

Notes On Sparse Sets- I
W.I. Gasarch

1 Defintions and Notation

Mahaney [2] proved that sparse sets cannot be NP-hard under m -reductions. Watanabe [3] showed that sparse sets cannot be NP-hard under btt -reductions (do not worry if you do not know what this means). Homer and Longpre [1] simplified the proof of Watanabe. In this writeup we present a proof of Mahaney's theorem in the spirit of Homer-Longpre.

Notation 1.1

1. Σ is a finite set.
2. Σ^* is the set of finite strings over Σ .
3. If $n \in \mathbb{N}$ then Σ^n is the set of strings over Σ that are of length n , and $\Sigma^{\leq n}$ is the set of strings of length $\leq n$ over Σ .
4. \preceq denotes the lexicographic ordering on Σ^* .
5. If A is a set then $|A|$ is the number of elements in it. If σ is a string then $|\sigma|$ is its length. Hence we are using the same notation for both things. Sorry about that.

Def 1.2 LSAT (called *Left Sat*) is the set of ordered pairs (ϕ, y) such that

1. ϕ is a Boolean formula. Let n be the number of variables.
2. $y \in \{0, 1\}^n$ is viewed as an assignment.
3. There exists $x \preceq y$ such that $\phi(x)$.

Exercise 1

1. Prove that LSAT is in NP.

2. Prove that LSAT is NP-complete.

Exercise 2 Let ϕ be a Boolean formula on n variables. Let $m \in \mathbf{N}$. Let $y \prec y'$ and $y_1 \prec \dots \prec y_m \in \{0, 1\}^n$.

1. Prove that $(\phi, y) \in \text{LSAT}$ then $(\phi, y') \in \text{LSAT}$.
2. Prove that one of the following occurs
 - (a) For all i , $(\phi, y_i) \in \text{LSAT}$.
 - (b) For all i , $(\phi, y_i) \notin \text{LSAT}$.
 - (c) There exists i , $1 \leq i \leq m$ such that $(\phi, y_1), \dots, (\phi, y_i) \notin \text{LSAT}$ and $(\phi, y_{i+1}), \dots, (\phi, y_m) \in \text{LSAT}$

Exercise 3 Show that if $A \leq_m^p B$ and $B \leq_m^p C$ then $A \leq_m^p C$.

Exercise 4 Show that if $0 < \delta < \frac{1}{10}$ then $1 - \delta < e^{-\delta} < 2^{-\delta}$.

Def 1.3 A set $S \subseteq \{0, 1\}^n$ is *sparse* if there exists a polynomial s such that, for all n , $|S \cap \{0, 1\}^{\leq n}| \leq s(n)$. (So the notion of sparse only applies to sets of strings over $\{0, 1\}$.)

Theorem 1.4 *If there exists a sparse set S such that $\text{SAT} \leq_m^p S$ then $\text{P} = \text{NP}$.*

Proof:

By Exercise 1.1 $\text{LSAT} \in \text{NP}$. Since SAT is NP-complete, $\text{LSAT} \leq_m^p \text{SAT}$. By the premise $\text{SAT} \leq_m^p S$. By Exercise 3 we have $\text{LSAT} \leq_m^p S$. Let f be the reduction, that is, $f \in P$ and

$$(\phi, y) \in \text{LSAT} \text{ iff } f(\phi, y) \in S.$$

Note that f returns strings so the notation $|f(\phi, y)|$ makes sense. Since f runs in poly time its output length is bounded by a poly in its input length. Let $p(n)$ be a polynomial such that

if ϕ has n variables and $y \in \{0, 1\}^n$ then $|f(\phi, y)| \leq p(n)$.

S is sparse, so there is a poly s such that $(\forall n)[|S \cap \{0, 1\}^{\leq n}| \leq s(n)]$.

Hence

$$|\{0, 1\}^{\leq p(n)} \cap S| \leq s(p(n)).$$

We can assume that, for all n , $s(n) \geq 10$. (We can do this since $s(n)$ is used as an upper bound.)

Before giving the algorithm for SAT we discuss the intuition. Initially when you are given ϕ you are looking at $[0^n, 1^n]$ for a satisfying assignment. Our algorithm will use the reduction f to eliminate large parts of the interval. We will actually present an algorithm that does the following:

Input: a formula ϕ on n variables and a set $POSS \subseteq \{0, 1\}^n$ such that, if $\phi \in SAT$, then it has a satisfying assignment in $POSS$.

Output: Either (1) a set $POSS'$ such that, if $\phi \in SAT$, then it has a satisfying assignment in $POSS'$, and $|POSS'| \leq |POSS| \cdot 2^{-\frac{1}{s(p(n))+1}}$, or (2) YES $\phi \in SAT$, or (3) NO $\phi \notin SAT$.

We will first exhibit an algorithm A for this problem We will then show how we can easily use algorithm A to solve SAT in P.

Algorithm A

1. Input $(\phi, POSS)$. Let $POSS = \{y_1 \prec \dots \prec y_{m'}\}$. (We use m' here since we will actually work with a small variant m in the algorithm. Do not worry about this now.)
2. If $|POSS| \leq s(p(n))$ then, for each $y \in POSS$, evaluate $\phi(y)$. If one of those y 's is a satisfying assignment then output YES. If none of the y 's is a satisfying assignment then output NO.
3. (If you got to this step then $|POSS| \geq s(p(n)) + 1$.)
 - (a) *Case 1:* If $m' \equiv 0 \pmod{s(p(n)) + 1}$ then let $m = m'$ and goto step 4.
 - (b) *Case 2:* If $m' \not\equiv 0 \pmod{s(p(n)) + 1}$ then let m be the largest number such that $m < m'$ and $m \equiv 0 \pmod{s(p(n)) + 1}$. For all i , $m + 1 \leq i \leq m'$, evaluate $\phi(y_i)$. (Note that there are a polynomial number of y_i 's to try.) If any of these y 's is a satisfying assignment then output YES. If none of these y 's is a satisfying assignment then goto the step 4.

