October 14, 2020 10:54 AM

Conditional Lower Bds Based on SAT

(CNF-) SAT Problem Given CNF formula F with n vars, decide if ∃ satisfying assignment

e.g. $\underbrace{(x_1 \vee x_2 \vee \overset{\text{literal}}{\bar{x}_3})}_{\text{clause}} \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge \cdots$

k-SAT: version for max clause length k

Exponential-Time Hypothesis (ETH)

No alg'm that solves k-SAT for all const k in $2^{\underset{\text{little-oh}}{o(n)}}$ time

Note: there are k-SAT alg'ms that beat 2^n

e.g., take a clause $\alpha_1 \vee \dots \vee \alpha_k$
try all $\underset{2^k-1}{\wedge}$ settings of $\alpha_1, \dots, \alpha_k$

$$T(n) \le (2^k-1)\, T(n-k)$$

$$\Rightarrow O^*\left((2^k-1)^{n/k}\right)$$

$$= O^*\left(\left[(2^k-1)^{1/k}\right]^n\right)$$

$$= O^*\left(\underline{\left(2-\Theta\left(\tfrac{1}{k2^k}\right)\right)}^n\right).$$

base converges to 2 as $k \to \infty$.

3SAT: $(7^{1/3})^n$ $< 1.92^n$

Paturi, Pudlak, Zane '97 $\left(2-\Theta\left(\frac{1}{k}\right)\right)^n$

Schöning '99 $\left(2-\frac{2}{k}\right)^n$

Strong Exp. Time Hypothesis (SETH)

No algm that solves k-SAT for all const k
in $O(2^{(1-\delta)n})$ time
for some fixed $\delta > 0$ indep of k

$(2-\delta')^n$

Sparsification Lemma (Impagliazzo, Paturi, Zane '98)

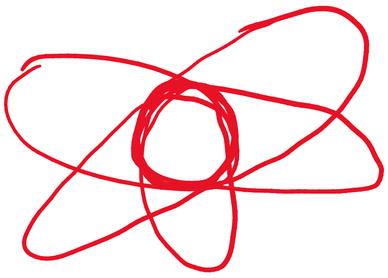
k-SAT could be solved in $O(2^{(1-\delta)n})$ time
for some $\delta > 0$ indep of k

$\Longleftarrow$ "f-sparse k-SAT" for any const f
could be solved in $O(2^{(1-\delta')n})$ time
for some $\delta' > 0$ indep of k & f.

every var
appears in
$\le f$ clauses

Pf: Omitted. □

weak sunflower
branching



Reduction: k-SAT → Subset Sum

Thm (Abboud, Bringmann, Hermelin, Shabtay '19)

Assuming SETH, no algm for subset sum
in $O(t^{1-\delta} \cdot 2^{o(n)})$ time.

(n positive ints, target t)

Pf:

Warm-up idea - variant of textbook NP-completeness pf
for subset sum.

Suppose subset-sum could be solved
in $T(n,t) \le O(t^{1-\delta} 2^{o(n)})$ time.

in $T(n,t) \le O(t \cdot 2 \quad)$ time.

Given f-sparse k-CNF formula F (f, k consts)
with vars $x_1, \dots, x_n$ & clauses $C_1, \dots, C_m$
($m \le 2fn$)

Ex $F = \underbrace{(x_1 \vee \overline{x_2})}_{C_1} \wedge \underbrace{(\overline{x_1} \vee x_3)}_{C_2}$ $\quad f=2$

