On Testing of Horn Samplers

Masters Dissertation

Uddalok Sarkar
Supervision: Sourav Chakraborty

Indian Statistical Institute

July 14, 2022

What is Sampler?

- Probabilistic reasoning is the core of most of the problems in practice.
- Probabilistic reasoning techniques rely highly on sampling techniques.
- We need to generate quality samples to support such probabilistic algorithms.

What is Sampler?

Definition

A Sampler is a randomized algorithm $\mathscr{A}$ that takes in an input set (or, multi-set), say S, and outputs $s \in S$ following some distribution $\mathscr{D}_{\mathscr{A}}$.

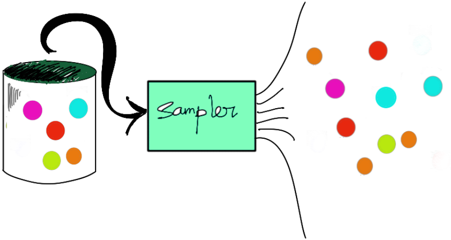

Figure 1: Sampler $\mathscr{A}$

Why testing a Sampler?

- Most of the sampling techniques are based upon Monte Carlo Markov Chain and variational approximation.

Why testing a Sampler?

- Most of the sampling techniques are based upon Monte Carlo Markov Chain and variational approximation.
- These techniques fail to scale to large real-world instances.

Why testing a Sampler?

- Most of the sampling techniques are based upon Monte Carlo Markov Chain and variational approximation.
- These techniques fail to scale to large real-world instances.
- So go for *Heuristic*!

Why testing a Sampler?

- Most of the sampling techniques are based upon Monte Carlo Markov Chain and variational approximation.
- These techniques fail to scale to large real-world instances.
- So go for *Heuristic*!

Why testing a Sampler?

- Most of the sampling techniques are based upon Monte Carlo Markov Chain and variational approximation.
- These techniques fail to scale to large real-world instances.
- So go for *Heuristic*!

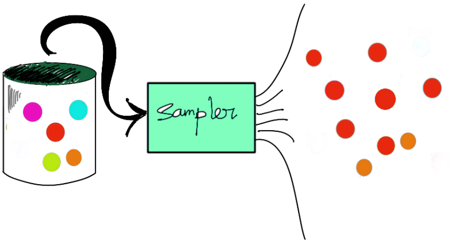

Figure 2: A Bad Sampler

Horn Sampler

Horn Clause

Horn Clause is a clause with atmost one positive (non-negated) literal. e.g., $(\neg x \vee y)$ is a Horn clause; whereas $(x \vee y)$ is not.

Horn Formula

Horn formula is a formula whose all the clauses are Horn clauses.

Horn Sampler

Why Horn Sampler?

- Horn Clause is simple yet powerful. Hence heavy uses.

Horn Sampler

Why Horn Sampler?

- Horn Clause is simple yet powerful. Hence heavy uses.
- Horn Clause forms the basis of logic programming, automated theorem proving etc.

Horn Sampler

Why Horn Sampler?

- Horn Clause is simple yet powerful. Hence heavy uses.
- Horn Clause forms the basis of logic programming, automated theorem proving etc.
- On practical scenario, most of the topological networks, power transmission lines, telecomm etc are modelled using Horn clauses.

Horn Sampler

Horn Sampler Testing

- Horn formula is a restrictive class of CNF formula.
- The satisfiability question of Horn formula (HORNSAT) is $P - Complete$.
- This gives us an intuitive understanding that, the samplers designed solely for Horn formulae, can't deal with general CNF clauses.

Horn Sampler

Horn Sampler Testing

- *UniGen*, developed by Meelgroup is an almost uniform generator for the CNF class of formulas with theoretical guarantees. So a natural question is:

  Does UniGen also works well for Horn class of formulas?

- Another important question was to check whether

  the samplers failing for CNF class work well for Horn?

Naturally, to answer such questions we need a tester for the Horn sampler Class.

Contribution of this Thesis

- Before this work, *Shayak Chakraborty* established a primary work by developing the algorithm of `Flash` for Uniform-Horn-sampler-tester in his thesis work in 2019.