Note that in either case we start the step 4 with the number of y 's to look at being $m \equiv 0 \pmod{s(p(n)) + 1}$.

4. Let $L = \frac{m}{s(p(n))+1}$ (this is an integer). For each j , $1 \leq j \leq s(p(n)) + 1$ let $z_j = f(\phi, y_{jL})$. (It may look awkward, but the subscript really is jL , that is $j \times L$.) Note that by Exercise 2.2 and the definition of reduction we have that one of the following must occur:

- (a) For all j , $z_j \in S$.
 - (b) For all j , $z_j \notin S$.
 - (c) There exists j , $1 \leq j \leq s(p(n)) + 1$ such that $z_1, \dots, z_j \notin S$ and $z_{j+1}, \dots, z_{s(p(n))+1} \in S$.
5. (a) *Case 1:* There exists $j_1 < j_2$ such that $z_{j_1} = z_{j_2}$. We know $y_{j_1L} \prec y_{j_2L}$. Let $z = z_{j_1} = z_{j_2}$. Note that we DO NOT KNOW if $z \in S$ or not. But this information is still useful. Since $f(\phi, y_{j_1L}) = f(\phi, y_{j_2L}) = z$ we know that $(\phi, y_{j_1L}) \in \text{LSAT}$ iff $(\phi, y_{j_2L}) \in \text{LSAT}$. Hence, if there is a satisfying assignment $y \preceq y_{j_2L}$ then there is a satisfying assignment $y \preceq y_{j_1L}$. Therefore we can eliminate the possibilities $y_{j_1L+1}, y_{j_1L+2}, \dots, y_{j_2L}$. We eliminate this NOT because none of these can satisfy ϕ , but because IF one of them satisfies ϕ , then some $y \preceq y_{j_1L}$ satisfies ϕ . Output the set $POSS' = POSS - \{y_{j_1L+1}, \dots, y_{j_2L}\}$. Note that

$$\begin{aligned} |POSS'| &\leq m - L = m - \frac{m}{s(p(n))+1} = m\left(1 - \frac{1}{s(p(n))+1}\right) \\ &\leq |POSS| \cdot \left(1 - \frac{1}{s(p(n))+1}\right) \end{aligned}$$

By Exercise 4 and the fact that $s(p(n)) + 1 \geq 10$ yields

$$\left| |POSS'| \cdot \left(1 - \frac{1}{s(p(n))+1}\right) \right| \leq |POSS| \cdot 2^{-\frac{1}{s(p(n))+1}}.$$

- (b) *Case 2:* $z_1, \dots, z_{s(p(n))+1}$ are all different. Since these are all of length $\leq p(n)$ we know that at most $s(p(n))$ can be in S . Hence there must be some z_j that is not in S . By the comment made in Step 4 we know that $z_1 \notin S$. Hence $(\phi, y_L) \notin \text{LSAT}$. Therefore we can eliminate y_1, \dots, y_L as possible satisfying assignments. Output the set $POSS' = POSS - \{y_1, \dots, y_L\}$. By similar reasoning to that used in Case 1 we have $|POSS'| \cdot \left(1 - \frac{1}{s(p(n))+1}\right) \leq |POSS| \cdot 2^{-\frac{1}{s(p(n))+1}}$.

We now use algorithm \mathcal{A} in a poly time algorithm for SAT.

1. Input ϕ .
2. $POSS = [0^n, 1^n]$.
3. Iterate the following procedure until an output of YES or NO occurs. (We later prove that at most a polynomial number of iterations are needed.)
 - (a) Run A on $(\phi, POSS)$.
 - (b) If output is YES then stop and output YES. If output is NO then stop and output NO. If output is $POSS'$ then let $POSS = POSS'$ and goto step a.

Let $POSS_i$ be the set $POSS$ after i iterations. Let $a_i = |POSS_i|$. It is easy to see that

$$a_0 = 2^n$$

$$a_i \leq a_{i-1} 2^{-\frac{1}{s(p(n))+1}}.$$

Hence

$$a_i \leq 2^n \cdot 2^{-\frac{i}{s(p(n))+1}} = 2^{n - \frac{i}{s(p(n))+1}}$$

It is easy to see that if $i = n(s(p(n)) + 1)$ then $a_i = 1$. Hence there exists $i_0 \leq n(s(p(n)) + 1)$ such that a_{i_0} is small enough that algorithm \mathcal{A} will output YES or NO. ■

We just showed that if S is sparse and $SAT \leq_m S$ then $P = NP$. Note that if $SAT \leq_m S$ then we can answer $\phi \in SAT$ by asking

References

- [1] S. Homer and L. Longpre. On reductions of np sets to sparse sets. *Journal of Computer and Systems Sciences*, 48, 1994. Prior version in *STRUCTURES* 1991.
- [2] S. Mahaney. Sparse complete sets for NP: Solution to a conjecture of Berman and Hartmanis. *Journal of Computer and Systems Sciences*, 25:130–143, 1982.

- [3] Ogiwara and Watanabe. On polynomial-time bounded truth-table reducibility of np sets to sparse sets. *SIAM Journal of Computing*, 20, 1991.