Sparse Sets I: Showing $SAT \leq_m S \rightarrow P = NP$ Using Intervals
Exposition by William Gasarch

1 Definitions and Notation

A sparse set is one that, for each n, does not have that many elements of length $\leq n$ (this will be defined rigorously soon). How helpful are they? The answer, in short, is 'not very helpful'. We will show this in three different ways.

1. Mahaney [3] proved that sparse sets cannot be NP-hard under m-reductions unless P = NP. Watanabe [4] showed that sparse sets cannot be NP-hard under *btt*-reductions (do not worry if you do not know what this means) unless P = NP. 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.

2. Karp and Lipton [2] show that sparse sets cannot be NP-hard under T-reductions unless $\Sigma_2^{\rm p} = \Pi_2^{\rm p}$. We present an alternative proof due to Hopcroft.

3. Yap [5] showed that if SAT $\in \Pi_1^{\rm p,SPARSE}$ then $\Sigma_3^{\rm p} = \Pi_3^{\rm p}$. Roughly speaking, SAT is not in $\Pi_1^{\rm p}$ even if the $\Pi_1^{\rm p}$ is based on a predicate that has access to a sparse set.

Why do we care about these things? There are historical reasons why Theorist care about some of them (search the literature for 'The Berman-Hartmanis Conjecture'); however, we have other motivations.
Our quest for Natural sets that are probably in $NP - P$. Consider the set
1) Ramsey's theorem states that

For all m there is and n such that for all 2-colorings of the edges of K_n there is a monochromatic K_m. It is known that $m = \theta(\log n)$; however, finding the exact Ramsey numbers seems hard.

Here is one attempt to pin down the difficulty of finding Ramsey numbers. Let

$RAM = \{(n, m) \mid$ there is a 2-coloring of the edges of K_n with no monochromatic $K_m\}$

Is this set in NP? This is unlikely. Note that the input (which are in binary) are of length $O(\log n)$. and the witness of of length $O(n^2)$. Hence the obvious witness does not work. So, how to pin down the complexity of finding the Ramsey numbers?

Consider $RAMUNARY$:

$\{(1^n, 1^m) \mid$ there is a 2-coloring of the edges of K_n with no monochromatic $K_m\}$

This set is in NP and pins down the complexity of finding the Ramsey numbers. However, this is a sparse set. Therefore it is likely not NP-complete.

2) We will show that if GI is NP-complete then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$.. We will need Yap's theorem: if Σ_2^{p} is Turing-reducible to a spare set then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$.

Another reason to care about sparse sets.

The notion of "Turing reduction to a sparse set" is equivalent to polynomial sized circuits, which are of interest.

Notation 1.1

1. If $n \in \mathsf{N}$ then $\{0,1\}^n$ is the set of strings over $\{0,1\}$ that are of length n, and $\{0,1\}^{\leq n}$ is the set of strings of length $\leq n$ over $\{0,1\}$.
2. $\preceq$ denotes the lexicographic ordering on $\{0,1\}^*$.
3. 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 the length of a string and the size of a set. Sorry about that.

Def 1.2 A set $S \subseteq \{0,1\}^*$ 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 $\{0,1\}$.)

2 If SAT $\leq_m^{\mathrm{p}} S$, S Sparse, then P = NP

Def 2.1 If A and B are sets then $A \leq_m^{\mathrm{p}} B$ means that there is a polynomial time function f such that $x \in A$ iff $f(x) \in b$. We use the $\leq_m^{\mathrm{p}}$ notation for two reasons: (1) We are allowing the function f to be many-to-1 instead of 1-to-1. I do not care about this either but it is historical. (2) To distinguish it from $\leq_T^{\mathrm{p}}$, Turing reduction, which we will use in a different set of notes.

Def 2.2 LSAT (called *Left Sat*) is the set of ordered pairs (ϕ, z) such that

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

Exercise 1

1. Prove that LSAT is in NP.
2. Prove that LSAT is NP-complete.

We will use the following fact, which we leave as exercises for the reader, in what follows.

Exercise 2 Let ϕ be a Boolean formula on n variables. Let $m \in \mathsf{N}$. Let $z \prec z'$ and $z_1 \prec \cdots \prec z_m \in \{0,1\}^n$.

1. Prove that if $(\phi, z) \in$ LSAT then $(\phi, z') \in$ LSAT.
2. Prove that one of the following occurs
   (a) For all i, $(\phi, z_i) \in$ LSAT.
   (b) For all i, $(\phi, z_i) \notin$ LSAT.
   (c) There exists i, $1 \leq i \leq m$ such that $(\phi, z_1), \ldots, (\phi, z_i) \notin$ LSAT and $(\phi, z_{i+1}), \ldots, (\phi, z_m) \in$ LSAT
   (d) From the above we can conclude that if for some i, $(\phi, z_i) \notin LSAT$ then $(\phi, z_1) \notin LSAT$.

Exercise 3 Show that if $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ and $B \leq_{\mathrm{m}}^{\mathrm{p}} C$ then $A \leq_{\mathrm{m}}^{\mathrm{p}} C$.

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

Theorem 2.3 *If there exists a sparse set S such that* $\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{p}} S$ *then* $\mathrm{P} = \mathrm{NP}$.