Contribution of this Thesis

- Before this work, *Shayak Chakraborty* established a primary work by developing the algorithm of `Flash` for Uniform-Horn-sampler-tester in his thesis work in 2019.
- But there were some gaps in the proofs which we close in this thesis and reduce down the complexity from previous version.

Contribution of this Thesis

- Before this work, *Shayak Chakraborty* established a primary work by developing the algorithm of `Flash` for Uniform-Horn-sampler-tester in his thesis work in 2019.
- But there were some gaps in the proofs which we close in this thesis and reduce down the complexity from previous version.
- We extend the work to develop a Horn Sampler Tester for general case, that is, a Weighted-Horn-sampler-tester.

Contribution of this Thesis

- Before this work, *Shayak Chakraborty* established a primary work by developing the algorithm of `Flash` for Uniform-Horn-sampler-tester in his thesis work in 2019.
- But there were some gaps in the proofs which we close in this thesis and reduce down the complexity from previous version.
- We extend the work to develop a Horn Sampler Tester for general case, that is, a Weighted-Horn-sampler-tester.
- We further provide a prototype implementation of `Flash` and `wFlash` and the empirical results over three state-of-the-art samplers on a set of benchmarks.

Contribution of this Thesis

- Before this work, *Shayak Chakraborty* established a primary work by developing the algorithm of `Flash` for Uniform-Horn-sampler-tester in his thesis work in 2019.
- But there were some gaps in the proofs which we close in this thesis and reduce down the complexity from previous version.
- We extend the work to develop a Horn Sampler Tester for general case, that is, a Weighted-Horn-sampler-tester.
- We further provide a prototype implementation of `Flash` and `wFlash` and the empirical results over three state-of-the-art samplers on a set of benchmarks.
- This work has been submitted in NeurIPS-2022.

Verification of Sampler

Complexity issues

Tightness of Birthday Paradox

Suppose n number of samples are randomly chosen from a distribution on $[k]$ with $n \leq k$. Then if wish to see some repetition the obtained samples then we have,

$$n = \Theta(\sqrt{k})$$

Verification of Sampler

Complexity issues

Tightness of Birthday Paradox

Suppose n number of samples are randomly chosen from a distribution on $[k]$ with $n \leq k$. Then if wish to see some repetition the obtained samples then we have,

$$n = \Theta(\sqrt{k})$$

- This motivates the result: If we use only random samples from the distribution and with probability $\geq 1 - \delta$ accepts a distribution, $\epsilon-$close to uniform and $\leq \delta$ rejects an η-far from uniform would require,

$$\Omega\left(\sqrt{R_\varphi}\frac{\log \delta^{-1}}{(\eta - \epsilon)^2}\right)$$

Conditional Sampling

Conditional Sampling

Suppose Σ is our domain of the distribution $\mathscr{D}$. If one *assumes* conditional sampling access, i.e., sampling from $S \subset \Sigma$ then ϵ-close and η-far $\mathscr{D}$ can be accepted with sample complexity

$$\mathcal{O}\left(\frac{\log \delta^{-1}}{(\eta - \epsilon)^2}\right)$$

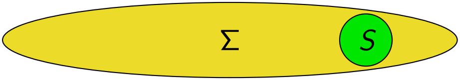

Figure 3: If we assume a small and constant size of S then sample complexity should be independent of $|R_\varphi|$

Weighted-Horn-sampler

Definition (Weight function)

A weight function $wt : \{0,1\}^S \rightarrow (0,1)$ assigns weight to each assignment that can be formed using the set S of Boolean variables.

Weight function gives rise to a distribution on the set of satisfying assignments of φ, that is, R_φ.

Weighted-Horn-sampler

Definition (Weighted-Horn-sampler)

A Weighted-Horn-sampler $\mathscr{G}$ is a randomized algorithm that can output samples from R_φ according to a weight function wt.

