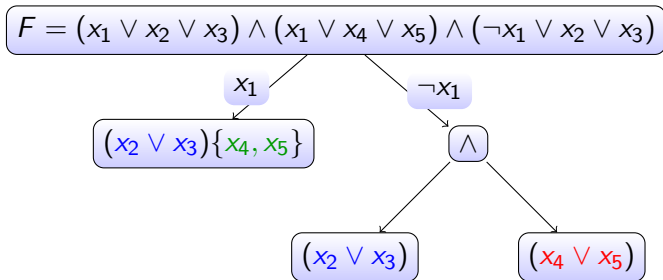
Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12

# Example



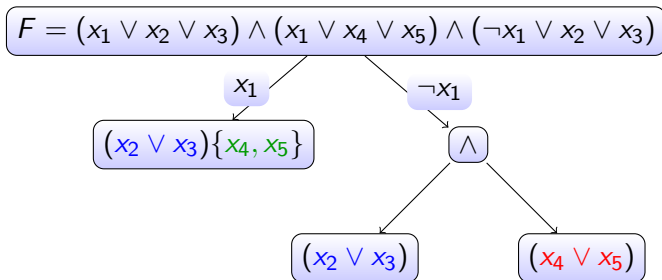
Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12

# Example



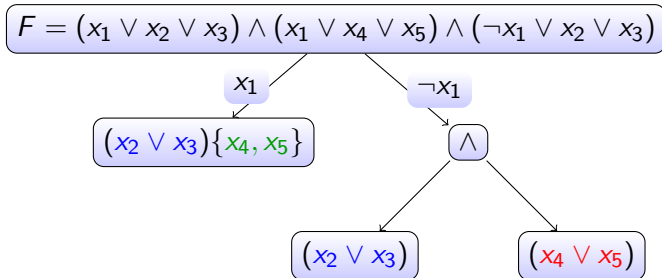
Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12
$(x_4 \vee x_5)$	3

# Example



Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12
$(x_4 \vee x_5)$	3
$(x_2 \vee x_3) \wedge (x_4 \vee x_5)$	9

# Example



Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12
$(x_4 \vee x_5)$	3
$(x_2 \vee x_3) \wedge (x_4 \vee x_5)$	9
$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3)$	21

- 1 **Probabilistic Component Caching (PCC)**
- 2 **Variable Branching Heuristic (CSVSAADS)**
- 3 **Phase Selection Heuristic (PC)**
- 4 Independent Support (IS)
- 5 Restarts (LSO)
- 6 Exponentially Decaying Randomness (EDR)



# Probabilistic Component Caching (PCC)

$$F = (\neg x_3 \vee \neg x_5 \vee x_6) \wedge (\neg x_1 \vee x_4 \vee \neg x_6) \wedge (x_2 \vee x_3 \vee x_6)$$

Schema	Key	Value
STD <sup>3</sup>	-3, -5, 6, 0, -1, 4, -6, 0, 2, 3, 6, 0	#(F)
HC <sup>4</sup>	1, 2, 3, 4, 5, 6, 1, 2, 3	#(F)
GANAK	Hash of HC/STD	#(F)

---

<sup>3</sup>Sang et al., 2005

<sup>4</sup>Thurley, 2006

# Variable Branching Heuristic (CSVSAADS)

- $\text{Score}(\text{VSADS})^5 = \underline{p \times \text{Score}(\text{VSIDS})} + \underline{q \times \text{Score}(\text{DLCS})}$ 
  - VSIDS: Prioritize variables present in recently generated conflict clauses
  - DLCS: Prioritize the highest occurring variable in the residual formula

---

<sup>5</sup>Sang et al., 2005

# Variable Branching Heuristic (CSVSADS)

- $\text{Score}(\text{VSADS})^5 = \underline{p \times \text{Score}(\text{VSIDS})} + \underline{q \times \text{Score}(\text{DLCS})}$ 
  - VSIDS: Prioritize variables present in recently generated conflict clauses
  - DLCS: Prioritize the highest occurring variable in the residual formula
- $\text{Score}(\text{CSVSADS}) = \underline{\alpha \times \text{CacheScore}} + \underline{\beta \times \text{Score}(\text{VSADS})}$ 
  - Prioritize variables not present in the components which are recently added to the cache

