Satisfiability of Boolean Formulas

Definition (CNF-SAT) / k-SAT:

Input: Boolean formula φ on N variables in conjunctive normal form (CNF):

$$\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_M \qquad C_i = l_{i1} \vee \dots \vee l_{ik_i}$$

clauses; literals

M: # clauses

where $l_{ij} = x$ or $l_{ij} = \bar{x}$ for one of the variables x

$k_i = k$ for all i "clause width"

Task: Decide if there is an assignment of true/false to the variables in φ s.t. φ evaluates to true

Example: $(a \vee b \vee \bar{c}) \wedge (\bar{a} \vee \bar{b} \vee d) \wedge (\bar{a} \vee c \vee \bar{d})$

Fastest Known Algorithms:

- 2-SAT: $O(N+M)$
- 3-SAT: $1.31^N \cdot poly(M)$ [Paturi et al. '98]
- 4-SAT: $1.47^N \cdot poly(M)$ "
- $\vdots$
- k-SAT: $2^{N(1-c/k)} poly(M)$ for some constant c "
- CNF-SAT: $2^{(1-O(1/\log(M/N)))\cdot N} \cdot poly(M)$ [Calabro et al. '06]

Conjecture [Impagliazzo/Paturi]: Exponential Time Hypothesis (ETH)
For every $k \geq 3$, there is a constant c_k s.t. there is no $2^{c_k \cdot N} \cdot poly(M)$-time algorithm for k-SAT
(implies: No $2^{o(N)}$ poly(M)-time algorithm for 3-SAT)

Conjecture [I/P]: Strong Exponential Time Hypothesis (SETH)
For every $\delta > 0$, there is a k s.t. ~~the time complexity of~~ there is no $2^{(1-\delta)N} \cdot poly(M)$-time algorithm for k-SAT
(implies: there is no $2^{(1-\delta)N}$ poly(M)-time algorithm for CNF-SAT with constant $\delta > 0$)

No poly(N)-time algorithm for 3-SAT

Observation: SETH $\Rightarrow$ ETH $\Rightarrow$ P $\neq$ NP

From k-SAT to Hitting Set

Definition: Hitting Set Problem

Given: Family $\mathcal{F} = \{S_1, \dots, S_m\}$ of sets over universe $\mathcal{U} = \{u_1, \dots, u_n\}$, Integer t $(S_i \subseteq \mathcal{U})$

Task: Decide if there is a subset $T \subseteq \mathcal{U}$ of size $|T| \leq k$ s.t. $\forall S_i \in \mathcal{F}: S_i \cap X \neq \emptyset$ (i.e. every S_i is "hit" by X)

Naive Algorithm: $O(2^n mn)$ time

Try all possible subset of $\mathcal{U}$ as candidates for T

(checking whether candidate T' hits all S_i takes $O(mn)$ time)

Theorem: Assuming SETH, there is no $\sqrt{2}^{(1-\delta)n}$ poly(n,m)-time algorithm⊛ for the Hitting Set Problem ⊛(with constant $\delta > 0$)

Proof: Let $\mathcal{U} = \{x_1, \dots, x_N, \bar{x}_1, \dots, x_N\}$ (under)

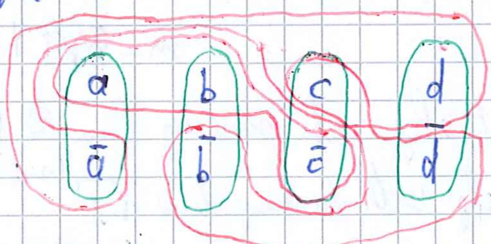
$\mathcal{F} = \{\{x_i, \bar{x}_i\} : 1 \leq i \leq N\} \cup$

$\{\{l_{j1}, \dots, l_{jk}\} \mid 1 \leq j \leq M\}$ (set of literals for each clause)

$t = N$

~~Example φ=~~

Example: $\varphi = (a \vee b \vee \bar{c}) \wedge (\bar{a} \vee c \vee d) \wedge (\bar{b} \vee c \vee \bar{d})$



Claim: φ satisfiable $\Leftrightarrow$ $\exists$ hitting set T of size $|T| \leq t$

"$\Rightarrow$" Consider satisfying assignment

For every variable x: add x to T if x is set to true

add $\bar{x}$ to T o.w.

Clearly: $|T| = N = t$

Every set $\{x_i, \bar{x}_i\}$ is hit by T by construction

Also every set $\{l_{j1}, \dots, l_{jk}\}$ is hit because some l_{ji} must evaluate to 1

"$\Leftarrow$" Hitting set T must contain either x_i or $\overline{x_i}$ for every variable x_i (because of sets $\{x_i, \overline{x_i}\}$ in $\mathcal{F}$)

~~As~~ $\cancel{|T| \leq t = N}$ This ~~impl~~ implies $|T| \geq N = t$

But we also know $|T| \leq t$

$\Rightarrow |T| = t = N$

Thus, T contains exactly one of x_i or $\overline{x_i}$ for every variable x_i

We can thus define an evalution $f : x_i \to \{\text{true}, \text{false}\}$

$f(x_i) \Rightarrow$ true if T contains x_i

$f(x_i) \Rightarrow$ false if T contains $\overline{x_i}$

Furthermore, T contains at least one l_{ji} out of

$l_{j1} \vee \dots \vee l_{jk}$ for every clause C_j

By its argued above, as l_{ji} contained in T, its complement $\overline{l_{ji}}$ is not contained in T. Thus $f(l_{ji}) =$ true and thus C_j is satisfied with eval. f

$\Rightarrow \varphi$ satisfied by evaluation f.