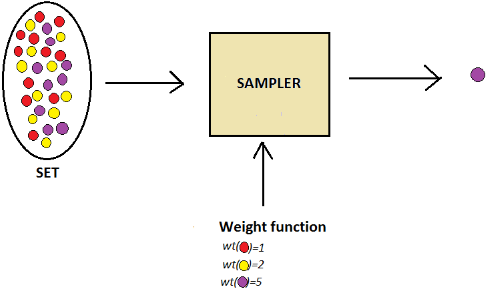

Figure 4: Weighted-Horn-sampler

Ideal Horn Samplers

- A Horn sampler $\mathcal{I}_{\mathscr{W}}(\varphi, S, wt)$ is said to be an **ideal Weighted-Horn-sampler** with respect to the weight function wt, if

$$\forall \sigma \in R_{\varphi}, \quad \mathbb{P}\left[\mathcal{I}_{\mathscr{W}}(\varphi) = \sigma\right] = \frac{wt(\sigma)}{\sum_{\sigma_1 \in R_{\varphi}} wt(\sigma_1)}$$

- The ideal Weighted-Horn-sampler is said to be an **ideal Uniform-Horn-sampler** when the weight function wt is uniform, that is,

$$wt(\sigma) = \frac{1}{|R_{\varphi}|}$$

for all σ.

ε-closeness and η-farness

- ε-**closeness**: $\mathscr{G}$ is said to be $\mathscr{I}_{\mathscr{W}}$, if for all Horn-formula φ and $\sigma \in R_\varphi$
  $(1-\varepsilon)\mathbb{P}\left[\mathscr{I}_{\mathscr{W}}(\varphi) = \sigma\right] \leq \mathbb{P}\left[\mathscr{G}(\varphi) = \sigma\right] \leq (1+\varepsilon)\mathbb{P}\left[\mathscr{I}_{\mathscr{W}}(\varphi) = \sigma\right].$

  Note, ε-closeness is defined in terms of ℓ_∞ norm.

  If $\mathscr{G}$ is ε-close to the ideal Uniform-Horn-sampler, then $\mathscr{G}$ is called an ε-Additive Almost Uniform-Horn-sampler (**AAU**).

ε-closeness and η-farness

- ε-**closeness**: $\mathcal{G}$ is said to be $\mathcal{I}_{\mathcal{W}}$, if for all Horn-formula φ and $\sigma \in R_\varphi$
  $(1-\varepsilon)\mathbb{P}[\mathcal{I}_{\mathcal{W}}(\varphi) = \sigma] \leq \mathbb{P}[\mathcal{G}(\varphi) = \sigma] \leq (1+\varepsilon)\mathbb{P}[\mathcal{I}_{\mathcal{W}}(\varphi) = \sigma]$.

  Note, ε-closeness is defined in terms of ℓ_∞ norm.

  If $\mathcal{G}$ is ε-close to the ideal Uniform-Horn-sampler, then $\mathcal{G}$ is called an ε-Additive Almost Uniform-Horn-sampler (**AAU**).

- η-**farness**: On the other hand, $\mathcal{G}$ is said to be η-far from $\mathcal{I}_{\mathcal{W}}$ with respect to some Horn-formula φ if

$$\sum_{\sigma \in R_\varphi} |\mathbb{P}[\mathcal{G}(\varphi) = \sigma] - \mathbb{P}[\mathcal{I}_{\mathcal{W}}(\varphi) = \sigma]| \geq \eta$$

  Note, ε-closeness is defined in terms of ℓ_1 norm.

Chain Formula

Let $m > 0$ be a natural number and $k < 2^m$ be a positive **odd** number.
Let $c_1 c_2 \ldots c_m$ be the m-bit representation of k, where c_m is the Least Significant Bit (LSB) in the representation of m.

If $c_j = 1$ then C_j is "$\vee$', else if $c_j = 0$, then C_j is "$\wedge$".

The chain formula $\psi_{k,m}$ is defined as:

$$\psi_{k,m}(a_1, a_2, \ldots, a_m) = a_1 C_1 (a_2 C_2 (\ldots (a_{m-1} C_{m-1} a_m) \ldots)$$

where $a_1, a_2, \ldots, a_m$ are variables, has exactly k many satisfying assignments.

