Sparse Sets III: $SAT \le_T S \rightarrow \Sigma_2 = \Pi_2$
Exposition by William Gasarch

1 $SAT \le_{\mathrm{T}}^{\mathrm{p}} S$, S Sparse $\rightarrow \Sigma_2^{\mathrm{p}} = \Pi_2^{\mathrm{p}} = \mathrm{PH}$

Def 1.1 An *Oracle Turing Machine* (OTM) is a Turing Machine that can, in addition to the usual operations, ask questions of membership of some set, called an oracle. It is denoted $M^{()}$. One can define if formally in terms of states and alphabet and transitions; we leave this as an exercise. The important points about it are as follows.

1. An oracle Turing Machine $M^{()}$ is defined independent of the oracle you intend to run it with.
2. Questions are asked by writing a query to a special tape.
3. The expression $M^A(x)$ means that you run the Turing machine with oracle A on input x.
4. If we write an oracle algorithm the step "ask $z \in A$" is allowed. This will take time $|z|$, which is the time it takes to write the question on the oracle tape.

A *Polynomial Oracle Turing Machine* (POTM) is an OTM that runs in polynomial time.

We are concerned with when $SAT \le_T^{\mathrm{p}} S$ where S is sparse. We will actually look at Sparseness in a different way.

2 A Different View of Sparseness:ppoly

Def 2.1 A set A is in P/poly if there exists a polynomial p, a function $\mathrm{ADV} : 0^* \rightarrow \{0,1\}^*$, and a polynomial predicate B such that the following hold.

1. For all n, $\mathrm{ADV}(0^n) \in \{0,1\}^{p(n)}$.
2. For all n
$$A \cap \{0,1\}^{\le n} = \{x \mid B(x, \mathrm{ADV}(0^n))\}.$$

We think of the string $\mathrm{ADV}(0^n)$ as giving advice for all strings of length $\le n$. The class P/poly is often referred to as 'poly time with advice'.

We leave the following as an exercise.

Lemma 2.2 *Let $A \subseteq \{0,1\}^*$. The following are equivalent.*

1. $A \le_{\mathrm{T}}^{\mathrm{p}} S$ *where* S *is sparse set.*
2. $A \in \mathrm{P/poly}$.

3 A Different View of Sparseness:Circuits

Def 3.1 Fix n. A *circuit on n inputs* is just what you think it is: n inputs and then AND, OR and NOT gates, and a final output gate. Note that these can only compute a function on $\{0,1\}^n$.

We will define a circuit (of a bounded size) to decide a set if there is a diff circuit for each n.

Def 3.2 Let $s(n)$ be a function. A *circuit of size $s(n)$* is a SEQUENCE of circuits $C_1, C_2, \ldots$.

Def 3.3 A set $A \subseteq \{0,1\}^n$ has a *circuit of size $s(n)$* if there is a sequence of circuits $C_1, C_2, C_3, \ldots$ such that (1) C_n has at most $s(n)$ gates, and (2) C_n computers A restricted to strings of length n.

We leave the following as an exercise.

Lemma 3.4 *Let $A \subseteq \{0,1\}^*$. The following are equivalent.*

1. $A \le_{\mathrm{T}}^{\mathrm{p}} S$ *where* S *is sparse set.*
2. $A \in \mathrm{P/poly}$.
3. *A has poly sized circuits.*

4 Main Theorem

We will express the theorem in terms of circuits.

Def 4.1 Let $FINDSAT$ be the following function:

1. If $\phi \in SAT$ then $FINDSAT(\phi)$ is the lex least satisfying assignment of ϕ.
2. If $\phi \notin SAT$ then $FINDSAT(\phi)$ outputs 0^v, where v is the number of variables in ϕ.

We leave the following proof as an easy exercise.

Lemma 4.2 *If* SAT *has poly sized circuits then* $FINDSAT$ *has poly sized circuits.*

We can now prove our main theorem.

Theorem 4.3 *If* SAT *has poly sized circuits then* $\mathrm{PH} = \Sigma_2^{\mathrm{p}} = \Pi_2^{\mathrm{p}}$.

Proof:
Let $A \in \Pi_2^p$. Then there exists $B \in \mathrm{NP}$ such that

$$A = \{x \mid (\forall^p y)[(x, y) \in B]\}.$$

By the Cook-Levin Theorem there exists a function in poly that takes x, y and maps to a formula $\phi_{x,y}$ such that

$$(x, y) \in B \text{ iff } \phi_{x,y} \in SAT.$$

Hence

$$A = \{x \mid (\forall^p y)[\phi_{x,y} \in \mathrm{SAT}]\}.$$

Since $FINDSAT$ has poly sized circuits we know THERE EXISTS a circuit that computes $FINDSAT(\phi_{x.y})$. But here is the key— the alleged circuit outputs an assignment THAT CAN BE TESTED!

We claim

$$A = \{x \mid (\exists^p C)(\forall^p y)[\phi_{x,y}(C(\phi_{x,y}))]\}.$$

Why is this?

If $x \in A$ then for all y, $\phi_{x,y}$ is satisfiable. Hence the CORRECT C that computes FINDSAT will always find an assignment that works.

If there is a circuit C such that $(\forall^p y)[\phi_{x,y}(C(\phi_{x,y}))]\}$ then clearly (whether or not C is the real circuit that computes FINDSAT) $(\forall^p y)[\phi_{x,y}(C(\phi_{x,y}))]\}$. Hence $x \in A$.

We have taken a Π_2^{p} set and showed it isin Σ_2^{p}. Hence $\Pi_2^{\mathrm{p}} \subseteq \Sigma_2^{\mathrm{p}}$. Complement both sides to obtain $\Sigma_2^{\mathrm{p}} \subseteq \Pi_2^{\mathrm{p}}$. Hence $\Pi_2^{\mathrm{p}} = \Sigma_2^{\mathrm{p}}$. ∎