GANAK: A Scalable Probabilistic Exact Model Counter

Shubham Sharma[1], Mate Soos[2], Subhajit Roy[1] and Kuldeep S. Meel[2]

[1]Indian Institute of Technology Kanpur, India
[2]National University of Singapore

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

Propositional Model Counting

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

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

[1] Chakraborty et al., 2019

Propositional Model Counting

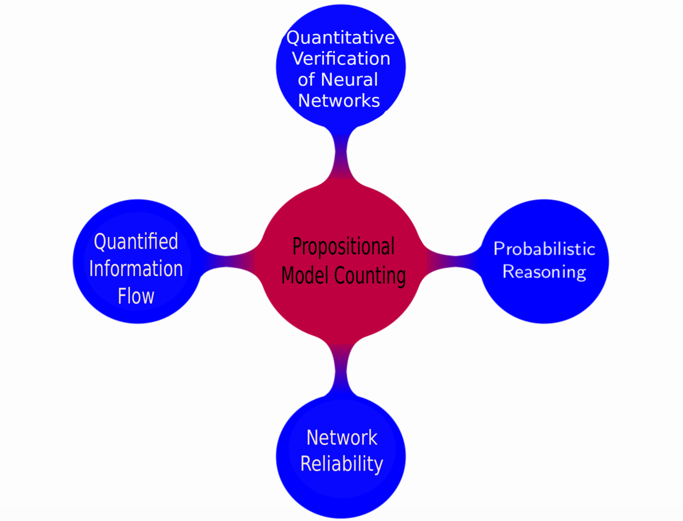
- Probabilistic Exact Model Counting
  - Given a propositional formula F (CNF) and confidence $\delta \in (0, 1]$, counter returns `count` such that:

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

- Probabilistic Exact Model Counting is almost as hard as Exact Model Counting[1]

[1] Chakraborty et al., 2019

Applications of Propositional Model Counting



Main Ingredients

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

Main Ingredients

- Decision Process:
  - $(F \wedge l) \vee (F \wedge \neg l)$ mutually inconsistent
  - $\#(F) = \#(F \wedge l) + \#(F \wedge \neg l)$
- 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:
  - $(F \wedge l) \vee (F \wedge \neg l)$ mutually inconsistent
  - $\#(F) = \#(F \wedge l) + \#(F \wedge \neg l)$
- 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
Δ_1	$\#(\Delta_1)$
Δ_2	$\#(\Delta_2)$

Main Ingredients

- Decision Process:
  - $(F \wedge l) \vee (F \wedge \neg l)$ mutually inconsistent
  - $\#(F) = \#(F \wedge l) + \#(F \wedge \neg l)$
- 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
Δ_1	$\#(\Delta_1)$
Δ_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

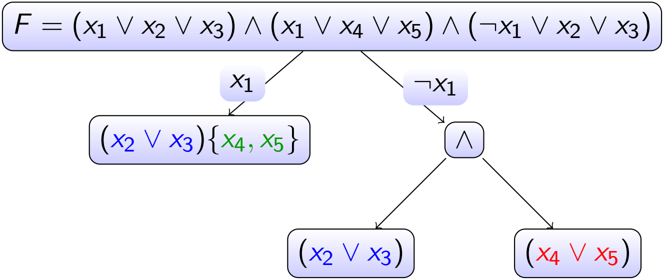
$$F = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_4 \lor x_5) \land (\neg x_1 \lor x_2 \lor x_3)$$

$$\downarrow x_1$$

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

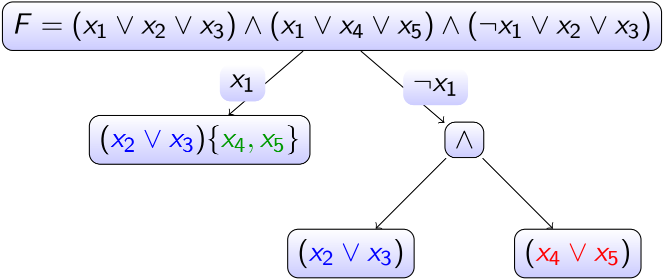
Key	Value
$(x_2 \lor x_3)$	3
$(x_2 \lor x_3)\{x_4, x_5\}$	12

Example



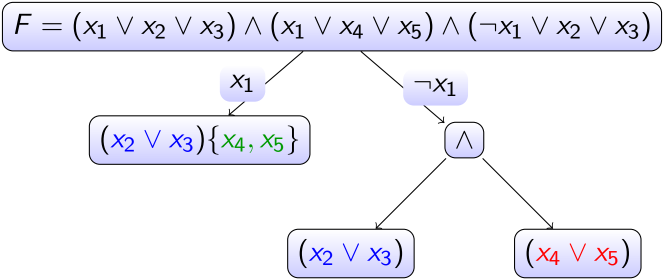
Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12