Chain Formula

Example

For $k = 7$ and $m = 5$, as the binary representation of 7 in 5 bits is 00111, the corresponding chain formula would be

$$\psi_{k,m}(a_1, a_2, a_3, a_4, a_5) = (a_1 \wedge (a_2 \wedge (a_3 \vee (a_4 \vee a_5))))$$

note that, $|R_{\psi_{7,5}}| = 7$.

$$R_{\psi_{7,5}} = \{11111, 11110, 11101, 11100, 11011, 11010, 11001\}$$

General framework for Testing Uniform Samplers

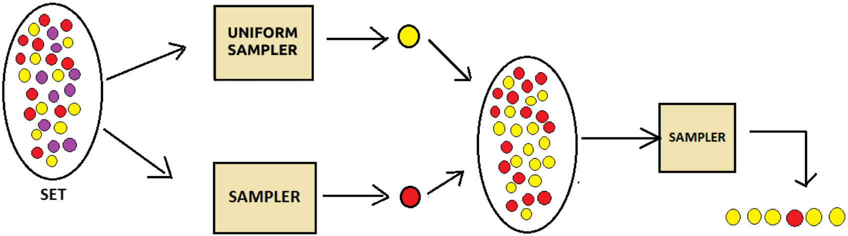

Figure 5: Uniform sampler Tester measures whether $\mathbb{P}(\bullet) \approx \mathbb{P}(\bullet)$ in the obtained samples from testing sampler

Barbarik

- `Barbarik` was proposed by S. Chakraborty and K. S. Meel in the context of testing samplers associated Conjunctive Normal Form (CNF) formulae.

- Adapting `Barbarik` to suit *Horn formula* is not straight forward.

Framework of Barbarik

- Given a CNF formula φ, Barbarik samples two assignments σ_1 and σ_2 of φ using uniform sampler and sampler under test.
- Barbarik employs the conditioning using $\varphi \wedge (\sigma_1 \vee \sigma_2)$
- Barbarik boosts up the conditioned sample space by employing Chain formula

$$\varphi' := \varphi \wedge (\sigma_1 \vee \sigma_2) \wedge \bigwedge_{l} ((l \rightarrow \psi_{k,m}(V)) \wedge (\neg l \rightarrow \psi_{k,m}(V)))$$

where $l \sim (\sigma_1 \setminus \sigma_2) \cup (\sigma_2 \setminus \sigma_1)$ and V is the set of new variables not appearing in φ.

Issues with Barbarik for Horn Formula

- $(\sigma_1 \vee \sigma_2)$ is NOT a Horn Formula.
- It is NOT possible to design a Horn Formula using σ_1 and σ_2 such that the resulting satisfying assignments are σ_1 and σ_2.
- The formulae $(I \rightarrow \psi_{k,m}(V))$ are not Horn always.

Our Results

- We developed our Uniform-Horn-sampler-tester `Flash` that checks whether a given Horn sampler $\mathscr{G}$ is ε-close to Uniform-Horn-sampler or η-far from Uniform-Horn-sampler.

Our Results

- We developed our Uniform-Horn-sampler-tester `Flash` that checks whether a given Horn sampler $\mathscr{G}$ is ε-close to Uniform-Horn-sampler or η-far from Uniform-Horn-sampler.

- We extended our work to develop Weighted-Horn-sampler tester `wFlash` that checks whether a given weighted Horn sampler $\mathscr{G}$ is ε-close to Weighted-Horn-sampler or η-far from Weighted-Horn-sampler.

1. Introduction
2. Formal Definitions
3. Previous Works
4. Our Results
   - Uniform-Horn-sampler-tester: `Flash`
   - Weighted-Horn-sampler-tester: `wFlash`
5. Evaluation Results

Flash

Tester for Uniform-Horn-sampler

Flash takes as input

(i) a black-box Horn sampler $\mathscr{G}$,

(ii) a Horn formula φ,

(iii) three parameters $\varepsilon, \eta, \delta$, such that $\varepsilon \in (0, \frac{1}{3}]$, $\eta > 9\varepsilon$, $\delta > 0$.

(iii) $\widetilde{\mathcal{O}}(\frac{1}{\eta(\eta-9\varepsilon)(\eta-3\varepsilon)^2})$ many samples

Flash

Tester for Uniform-Horn-sampler

Flash takes as input

(i) a black-box Horn sampler $\mathscr{G}$,

(ii) a Horn formula φ,

(iii) three parameters $\varepsilon, \eta, \delta$, such that $\varepsilon \in (0, \frac{1}{3}]$, $\eta > 9\varepsilon$, $\delta > 0$.

