Advanced Algorithms (X)

Shanghai Jiao Tong University

Chihao Zhang

May 11, 2020

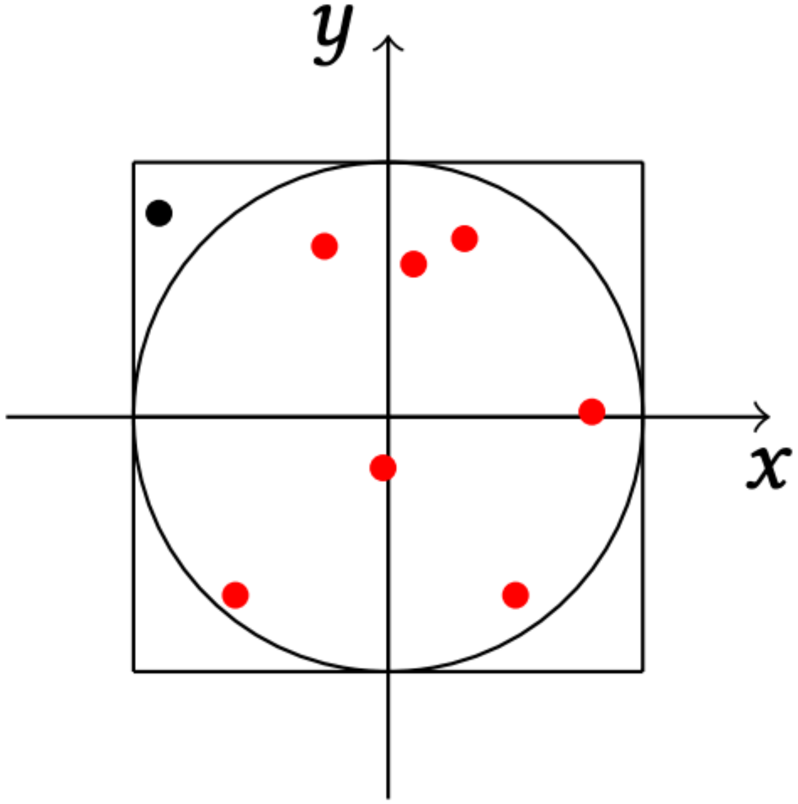
Estimate π

Estimate π

One can design a Monte-Carlo algorithm to estimate the value of π

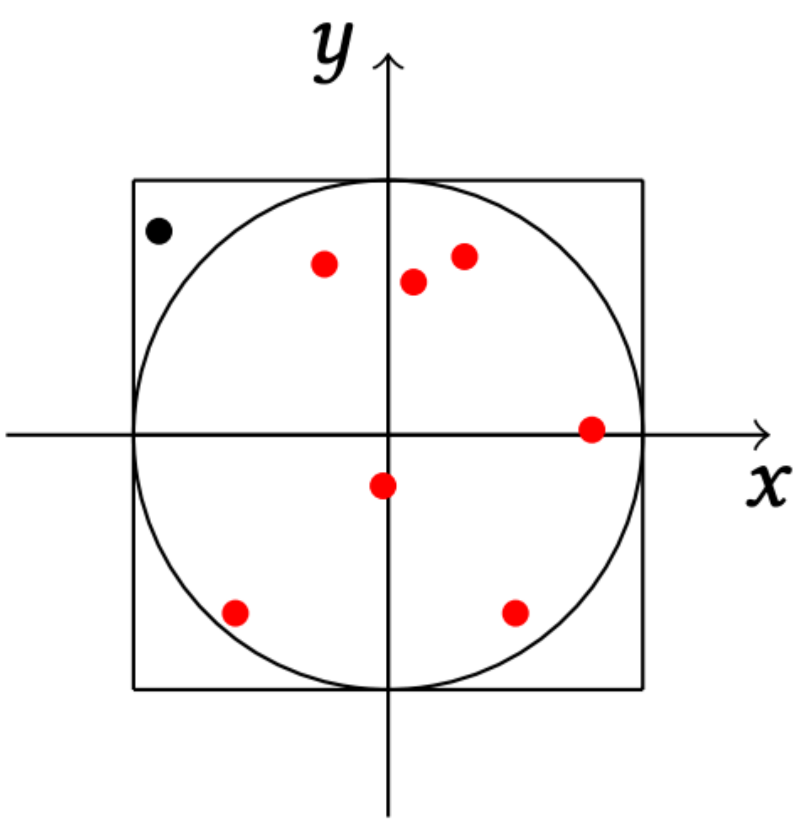
Estimate π

One can design a Monte-Carlo algorithm to estimate the value of π



Estimate π

One can design a Monte-Carlo algorithm to estimate the value of π



- $X_i \in [-1,1] \times [-1,1]$
- $Z_n = \sum_{i=1}^{n} \mathbf{1}[\|X_i\| \leq 1]$