Proof:

By Exercise 1.1

$$\mathrm{LSAT} \in \mathrm{NP}.$$

Since SAT is NP-complete,

$$\mathrm{LSAT} \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{SAT}.$$

By the premise $\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{p}} S$. By Exercise 3 we have

$$\mathrm{LSAT} \leq_{\mathrm{m}}^{\mathrm{p}} S.$$

Let f be the poly time reduction such that

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

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

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

S is sparse, so there is a polynomial 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 the interval $[0^n, 1^n]$ 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 $POSS$. $POSS$ will be represented by a set of t intervals. Note that $POSS$ is a set of POSSIBILITIES for where the satisfying assignment may be.

Output one of the following:

1. A set $POSS'$ (which is at most $t+1$ intervals) such that, if $\phi \in$ SAT, then it has a satisfying assignment in $POSS'$, and $|POSS'| \leq |POSS| \cdot 2^{-\frac{1}{s(p(n))+1}}$.

2. YES $\phi \in$ SAT.

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 $(\phi, POSS)$. $POSS$ is represented by a set of intervals so

$$POSS = \{[b_1, e_1], \ldots, [b_t, e_t]\}.$$

   Let $BEGIN$ be the least element of $POSS$ and END be the max element of $POSS$.

2. If $|POSS| \leq s(p(n))$ then, for each $z \in POSS$, evaluate $\phi(z)$. If one of those z's is a satisfying assignment then output YES. If none of the z''s is a satisfying assignment then output NO.

3. If you got to this step then $|POSS| \geq s(p(n))+1$. Let $L = s(p(n))+1)$. We are going to break $POSS$ into L intervals of roughly equal size. Pick $z_1 \prec z_2 \prec \cdots \prec z_{L-1} \prec z_L$ such that the intervals $[BEGIN, z_1]$, $[z_2, z_3]$, $\ldots$ $[z_L, END]$ are all roughly the same size. For $1 \leq j \leq L$ let $w_j = f(\phi, z_j)$. Note that each interval has roughly $|POSS|/L$ elements.

   Note that by Exercise 2.2 and the definition of reduction we have that one of the following must occur, though note that we *do not know* which of these happens.

   (a) For all j, $w_j \in S$.

(b) For all j, $w_j \notin S$.

(c) There exists j, $1 \leq j \leq L$ such that $w_1, \ldots, w_j \notin S$ and $w_{j+1}, \ldots, w_L \in S$.

Thought experiment: Lets say we know that SOME $w_i \notin S$. Then we know that $w_1 \notin S$. Then we know that $(\phi, z_1) \notin LSAT$. Then we know that there is NO satisfying assignment for SAT in $[BEGIN, z_1]$. and hence can eliminate $[BEGIN, z_1]$ from $POSS$.

4. (a) *Case 1:* There exists $i < j$ such that $w_i = w_j$. We know $z_i \prec z_j$. Let $w = w_i = w_j$. Note that we DO NOT KNOW if $w \in S$ or not. But what do we know? Since $f(\phi, z_i) = f(\phi, z_j) = w$ we know that $(\phi, z_i) \in$ LSAT iff $(\phi, z_j) \in$ LSAT. Hence, if there is an satisfying assignment $z \preceq z_j$ then there is a satisfying assignment $z \preceq z_i$. Therefore we can eliminate $[z_i, z_j]$. We eliminate this NOT because none of these can satisfy ϕ, but because IF one of them satisfies ϕ,then some $z \preceq z_i$ satisfies ϕ. Output the set

$$POSS' = POSS - [z_i, z_j].$$

Note that

$$\begin{aligned} |POSS'| \leq & \ |POSS| - |POSS|/L \\ \leq & \ |POSS| \cdot (1 - \tfrac{1}{s(p(n))+1}) \end{aligned}$$

By Exercise 4 and the fact that $s(p(n)) + 1 \geq 10$ yields $|POSS| \cdot (1 - \frac{1}{s(p(n))+1})| \leq |POSS| \cdot 2^{-\frac{1}{s(p(n))+1}}$.
Recall that $POSS$ consisted of t intervals. Note that the set of possibilities being discarded is contiguous. It is easy to see that the number of intervals in $POSS'$ is at most $t + 1$.

(b) *Case 2:* $w_1, \ldots, w_L$ 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 w_j that is not in S. By the thought experiment above we can eliminate $[BEGIN, z_1]$ from $POSS$. Output the set

$$POSS' = POSS - [BEGIN, z_1].$$

By similar reasoning to that used in Case 1 we have

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

and that the number of intervals in $POSS'$ is at most $s+1$.

We now use algorithm $\mathcal{A}$ in a polynomial 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 $\mathcal{A}$ 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 in polynomial time. Hence $SAT \in P$. ∎

References

[1] S. Homer and L. Longpre. On reductions of NP sets to sparse sets. *Journal of Computer and System Sciences*, 48, 1994. Prior version in *S*TRUCTURES 1991.

[2] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the Twelfth Annual ACM Symposium on the Theory of Computing,* Los Angeles CA, pages 302–309, 1980.

[3] S. Mahaney. Sparse complete sets for NP: Solution to a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25:130–143, 1982.

[4] Ogiwara and Watanabe. On polynomial-time bounded truth-table reducibility of np sets to sparse sets. *SIAM Journal on Computing*, 20, 1991. Earlier version in STOC 1990.

[5] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.