(iii) $\widetilde{\mathscr{O}}(\frac{1}{\eta(\eta-9\varepsilon)(\eta-3\varepsilon)^2})$ many samples

Flash outputs

(i) ACCEPT with probability at least $1-\delta$, if $\mathscr{G}$ is an ε-AAU Horn-sampler;

(ii) REJECT with probability at least $1-\delta$, if $D_{\mathscr{G}(\varphi)}$, is η-far in ℓ_1 distance from the uniform distribution and $\mathscr{G}$ is **subquery consistent**.

Flash

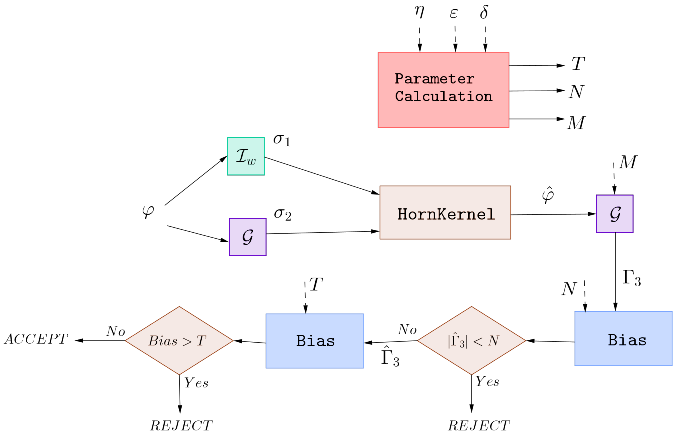

Figure 6: Overview of Flash framework

Flash

subroutines

Flash uses the following subroutines:

(i) HornKernel

(ii) Encode

(iii) Bias

HornKernel

Figure 7: HornKernel block

(1) $|R_{\varphi'}| \geq \tau$.

(2) $Supp(\varphi) \subseteq Supp(\varphi')$.

HornKernel

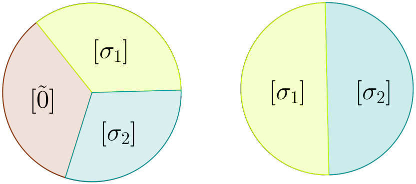

Figure 8: $R_{\varphi'}$ consists of the equivalence classes $[\sigma_1], [\sigma_2], [\tilde{0}]$ or, $[\sigma_1], [\sigma_2]$

$[\sigma_1]$ denotes the set $\{x \mid x \in R_{\varphi'} \;\&\; x_{\downarrow Supp(\varphi)} = \sigma_1\}$

$\tilde{0}$ denote the assignment whose only True literals are the common true literals of σ_1 and σ_2.

HornKernel

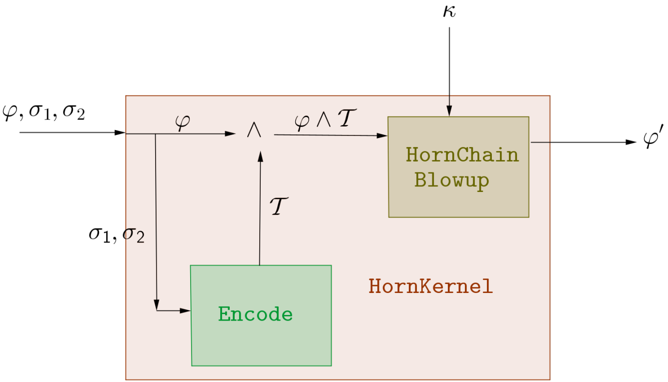

Figure 9: HornKernel

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

Lets define the following sets:

- cmmTrueLits (common true literals) : x_1, x_2
- cmmFalseLits (common false literals) : x_7, x_8
- uncmmLits (literals having different truth-values) : x_3, x_4, x_5, x_6

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

cmmTrueLits

$$x_1\,(x_1 \iff x_2)\ldots$$

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

cmmTrueLits

$$x_1 \, (x_1 \iff x_2) \ldots$$

cmmFalseLits

$$\neg x_7 \, (x_7 \iff x_8) \ldots$$

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

uncmmLits

Let's subdivide this set into two sets:

- $\forall x_i, x_j$ such that $x_i, x_j = 0$ in σ_1 but $x_i, x_j = 1$ in σ_2, we add:

$$(x_i \iff x_j)$$

- $\forall x_i, x_j$ such that $x_i, x_j = 1$ in σ_1 but $x_i, x_j = 0$ in σ_2, we add:

$$(x_i \iff x_j)$$

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

We aren't done, We have to ensure that, whenever x_3 is True, x_5 goes False and vice-versa.