$$X_i \sim \mathrm{Ber}\left(\frac{\pi}{4}\right), \quad \mathbf{E}[Z_n] = \frac{\pi}{4} \cdot n$$

$$X_i \sim \mathrm{Ber}\left(\frac{\pi}{4}\right), \quad \mathbf{E}[Z_n] = \frac{\pi}{4} \cdot n$$

Therefore, by Chernoff bound

$$X_i \sim \mathrm{Ber}\left(\frac{\pi}{4}\right), \quad \mathbf{E}[Z_n] = \frac{\pi}{4} \cdot n$$

Therefore, by Chernoff bound

$$\Pr\left[\left|Z_n - \frac{\pi}{4} \cdot n\right| \geq \varepsilon \cdot \frac{\pi}{4} \cdot n\right] \leq 2\exp\left(-\frac{\varepsilon^2 \pi n}{12}\right)$$

$$X_i \sim \mathrm{Ber}\left(\frac{\pi}{4}\right), \quad \mathbf{E}[Z_n] = \frac{\pi}{4} \cdot n$$

Therefore, by Chernoff bound

$$\Pr\left[\left|Z_n - \frac{\pi}{4} \cdot n\right| \geq \varepsilon \cdot \frac{\pi}{4} \cdot n\right] \leq 2\exp\left(-\frac{\varepsilon^2 \pi n}{12}\right)$$

If $n \geq \dfrac{12}{\varepsilon^2 \pi} \log \dfrac{2}{\delta}$, we have an $1 \pm \varepsilon$ approximation of π with probability at least $1 - \delta$

Rejection Sampling

Rejection Sampling

The method is often called rejection sampling

Rejection Sampling

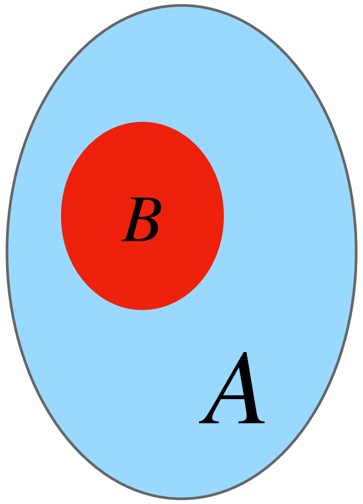
The method is often called rejection sampling

It is useful to estimate the size of some good sets in a large set

Rejection Sampling

The method is often called rejection sampling

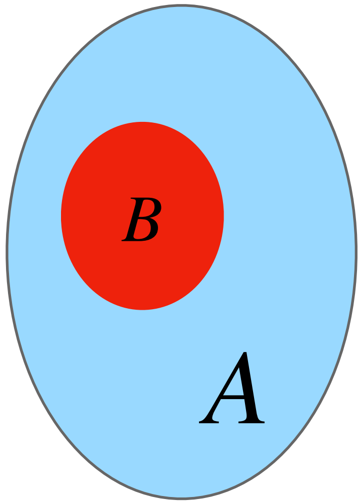
It is useful to estimate the size of some good sets in a large set



Rejection Sampling

The method is often called rejection sampling

It is useful to estimate the size of some good sets in a large set



The number of samples is proportional to $\frac{|A|}{|B|}$

Counting DNF

Counting DNF

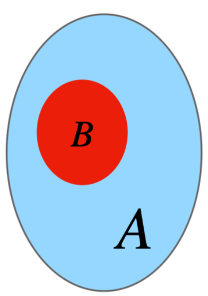
A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m$,

Counting DNF

A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m,\ C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$

Counting DNF

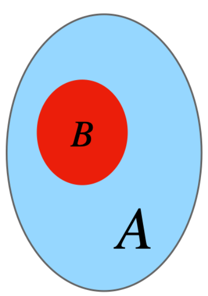
A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m,\ C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$



The number of samples is proportional to $\dfrac{|A|}{|B|}$

Counting DNF

A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m, \; C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$

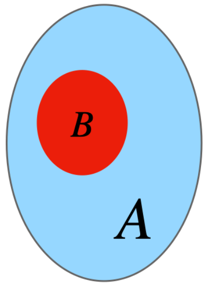


The number of samples is proportional to $\frac{|A|}{|B|}$

$B =$ satisfying assignments

Counting DNF

A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m,\ C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$