Thus: φ sat. $\Leftrightarrow \exists$ h. s. of size $\leq t$

Time complexity: hitting set instance has $n = |U| = 2N$

$m = |\mathcal{F}| = N + M$

If there were a hitting set alg with running time $\sqrt{2}^{(1-\delta)n} \text{poly}(n, m)$

we would get k-SAT alg with running time $\sqrt{2}^{(1-\delta)2N} \text{poly}(N, M)$

$$= \left(\sqrt{2}^{2}\right)^{(1-\delta)N} \text{poly}(N, M)$$

$$= 2^{(1-\delta)N} \text{poly}(N, M)$$

This contradicts SETH $\square$

<u>Remark</u>: One can even show that there is no $2^{(1-\delta)n} \text{poly}(n, m)$ time alg for hitting set assuming SETH

(More complicated reduction, needs some math)

From SAT to orthogonal vectors

Goal: Reduction from CNF-SAT instance φ to OV instance (A, B) s.t.
φ satisfiable iff (A, B) has orthogonal pair

Variables of φ: $x_1, \dots, x_N$ Clauses of φ: $C_1, \dots, C_M$

Consider all possible partial truth value assignments to $x_1, \dots, x_{N/2}$

$2^{N/2}$ such assignments, Let $U = \{true, false\}^{N/2}$ represent these assignments

Similar, $V = \{true, false\}^{N/2}$ represents truth ~~vals~~ value assignments to $x_{N/2+1}, \dots, x_N$

Partial assignment $u \in U$ <u>satisfies</u> clause C if
$\exists i$ s.t. x_i set to true in u and x_i appears unnegated in C
or $\exists i$ s.t. " false " " negated " "

We write $sat(u, C) = 1$ if u sat. C and $sat(u, C) = 0$ o.w.
$unsat(u, C) = 1 - sat(u, C)$
Similar def. for $v \in V$ and clause C
Now define OV instance as follows:

$A = \{ (unsat(u, C_1), \dots, unsat(u, C_M)) \mid u \in U \}$
$B = \{ (unsat(v, C_1), \dots, unsat(v, C_M)) \mid v \in V \}$ M-dim. vectors

<u>Claim:</u> φ satisfiable iff (A, B) has orthogonal pair

"$\Rightarrow$" ~~For every clause C_i:~~ Let w^* be a truth value assignment s.t. φ evaluates to "true". $w^* = (\underbrace{x_1, \dots, x_{N/2}}_{u^*}, \underbrace{x_{N/2+1}, \dots, x_N}_{v^*})$

Let u^*, v^* be partial assignments of w^* corresponding to $x_1, \dots, x_{N/2}$ and $x_{N/2+1}, \dots, x_N$ resp.

Then every clause C_i has $sat(u^*, C_i) = 1$ or $sat(v^*, C_i = 1) = 1$
$\Leftrightarrow unsat(u^*, C_i) = 0$ or $unsat(v^*, C_i = 0) = 0$

$\Rightarrow$ ~~Vectors~~ Vectors in A, B (corresponding to u^*, v^*) that are orthogonal to each other

"$\Leftarrow$" ✓

~~"#" Running~~ Time Complexity:

size of OV instance: $|A| = |B| = 2^{N/2} =: n$

vectors of dimension $M =: d$

Running time for ~~creating~~ creating OV instance: $O(2^{N/2} \cdot M) = O(nd)$

Theorem: Assuming SETH there is no $O(n^{2-\delta}\, poly(d))$ algorithm (for $\delta > 0$) for the Orthogonal Vectors Problem

Proof: Reduction would imply a CNF-SAT algorithm with running time

$$O(nd + n^{2-\delta}\, poly(d)) = O(2^{N/2} \cdot M + (2^{N/2})^{2-\delta} \cdot poly(M))$$
$$= O(2^{(1-\delta/2)N} \cdot poly(M))$$

violates SETH!

Corollary: Assuming SETH, there is no $O(|s||G|^{1-\delta})$-time algorithm (for a constant $\delta > 0$) for the NFA acceptance problem

We will see more results of this kind

$$\underbrace{\text{SETH}}_{\text{"exponential regime"}} \longrightarrow \underbrace{\text{OVH} \to \text{NFA acceptance},\ \vdots}_{\text{"polynomial regime"}}$$

General Scheme:

Problem P		Problem Q
instance I	$\xrightarrow[\text{time } r(n)]{\text{Reducing}}$	instance J
size n		size $s(n)$

I is a 'yes'-instance $\iff$ J is a 'yes'-instance

Then: $t(n)$ ~~alg~~-time algorithm for Q implies $r(n) + t(s(n))$-alg. for P

Or put in different words: If P has no $r(n) + t(s(n))$-time alg., then Q has no $t(n)$-time algorithm