We could have add $(\neg x_3 \iff x_5)$

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

We aren't done, We have to ensure that, whenever x_3 is True, x_5 goes False and vice-versa.

We could have add $(\neg x_3 \iff x_5)$

But

$$(\neg x_3 \implies x_5) \equiv (x_3 \vee x_5)$$

Not a Horn clause

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

So our complete conditioning formula becomes:

$$\varphi' := \varphi \;\; x_1 \, (x_1 \iff x_2) \, \neg x_7 \, (x_7 \iff x_8)$$
$$(x_3 \iff x_4) \, (x_5 \iff x_6)$$
$$(x_3 \implies \neg x_5)$$

Encode

$$\sigma_1 = 11110000$$

$$\sigma_2 = 11001100$$

So our complete conditioning formula becomes:

$$\begin{aligned}\varphi' := \varphi \;\; & x_1 \, (x_1 \iff x_2) \, \neg x_7 \, (x_7 \iff x_8) \\ & (x_3 \iff x_4) \, (x_5 \iff x_6) \\ & (x_3 \implies \neg x_5)\end{aligned}$$

But, note that, if $\tilde{0} = 11000000$ is a satisfying assignment of φ then it is also a satisfying assignment of φ'.

Horn Chain Formula

Definition (Pure Horn Chain Formula)

From a Chain formula $\psi_{k,m}$ we can generate a Horn chain formula $\psi'_{k,m}$ by replacing every literal a_i in $\psi_{k,m}$ by $\psi'_{k,m}$.

$$\psi_{5,4} = a_1 \wedge (a_2 \vee (a_3 \wedge a_4))$$

$$\psi'_{5,4} = \neg a_1 \wedge (\neg a_2 \vee (\neg a_3 \wedge \neg a_4))$$

Horn Chain Formula

Definition (Pure Horn Chain Formula)

From a Chain formula $\psi_{k,m}$ we can generate a Horn chain formula $\psi'_{k,m}$ by replacing every literal a_i in $\psi_{k,m}$ by $\psi'_{k,m}$.

$$\psi_{5,4} = a_1 \wedge (a_2 \vee (a_3 \wedge a_4))$$
$$\psi'_{5,4} = \neg a_1 \wedge (\neg a_2 \vee (\neg a_3 \wedge \neg a_4))$$

$$\psi'_{5,4} \equiv \neg a_1 \wedge (\neg a_2 \vee \neg a_3) \wedge (\neg a_2 \vee \neg a_4)$$

$$|R_{\psi'_{5,4}}| = 7$$

Horn Chain Blowup

- First note that, $(\sigma_1 \setminus \sigma_2) \cup (\sigma_2 \setminus \sigma_1) \neq \emptyset$.
- So we sample some literals $l \sim (\sigma_1 \setminus \sigma_2) \cup (\sigma_2 \setminus \sigma_1)$ and conjunct the following to φ',

$$(l \rightarrow \psi'_{k,m}(V)) \wedge (\neg l \rightarrow \psi'_{k,m}(V))$$

- This blows up our Support set S to $S' = S \cup V$. And blows up the solution space $|R_{\varphi'}|$, such that,

$$|R_{\varphi'} \downarrow_S| = 3 \textit{ or } 2^1$$

[1]depends on whether $\tilde{0} \in R_{\varphi}$

Assumption : Subquery Consistency of Sampler

Let $\widehat{\varphi}$ be the Horn formula obtained from the subroutine `HornKernel` .

A Horn sampler $\mathscr{G}$ is said to be *subquery consistent*, if the output of $\mathscr{G}(\widehat{\varphi})$ are independent samples from the distribution $\mathscr{D}_{\mathscr{G}(\varphi)|X}$, where either $X = \{\sigma_1, \sigma_2\}$ or $X = \{\sigma_1, \sigma_2, \tilde{0}\}$ (depending on whether $\tilde{0} \in R_\varphi$ or not).