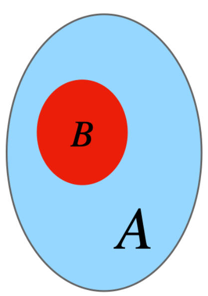
The number of samples is proportional to $\frac{|A|}{|B|}$

$B =$ satisfying assignments

$A =$ all assignments

Counting DNF

A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m,\ C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$



The number of samples is proportional to $\frac{|A|}{|B|}$

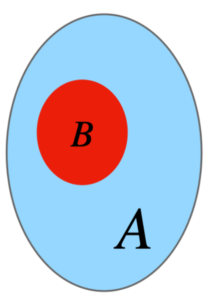
B = satisfying assignments

A = all assignments

φ may contain only polynomial many solutions

Counting DNF

A DNF formula $\varphi = C_1 \vee C_2 \vee \cdots \vee C_m,\ C_i = \bigwedge_{j=1}^{\ell_i} x_{ij}$



The number of samples is proportional to $\frac{|A|}{|B|}$

B = satisfying assignments

A = all assignments

φ may contain only polynomial many solutions

The Monte Carlo method using rejection sampling is slow!

For each clause C_i, define the set

For each clause C_i, define the set

$$S_i := \text{the set of assignments satisfying } C_i$$

For each clause C_i, define the set

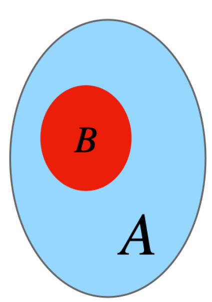
$$S_i := \text{the set of assignments satisfying } C_i$$

We want to estimate $\left| \bigcup_{1 \le i \le m} S_i \right|$

For each clause C_i, define the set

$$S_i := \text{the set of assignments satisfying } C_i$$

We want to estimate $\left| \bigcup_{1 \le i \le m} S_i \right|$

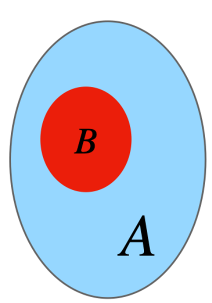


The number of samples is proportional to $\frac{|A|}{|B|}$

For each clause C_i, define the set

$$S_i := \text{the set of assignments satisfying } C_i$$

We want to estimate $$\left| \bigcup_{1 \le i \le m} S_i \right|$$



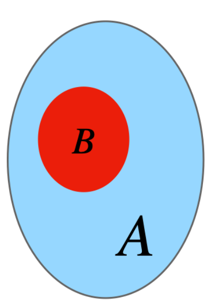
The number of samples is proportional to $\frac{|A|}{|B|}$

$$B = \bigcup_{1 \le i \le m} S_i$$

For each clause C_i, define the set

$$S_i := \text{the set of assignments satisfying } C_i$$

We want to estimate $\left| \bigcup_{1 \le i \le m} S_i \right|$



The number of samples is proportional to $\frac{|A|}{|B|}$

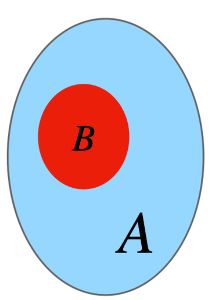
$$B = \bigcup_{1 \le i \le m} S_i$$

$$A = \dot{\bigcup_{1 \le i \le m}} S_i$$

For each clause C_i, define the set

$$S_i := \text{ the set of assignments satisfying } C_i$$

We want to estimate $\left| \bigcup_{1 \le i \le m} S_i \right|$



The number of samples is proportional to $\frac{|A|}{|B|}$

$$B = \bigcup_{1 \le i \le m} S_i$$

$$A = \dot{\bigcup_{1 \le i \le m}} S_i$$

(disjoint union)

How about CNF?

How about CNF?

We consider a very special case: monotone 2-CNF

How about CNF?

