Uniform Generation & Approximate Counting

Given graph G:

Tasks:
- output a perfect matching
- output a uniformly chosen perfect matching
  - e.g. Let $M_G = \{ M \mid M \text{ is a perfect matching in } G \}$
  - output $y \in_u M_G$
- count number of perfect matchings

- output a spanning tree
- output a uniformly chosen spanning tree
  - e.g. Let $S_G = \{ S \mid S \text{ is a spanning tree of } G \}$
  - output $y \in_u S_G$
- Count # of spanning trees.

uniform generation problem

Counting problem

Given Boolean formula ϕ (DNF, CNF?)

Tasks:
- output sat assignment
- output uniformly chosen sat assignment
- count # of sat assignments

Complexities:

#P.M.: Counting # perfect Matchings
#SAT: Counting # SAT assignments to CNF
#DNF: Counting # SAT assignments to DNF

} #P-complete: note if can count #SAT can solve SAT so #SAT is at least as hard.

What about #DNF?
- transform ϕ in CNF to $\bar{\phi}$ in DNF

$\#\phi = 2^n - \#\bar{\phi}$

so #DNF is also #P-complete

#SpanTree: Counting # Spanning trees } Poly time

Uniform generation?

	Decision problem	Counting problem	approx counting problem
CNF	NP-complete	#P-complete	hard
DNF	poly time	#P-complete	poly time (last lecture + this lecture)
Matching	poly time	#P-complete	in general ? poly time for {dense graphs, bipartite}
Spanning graphs	poly time	poly time	polytime
your favorite problem	?	?	?

Approximate Counting:

"Fully polynomial Randomized Approximation Scheme" (fpras)

Given φ

s.t. $Z \equiv$ # sat assignments to φ

Output y st.

$$\frac{Z}{(1+\varepsilon)} \le y \le Z \cdot (1+\varepsilon)$$

with prob $\ge 3/4$

Hope runtime poly in $|\varphi|, \frac{1}{\varepsilon}$

Note problem 1 on H.W. 1: such an algorithm can be used to give alg with success prob $\ge 1-\delta$ in time $\text{poly}(|\varphi|, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$

Question: can we get fpras for #SAT?

Answer: no if $BPP \neq NP$

fpras for #SAT $\Rightarrow$ poly time randomized alg for SAT

if φ SAT then $\#\varphi \geq 1 \Rightarrow y > \frac{1}{1+\varepsilon}$ (output)

if φ UNSAT then $\#\varphi = 0 \Rightarrow y = 0$

next Question: can we get fpras for #DNF?

Answer: Yes !!

will use

(1) uniform generation of DNF assignments

(2) "Downward self-reducibility" of DNF:

Downward Self-reducibility: (dsr)

Can compute problem soln by solving problem on smaller subproblems + putting together answers via poly time computation.

why is #-DNF d.s.r.?

$$\#\phi(x_1, x_2, \dots, x_n) = \#\phi(\underbrace{x_1 = T, x_2, \dots, x_n}) + \#\phi(\underbrace{x_1 = F, x_2, \dots, x_n})$$

both are still DNFs but in $n-1$ vars

e.g. $\#(x_1\bar{x}_2 \vee x_1 x_2 x_3 \vee \bar{x}_2\bar{x}_3) = \#(\bar{x}_2 \vee x_2 x_3 \vee \bar{x}_2\bar{x}_3)$