Example



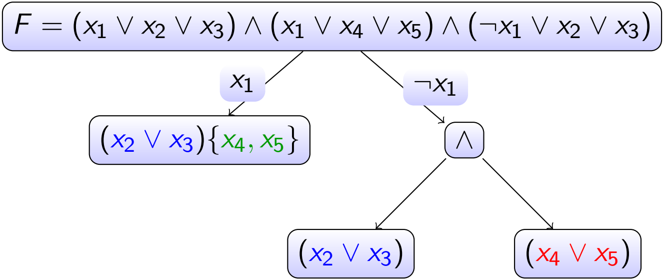
Key	Value
$(x_2 \vee x_3)$	3
$(x_2 \vee x_3)\{x_4, x_5\}$	12
$(x_4 \vee x_5)$	3

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

Our Contribution

1. **Probabilistic Component Caching (PCC)**
2. **Variable Branching Heuristic (CSVSADS)**
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[2]	-3, -5, 6, 0, -1, 4, -6, 0, 2, 3, 6, 0	$\#(F)$
HC[3]	1, 2, 3, 4, 5, 6, 1, 2, 3	$\#(F)$
GANAK	Hash of HC/STD	$\#(F)$

[2] Sang et al., 2005
[3] Thurley, 2006

Variable Branching Heuristic (CSVSADS)

- Score(VSADS)[4] $= \underline{\text{p} \times \text{Score(VSIDS)}} + \underline{\text{q} \times \text{Score(DLCS)}}$
  - VSIDS: Prioritize variables present in recently generated conflict clauses
  - DLCS: Prioritize the highest occurring variable in the residual formula

[4] Sang et al., 2005

Variable Branching Heuristic (CSVSADS)

- Score(VSADS)[4] $=$ $\underline{\text{p} \times \text{Score(VSIDS)}}$ $+$ $\underline{\text{q} \times \text{Score(DLCS)}}$
  - VSIDS: Prioritize variables present in recently generated conflict clauses
  - DLCS: Prioritize the highest occurring variable in the residual formula
- Score(CSVSADS) $=$ $\underline{\alpha \times \text{CacheScore}}$ $+$ $\underline{\beta \times \text{Score(VSADS)}}$
  - Prioritize variables not present in the components which are recently added to the cache

[4] Sang et al., 2005

Phase Selection Heuristic (PC)

$$\text{DLIS}^5 \begin{cases} l & |l| \geq |\neg l| \\ \neg l & \textit{otherwise} \end{cases}$$

[5] Sang et al., 2005

Phase Selection Heuristic (PC)

$$\text{DLIS}^5 \begin{cases} l & |l| \geq |\neg l| \\ \neg l & \textit{otherwise} \end{cases}$$

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

[5] Sang et al., 2005

Tool

- GANAK[6]: First Scalable Probabilistic Exact Model Counter
- Given a propositional formula F (CNF) and confidence $\delta \in (0, 1]$ GANAK(F, δ) returns count such that

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

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

[6] GANAK (गणक in Sanskrit) refers to a device that counts

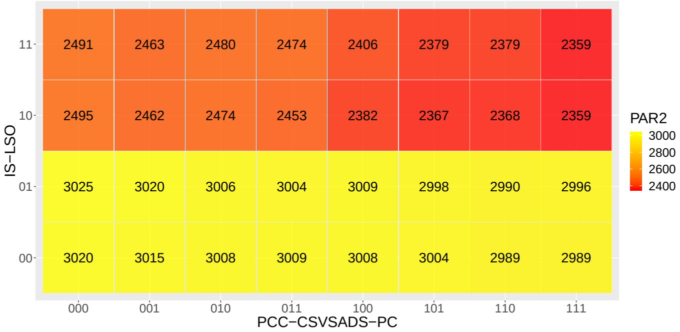
Experimental Evaluation

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

Experimental Evaluation

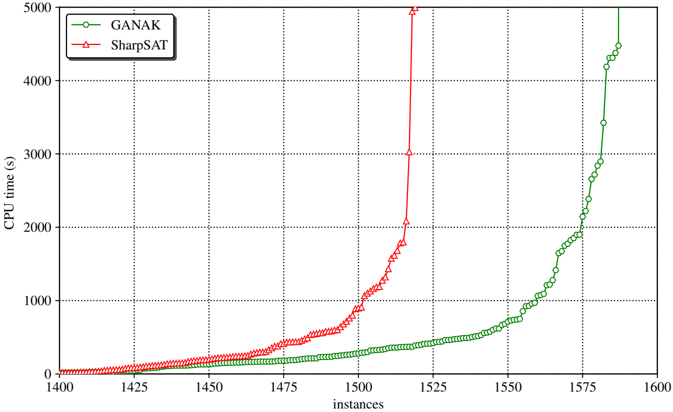
- Benchmarks arising from probabilistic reasoning, plan recognition, DQMR networks, ISCAS89 combinatorial circuits, quantified information flow, etc
- Objectives:
  1. Study the impact of different configurations of heuristics
  2. Study the performance of GANAK with respect to the state-of-the-art model counters

Experimental Evaluation: Individual Analysis



- GANAK performed best when all the heuristics are turned on

Experimental Evaluation: Comparison with other tools



- $\delta = 0.05$, Component Cache Size = 2 GB, Timeout=5000 secs
- 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