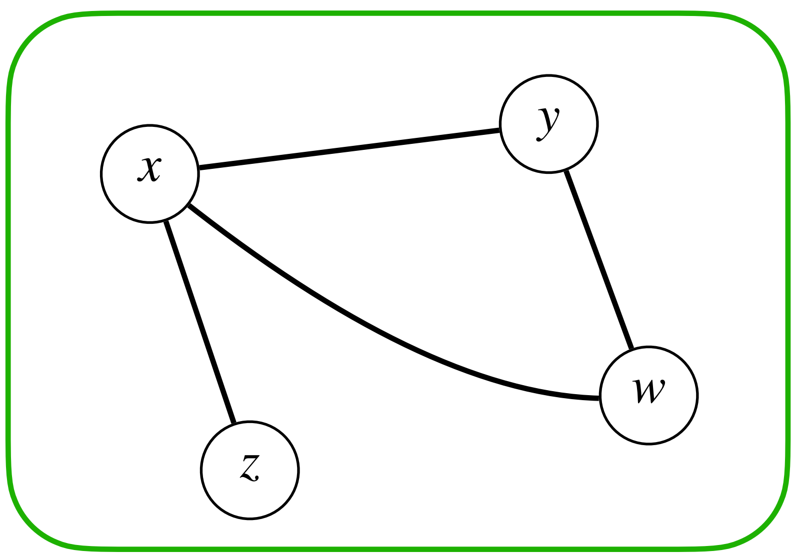
We consider a very special case: monotone 2-CNF

$$\varphi = (x \vee y) \wedge (x \vee z) \wedge (x \vee w) \wedge (y \vee w)$$

How about CNF?

We consider a very special case: monotone 2-CNF

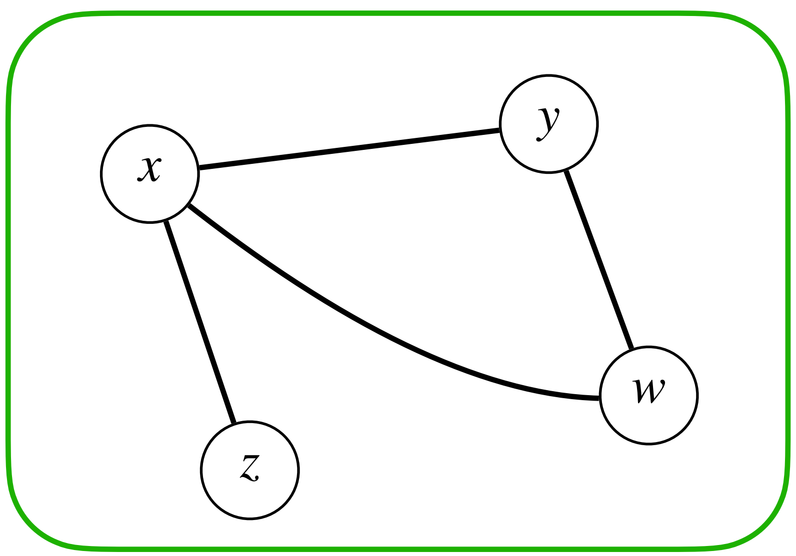
$$\varphi = (x \vee y) \wedge (x \vee z) \wedge (x \vee w) \wedge (y \vee w)$$



How about CNF?

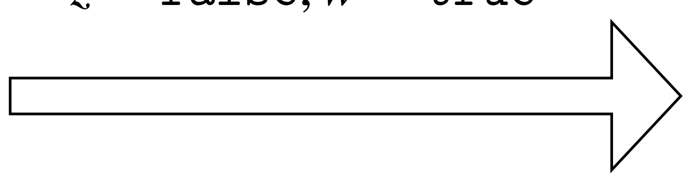
We consider a very special case: monotone 2-CNF