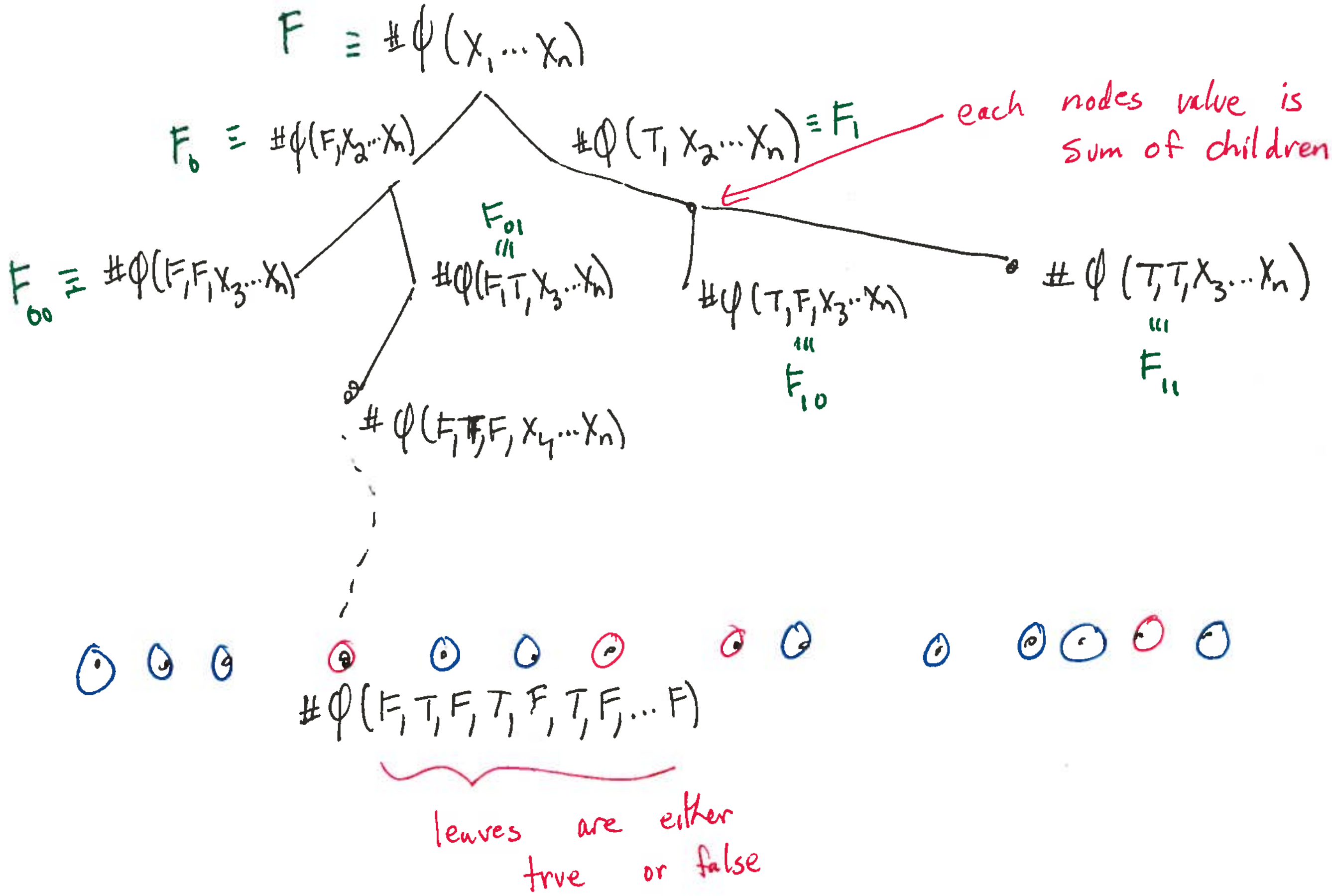
$+ \#(\bar{x}_2\bar{x}_3)$

count #settings of x_2, x_3 that satisfy (not x_1, x_2, x_3)

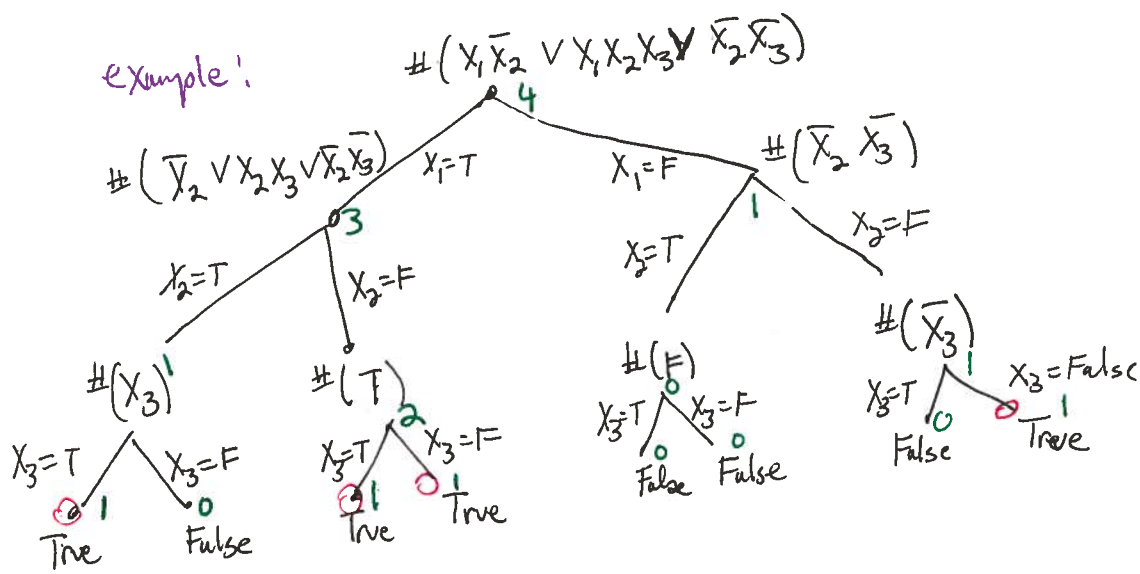
both are DNFs in $n-1$ vars

Downward self-reducibility tree:

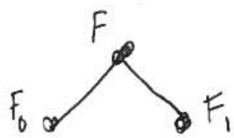
Let $F_{b_1 b_2 \cdots b_i} \equiv \#\phi(x_1=b_1, x_2=b_2, \ldots, x_i=b_i, x_{i+1} \cdots x_n)$



example:



Approximate Counting algorithm for #DNF:



Let $S_1 = F_1 / F$

$\Downarrow$

$F = \frac{F_1}{S_1}$

fraction of sat assignments s.t. $x_1 = T$

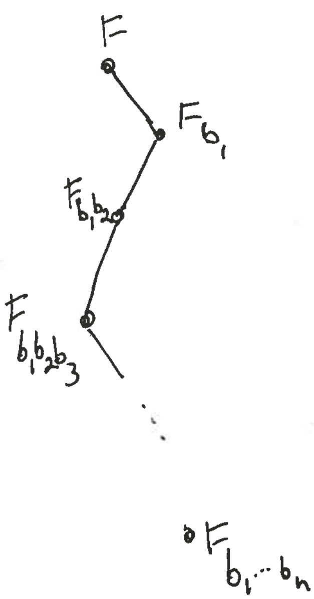
Can estimate via sampling:

- uniformly generate K SAT assignments to f
- $\tilde{S}_1 \leftarrow \frac{\#\text{ generated assignments in which } x_1 = T}{K}$

to compute F:

- estimate S_1 via sampling uniformly
- recursively compute F_1

Can do this from $S_0 + F_0$ as well



Continuing: if $S_{b_1 \cdots b_i} = \frac{F_{b_1 \cdots b_i}}{F_{b_1 \cdots b_{i-1}}}$

then $F_{b_1 \cdots b_{i-1}} = \frac{F_{b_1 \cdots b_i}}{S_{b_1 \cdots b_i}}$ ← recurse ← estimate

So $F = \frac{F_{b_1}}{S_{b_1}}$

$$= \frac{F_{b_1 b_2}}{S_{b_1} \cdot S_{b_1 b_2}} = \frac{F_{b_1 b_2 b_3}}{S_{b_1} \cdot S_{b_1 b_2} \cdot S_{b_1 b_2 b_3}} = \cdots = \frac{1}{\prod_{i=1}^{n} S_{b_1 \cdots b_i}}$$

Problems: (1) what if $S_{b_1 \cdots b_i} = 0$?

(2) we only approximate $S_{b_1 \cdots b_i}$

Problem 1:

What if $S_{b_1 \cdots b_i} = 0$?

idea: go down to "larger" child
(since sampling, might guess wrong when picking larger child but this only happens when both children have lots of sat assignments)

Claim if always pick b_i s.t. $F_{b_1 \cdots b_i} > F_{b_1 \cdots \bar{b}_i}$
then reach a satisfying assignment leaf.

Problem 2:

only get additive estimates

idea estimate each $S_{b_1 \cdots b_i}$ to within $(1+\frac{\varepsilon}{2n})$ $\boxed{\text{multiplicative}}$ factor (via Chernoff bnds need only $\text{poly}(\frac{2n}{\varepsilon}, \log 4n)$ to achieve this with prob of error $< \frac{1}{4n}$)

can union bnd over all calls to estimate S_i & argue that none fail with prob $> 1 - \frac{1}{4n} \cdot n = 3/4$

but we only get additive estimates?
Since pick "larger" child, additive estimate $\Rightarrow$ multiplicative estimate! if $r \geq 1/2$

$r + \frac{\varepsilon}{4n} \leq r\left(1 + \frac{\varepsilon}{4n \cdot r}\right) \leq r\left(1 + \frac{\varepsilon}{2n}\right)$