satisfying assignment
$x_1=1,\ x_2=0,\ x_3=1$

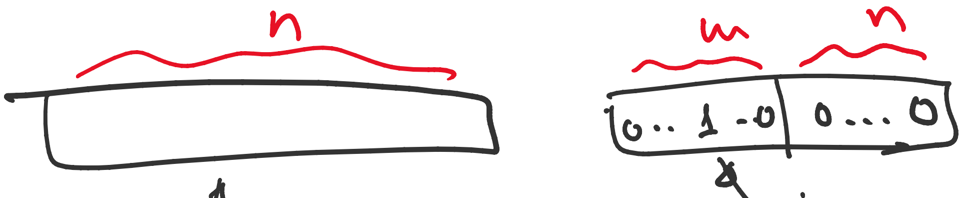
create numbers

	x_1	x_2	x_3	C_1	C_2	x_1	x_2	x_3
C_1	0	0	0	1	0	0	0	0
	1	0	0	1	0	0	0	0
	1	1	0	1	0	0	0	0
C_2	0	0	0	0	1	0	0	0
	0	0	1	0	1	0	0	0
	1	0	1	0	1	0	0	0
x_1	2	0	0	0	0	1	0	0
	0	0	0	0	0	1	0	0
x_2	0	2	0	0	0	0	1	0
	0	1	0	0	0	0	1	0
x_3	0	0	2	0	0	0	0	1
	0	0	1	0	0	0	0	1
t	2	2	2	1	1	1	1	1

1. For each clause C_j & each possible assignment φ of its $\le k$ vars that satisfies C_j,

Create number

$z(C_j, \varphi) =$



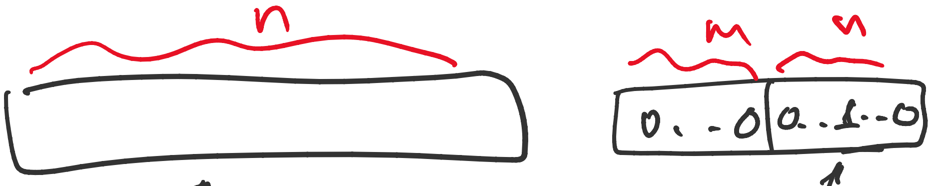
ith digit is $\begin{cases} 1 & \text{if } x_i=1 \text{ in } \varphi \\ 0 & \text{if } x_i=0 \text{ in } \varphi \\ 0 & \text{if } x_i \text{ not in } C_j \end{cases}$

jth bit is 1
0 else

2. For each var x_i & $\alpha \in \{0,1\}$,

Create number

$z(x_i, \alpha) =$

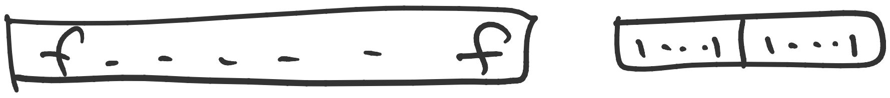


ith digit is $\begin{cases} f \text{ if } \alpha = 0 \\ f - \underline{\text{freq}(x_i)} \text{ if } \alpha = 1 \end{cases}$

all other 0.

ith bit is 1
0 else

3. Set $t =$



Use base $\max(f+1, 2^k)$

unfortunately, numbers too big

$t = 2^{O(m+n)} = 2^{\underline{O}(n)} >> 2^n.$

next idea - reduce # vars & # constraints

divide into $\frac{n}{B}$ groups of B vars → "super-vars"

& $\frac{m}{B}$ groups of B clauses → "super-constraints"

each super-var lies in $\underline{[2^B]}$ ←

each super-constraint contains $\leq kB$ supervars

each supervar appears $\leq fB$ super-constraints

Lemma (from additive combinatorics, Behrend '46)

For any $N, M, \varepsilon > 0$,

$\exists$ set of N numbers $\ell[1], \ldots, \ell[N]$ in $\underline{[M^{O(1/\varepsilon)} N^{1+\varepsilon}]}$ ← naively, $\boxed{M^N}$

which is M-average-free

i.e. $\frac{\ell(i_1) + \ldots + \ell(i_{M'})}{M'} = \ell[i], \quad M' \leq M$

$\implies \ell[i_1] = \ldots = \ell[i_{M'}] = \ell[i]$

M

$$\Rightarrow \quad \ell[i_1] = \ldots = \ell[i_M] = \ell[i].$$

TO BE CONT'D . . .