$$\varphi = (x \vee y) \wedge (x \vee z) \wedge (x \vee w) \wedge (y \vee w)$$



$x = \texttt{true}, y = \texttt{false}$

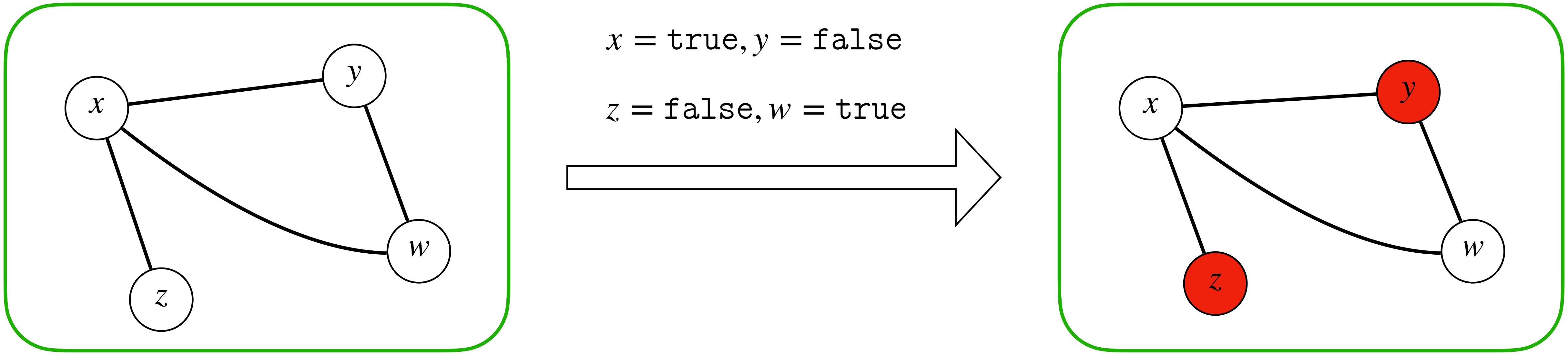
$z = \texttt{false}, w = \texttt{true}$



How about CNF?

We consider a very special case: monotone 2-CNF

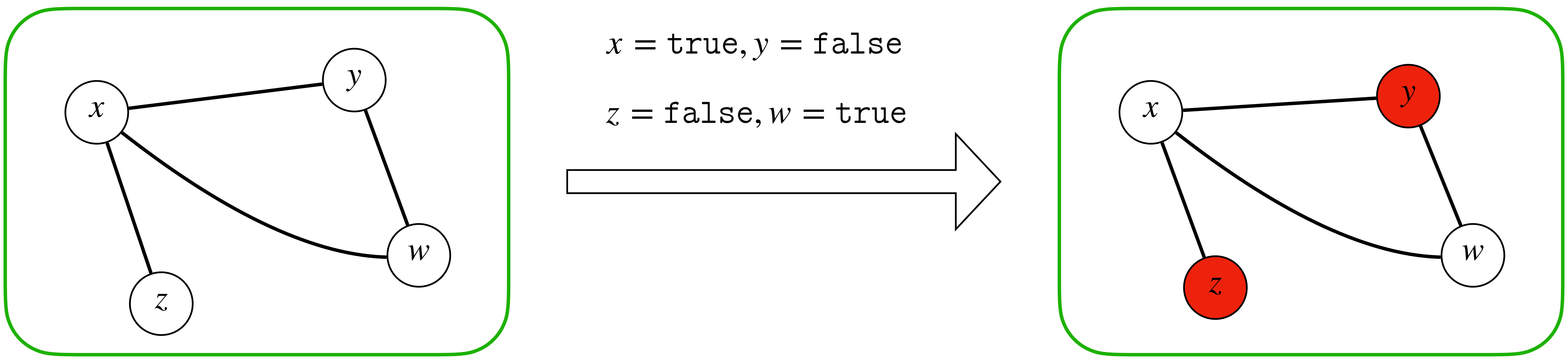
$$\varphi = (x \lor y) \land (x \lor z) \land (x \lor w) \land (y \lor w)$$



How about CNF?

We consider a very special case: monotone 2-CNF

$$\varphi = (x \vee y) \wedge (x \vee z) \wedge (x \vee w) \wedge (y \vee w)$$



$$\#\varphi = \#\text{ of independent sets}$$

Sampling seems to be harder than DNF case…

Sampling seems to be harder than DNF case…

Rejection sampling is correct but inefficient

Sampling seems to be harder than DNF case…

Rejection sampling is correct but inefficient

A natural idea is to resample those violated edges…

Sampling seems to be harder than DNF case…

Rejection sampling is correct but inefficient

A natural idea is to resample those violated edges…

Unfortunately, this is not correct.

Sampling seems to be harder than DNF case…

Rejection sampling is correct but inefficient

A natural idea is to resample those violated edges…

Unfortunately, this is not correct.

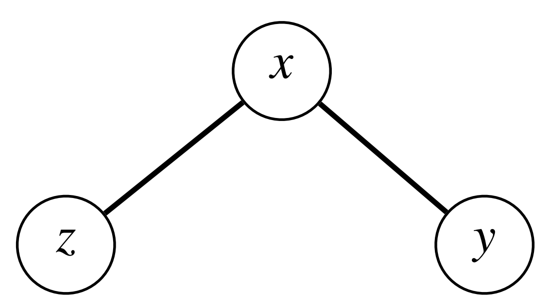
Think about

Sampling seems to be harder than DNF case…

Rejection sampling is correct but inefficient

A natural idea is to resample those violated edges…

Unfortunately, this is not correct.