Testing Strategy of Flash

- The main strategy of `Flash` is to sample as many witnesses from $R_{\varphi'}$ so that, with high probability the sampler $\mathcal{G}$ can be accepted or rejected.
- Let $Z = |\{x : R_{\varphi'} \downarrow_S = \sigma_1\}|$
- If $\mathcal{G}$ is ε-close to AAU Horn sampler then,

$$\mathbb{E}[Z] \leq L = f(\varepsilon)$$

- If $\mathcal{G}$ is η-far from AAU Horn sampler then,

$$\mathbb{E}[Z] \geq H = f(\varepsilon, \eta)$$

- we fix,

$$T = \frac{L + H}{2}$$

Testing Strategy of Flash

- Flash has to sample N many samples of $[\sigma_1]$ and $[\sigma_2]$, such that, with high probability
  (1) $Z < T$ if $\mathscr{G}$ is AAU Horn Sampler
  (2) $Z > T$ if $\mathscr{G}$ is η-far from Uniform-Horn-sampler
- Flash turns out sampling some samples of $[\tilde{0}]$.
- So, Flash samples M many samples such that, if $\mathscr{G}$ is AAU, it receives atleast N many samples of $[\sigma_1]$ and $[\sigma_2]$.

M, N, T are calculated on the basis of Chernoff bounds.

wFlash

Tester for Weighted-Horn-sampler

wFlash takes as input

(i) a black-box Horn sampler $\mathscr{G}$,

(ii) a Horn formula φ,

(iii) three parameters $\varepsilon, \eta, \delta$, such that $\varepsilon \in (0, \frac{1}{3}]$, $\eta > 9\varepsilon$, $\delta > 0$.

(iv) a weight function *wt*

(iii) $\widetilde{\mathcal{O}}(\frac{\mathtt{tilt}(wt,\varphi)^3}{\eta(\eta-9\varepsilon)(\eta-3\varepsilon)^2})$ many samples

wFlash

Tester for Weighted-Horn-sampler

`wFlash` takes as input

(i) a black-box Horn sampler $\mathcal{G}$,

(ii) a Horn formula φ,

(iii) three parameters $\varepsilon, \eta, \delta$, such that $\varepsilon \in (0, \frac{1}{3}]$, $\eta > 9\varepsilon$, $\delta > 0$.

(iv) a weight function wt

(iii) $\widetilde{\mathcal{O}}(\frac{\mathtt{tilt}(wt,\varphi)^3}{\eta(\eta-9\varepsilon)(\eta-3\varepsilon)^2})$ many samples

`wFlash` outputs

(i) If $\mathcal{G}$ is ε-close to the Weighted-Horn-sampler $\mathcal{I}_{\mathcal{W}}$, `wFlash` outputs ACCEPT with probability at least $1-\delta$.

(ii) If $\mathcal{G}$ is η-far from the Weighted-Horn-sampler $\mathcal{I}_{\mathcal{W}}$, `wFlash` outputs REJECT with probability at least $1-\delta$.

where $\mathtt{tilt}(wt, \varphi)$ denotes the maximum ratio between any two satisfying assignments of φ with respect to the weight function wt.

wFlash

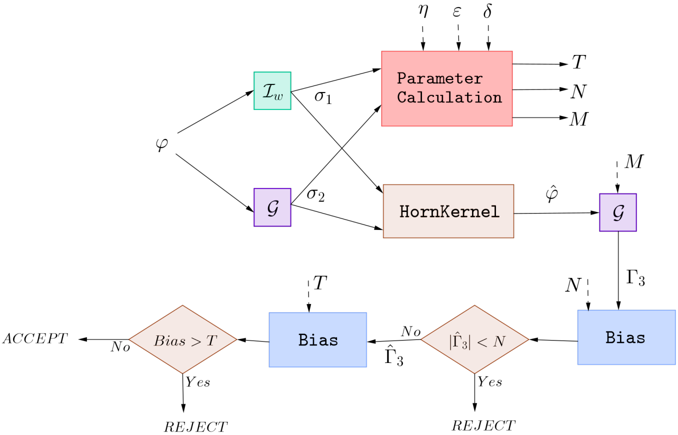

Figure 10: Overview of wFlash framework. T, M, N are dependent of weights of σ_1 and σ_2.