Claim

$$\text{output} \leq \frac{F_{b_1}}{\hat{S}_{b_1}} \leq \frac{F_{b_1 b_2}}{\hat{S}_{b_1} \hat{S}_{b_1 b_2}} \leq \cdots \leq \frac{1}{\prod_{i=1}^{n} \hat{S}_{b_1 \cdots b_i}}$$

$$\leq \frac{\left(1+\frac{\varepsilon}{2n}\right)^n}{\prod S_{b_1 \cdots b_i}} = F \cdot \underbrace{\left(1 + \frac{\varepsilon}{2n}\right)^n}_{\leq (1+\varepsilon)} \leq F(1+\varepsilon)$$

Similarly, output $\geq \frac{F}{(1+\varepsilon)}$

$\blacksquare$

Algorithm to estimate #DNF:

- estimate s_0, s_1 using uniformly generated sat assignments
- let $b_1 \Leftarrow \arg\max \{s_0, s_1\}$
- Recurse on F_{b_1}

Runtime?

$n \cdot$ # samples required to get $\frac{\varepsilon}{4n}$ additive error $\cdot$ runtime of uniform generator

↑ poly in $\left(\frac{\varepsilon}{4n}\right)^{-1}$ via Chernoff bnds

↑ poly in n

$= \text{poly}(\frac{1}{\varepsilon}, n)$

Prob [algorithm works] = Pr [estimate falls within $\frac{\varepsilon}{4n}$ additive error at each of n calls]

$\geq 1 - n \cdot \underbrace{\Pr[\text{estimate bad in single call}]}_{\text{Chernoff bnds}}$

This works for any dsr problem

polytime (almost)-uniform generation of solns to #? $\Rightarrow$ polytime approximate counting of solns to #?

("?" needs to be dsr)

What about $\Leftarrow$?

Uniform Generation + Approx Uniform generation

Computational problem of SAT, graph matching, ...

e.g.

def. Uniform generator for solns of problem π

on input x,

e.g. assignment   e.g. Boolean formula

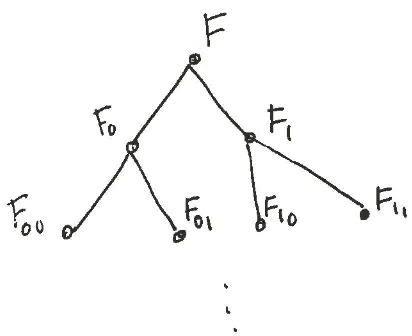
- Define set of solns $S_x \equiv \{ z \mid z \text{ is a soln to } x \}$
- outputs y uniformly from S_x
  - $\forall y \in S_x$ $\Pr[\text{output } y] = \frac{1}{|S_x|}$
  - do not output $y \notin S_x$
- runs in time $\text{poly}(|x|)$

def almost uniform generator as above, but has extra input ε

- $\forall y \in S_x$ $\frac{1}{|S_x|} \cdot \frac{1}{1+\varepsilon} \leq \Pr[\text{output } y] \leq \frac{1}{|S_x|} \cdot (1+\varepsilon)$
- runs in time $\text{poly}(|x|, \frac{1}{\varepsilon})$

Approximate Uniform generation from approximate counting algorithms:

Assume (perfect) counting alg for #DNF



Recursive Algorithm: At $b_1 \dots b_i$

use (perfect) counter to compute

$r_0 = F_{b_1 \dots b_i 0}$ + $r_1 = F_{b_1 \dots b_i 1}$

Go down left branch with prob $\frac{r_0}{r_0 + r_1}$

+ right branch o.w.

Claim (1) always reach a sat assignment

(2) $\Pr[\text{output assignment } b = (b_1 \dots b_n)] = \frac{F_{b_1}}{F} \cdot \frac{F_{b_1 b_2}}{F_{b_1}} \cdot \frac{F_{b_1 b_2 b_3}}{F_{b_1 b_2}} \cdots \frac{1}{F_{b_1 \dots b_{n-1}}}$

$= \frac{1}{F}$

which is same for each sat assignment.

$\Rightarrow$ Uniformly generate a sat assignment

Question What if only approx counter?

Answer RHS $\leq \frac{1}{F} \cdot \left(\frac{1+\varepsilon'}{1-\varepsilon'}\right)^n \leq \frac{1}{F} \cdot \frac{1}{(1-\varepsilon)}$ when choose $\varepsilon' < \frac{\varepsilon}{2n}$

$\Rightarrow$ close to uniform generation of sat assignment

This also works for any DSR problem!

Thm [Jerrum Valiant Vazirani] for any problem in NP that is DSR (almost all we know are!)

Approx counting #solns doable in Poly time iff Almost uniform generation doable in Poly time