---

<sup>5</sup>Sang et al., 2005

# Phase Selection Heuristic (PC)

$$\text{DLIS}^6 \begin{cases} I & |I| \geq |\neg I| \\ \neg I & \textit{otherwise} \end{cases}$$

---

<sup>6</sup>Sang et al., 2005

## Phase Selection Heuristic (PC)

$$\text{DLIS}^6 \begin{cases} I & |I| \geq |\neg I| \\ \neg I & \textit{otherwise} \end{cases}$$

- We reduce our trust on DLIS by adding randomness in DLIS if the difference in  $|I|$  and  $|\neg I|$  is not overwhelmingly high

---

<sup>6</sup>Sang et al., 2005

- GANAK<sup>7</sup>: First Scalable Probabilistic Exact Model Counter
- Given a propositional formula  $F$  (CNF) and confidence  $\delta \in (0, 1]$  GANAK( $F, \delta$ ) returns count such that

$$\Pr[|Sol(F)| = \text{count}] \geq 1 - \delta$$

- Tool is available at: <https://github.com/meelgroup/ganak>

---

<sup>7</sup>GANAK (गणक in Sanskrit) refers to a device that counts

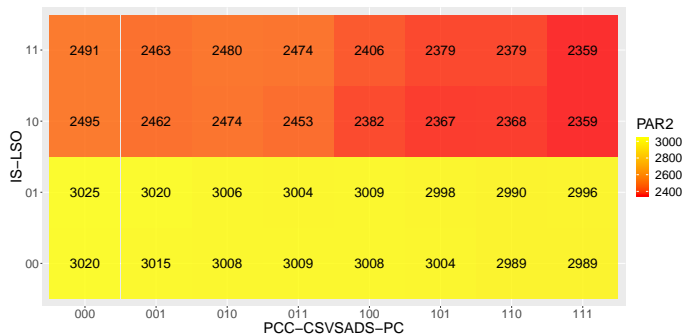
- Benchmarks arising from probabilistic reasoning, plan recognition, DQMR networks, ISCAS89 combinatorial circuits, quantified information flow, etc

# Experimental Evaluation

- Benchmarks arising from probabilistic reasoning, plan recognition, DQMR networks, ISCAS89 combinatorial circuits, quantified information flow, etc
- Objectives:
  - ① Study the impact of different configurations of heuristics
  - ② Study the performance of GANAK with respect to the state-of-the-art model counters
- $\delta = 0.05$ , Total Memory = 4 GB, Component Cache Size = 2 GB, Timeout=5000 secs

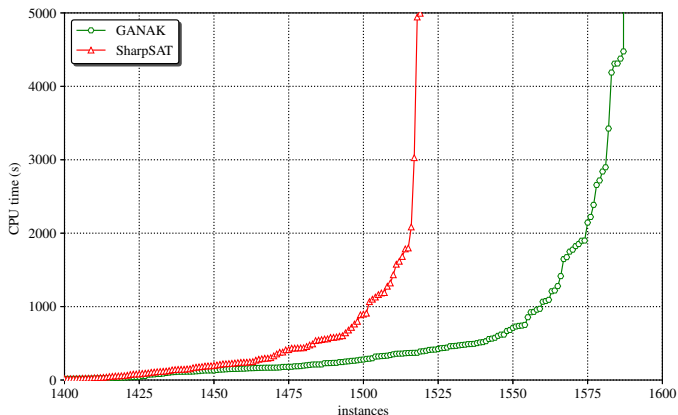


# Experimental Evaluation: Individual Analysis



- GANAK performed best when all the heuristics are turned on

# Experimental Evaluation: Comparison with other tools



- In our experiments, the model count returned by GANAK was equal to the exact model count for all benchmarks

# Thank You

Tool is available at: <https://github.com/meelgroup/ganak>