Sampler Tested

We employ the following state of the art samplers:

- UNIGEN3 : Developed by MeelGroup. Has strong theoretical Guarantees.
- QUICKSAMPLER : probably the fastest SAT sampler which generates varied samples from the witness space. Built on top of SMT solver $Z3$.
- STS : uses a simple recursive strategy on the recursion tree to generate samples.

Engineering Challenge - I : Weight Function

- It is not possible to define a explicit weight function wt for large scale R_φ.

- We use **Literal weighted function** as follows,

$$wt(\sigma) = \prod_{x \in \sigma} \begin{cases} W(x), & \text{if } \; x = 1 \\ 1 - W(x), & \text{if } \; x = 0 \end{cases}$$

where, $W : S \to (0, 1)$

Engineering Challenge - II : Inverse Transform Sampling

- Existing Samplers are not capable of handling weight functions.
- Literal weighted functions can be handled by employing inverse sampling box before a sampler.
- Given a φ and wt, an inverse transform sampling come up with a new formula $\hat{\varphi}$ with $S \subset \hat{S}$, such that,

$$\mathbb{P}_{\mathcal{I}_{\mathcal{U}}}(\hat{\varphi}, S, \sigma) = \frac{wt(\sigma)}{\sum\limits_{\sigma_1 \in R_{\varphi}} wt(\sigma_1)}$$

Flash Evaluations

	UniGen		QuickSampler		STS	
Benchmark	o/p	#Samples	o/p	#Samples	o/p	#Samples
Net6_count_91	A	218505	R	52025	R	20810
Net8_count_96	A	218505	R	166480	R	31215
Net12_count_106	A	218505	R	72835	R	52025
Net22_count_116	A	218505	R	72835	R	41620
Net27_count_118	A	218505	R	72835	R	10405
Net29_count_164	A	218505	R	114455	R	20810
Net39_count_240	A	218505	R	114455	R	114455
Net43_count_243	A	218505	R	93645	R	114455
Net46_count_322	A	218505	R	10405	R	10405
Net52_count_362	A	218505	R	10405	R	20810
Net53_count_339	A	218505	R	31215	R	72835

Table 1: Evaluation results of Flash [2] [3]

[2] Benchmark consist of formulas arising from the reliability computation of power transmission networks in US cities

[3] Experiments carried out on a cluster with each node consists of E5-2690 v3 @2.60GHz CPU with 24 cores and 4GB memory per core.

wFlash Evaluations

	wUniGen		wQuickSampler		wSTS	
Benchmark	o/p	#Samples	o/p	#Samples	o/p	#Samples
Net6_count_91_w2	A	274175	R	17667	R	26995
Net8_count_96_w2	A	397169	A	388885	R	16385
Net12_count_106_w2	A	197713	R	6085	R	5930
Net22_count_116_w2	A	302546	R	22947	R	24561
Net27_count_118_w2	TLE	-	R	10405	R	26245
Net29_count_164_w2	A	238673	R	7226	R	17706
Net39_count_240_w2	A	282138	R	13690	R	14885
Net43_count_243_w2	TLE	-	R	238260	R	9217
Net46_count_322_w2	A	437529	R	135368	R	30819
Net52_count_362_w2	TLE	-	R	210925	R	23127
Net53_count_339_w2	A	191806	R	8650	R	9605

Table 2: Evaluation results of wFlash [4]

[4] wUniGen is UniGen Preceded by inverse sampling transform and same for others.

Conclusions

- To best of our knowledge `Flash` and `wFlash` are the first testing frameworks for checking reliability of the Uniform-Horn-sampler and Weighted-Horn-sampler .
- We are glad to announce that we have been able to submit the work in NeurIPS-2022 and waiting for the review.
- Apart from Horn, the other classes of CNF like 2-SAT, Dual-Horn and some non-CNF classes like XOR-CNF are of keen interest in various elds. Thus coming up with testing frameworks exclusively for such classes could give a new direction to this research.

Thank you!