Think about 

Partial Rejection Sampling

Partial Rejection Sampling

Guo, Jerrum and Liu (JACM, 2019) proposed the following fix:

Partial Rejection Sampling

Guo, Jerrum and Liu (JACM, 2019) proposed the following fix:

"Resample violated vertices and their neighbors"

Partial Rejection Sampling

Guo, Jerrum and Liu (JACM, 2019) proposed the following fix:

"Resample violated vertices and their neighbors"

We will prove the correctness and analyze its efficiency next week

From Sampling to Counting

We will show that, in many cases, if one can sample from a space, then he can also estimate the size of the space

We will show that, in many cases, if one can sample from a space, then he can also estimate the size of the space

Consider independent sets again

We will show that, in many cases, if one can sample from a space, then he can also estimate the size of the space

Consider independent sets again

$$G = (V, E), E = \{e_1, e_2, \ldots, e_m\}$$

We will show that, in many cases, if one can sample from a space, then he can also estimate the size of the space

Consider independent sets again

$$G = (V, E), E = \{e_1, e_2, \ldots, e_m\}$$

We want to estimate $I(G)$, the number of i.s. in G

Define $G_0 = G, G_i = G_{i-1} - e_i$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot |I(G_m)|$$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot \boxed{|I(G_m)|}$$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot \boxed{|I(G_m)|}$$

||

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot \boxed{|I(G_m)|}$$

$$||$$

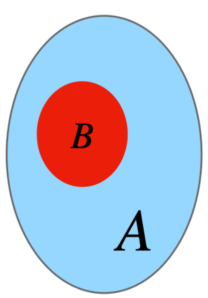
$$2^n$$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot \boxed{|I(G_m)|}$$

$$\|$$

$$2^n$$

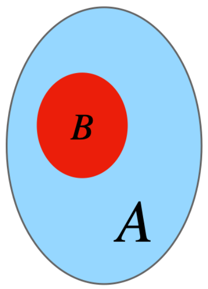


The number of samples is proportional to $\frac{|A|}{|B|}$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot |I(G_m)|$$

$$|I(G_m)| = 2^n$$



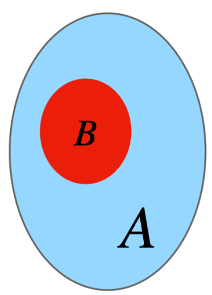
The number of samples is proportional to $\frac{|A|}{|B|}$

$A = I(G_i)$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot |I(G_m)|$$

$$|I(G_m)| = 2^n$$



The number of samples is proportional to $\frac{|A|}{|B|}$

$A = I(G_i)$

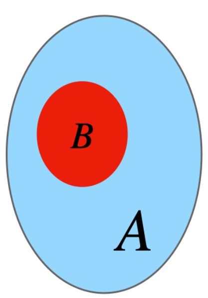
$B = I(G_{i+1})$

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot |I(G_m)|$$

$||$

2^n



The number of samples is proportional to $\frac{|A|}{|B|}$

$A = I(G_i)$

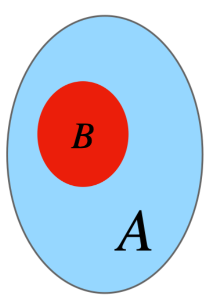
$B = I(G_{i+1})$

$\frac{|A|}{|B|}$ can't be too large!

Define $G_0 = G, G_i = G_{i-1} - e_i$

$$|I(G)| = |I(G_0)| = \frac{|I(G_0)|}{|I(G_1)|} \cdot \frac{|I(G_1)|}{|I(G_2)|} \cdots \frac{|I(G_{m-1})|}{|I(G_m)|} \cdot |I(G_m)|$$

$$|I(G_m)| = 2^n$$



The number of samples is proportional to $\frac{|A|}{|B|}$

$$A = I(G_i)$$

$$B = I(G_{i+1})$$

$\frac{|A|}{|B|}$ can't be too large!

$$\frac{I(G_i)}{I(G_{i+1})} \leq 2$$

From Counting to Sampling

On the other hand, one can consecutively sample each vertex as long as $\Pr[v \in I]$ is known

On the other hand, one can consecutively sample each vertex as long as $\Pr[v \in I]$ is known

The value can be obtained via a counting oracle

On the other hand, one can consecutively sample each vertex as long as $\Pr[v \in I]$ is known

The value can be obtained via a counting oracle

The above two reductions require the system to satisfy "self-reducible" property