Logic and Computation Seminar

Bob Chen

Fall 2011

1 10/12/11

Definition 1.1. The class Parity$\mathbb{P}$, denoted $\oplus\mathbb{P}$, is the collection of languages L whose members have a number of verifying strings congruent to 1 modulo 2.

The class Probabilistic$\mathbb{P}$, denoted $P\mathbb{P}$, is the collection of languages whose members have at least half of possible strings act as verification strings.

Remark 1.2. We have $\oplus P \subset \#\mathbb{P}, \mathbb{P}^{\#\mathbb{P}}$, and also $P\mathbb{P} \subset P^{\#\mathbb{P}}$.

Example 1.3. Show that $\#\mathbb{P} \in F\mathbb{P}^{P\mathbb{P}}$.

Theorem 1.4. *(Toda.) The polynomial heirarchy is contained in $\mathbb{P}^{\#\mathbb{P}}$.*

Definition 1.5. *Propositional formulas* (or Boolean formulas) are built from variables $x, y, z, \ldots, p, q, r, \ldots$ and connectives $\wedge, \vee, \neg$. The variables can be assigned values of T, F.

A *truth assignment* is a map $\tau : \{\text{variables}\} \to \{1, 0\}$. If φ is a formula, the natural extension allows us to interpret $\tau(\varphi)$. We say φ is *satisfiable* if there exists a truth assignment τ so that $\tau(\varphi) = 1$.

A *literal* is a variable x or its negation $\overline{x}$. A *clause* is the disjunction($\bigvee$) of literals, and a *conjunctive normal form* formula is a conjunction ($\bigwedge$) of clauses.

Theorem 1.6. *(Cook.) Let M be a polynomial time NTM. Then there exists a polynomial time $f(x)$ such that for all strings x, $f(x)$ is (codes) a propositional formula with the same number of satisfying assignments as the number of accepting computations of $M(x)$.*

Corollary 1.7. *SAT is (many-one) $\mathbb{NP}$-complete (that is, it's in $\mathbb{NP}$ and also any $\mathbb{NP}$ problem can reduced to a SAT problem).*

Remark 1.8. The other type of completeness is Turing completeness (for $\mathbb{NP}$), which would say that every $\mathbb{NP}$ problem can be reduced to some number of SAT problems.

Proof. (Sketch.) Suppose M is an NTM, clocked explicitly and all computations have the same length and every state has nondeterminism. Moreover, suppose wehave only 1 tape that only moves to the right (all this can be done WLOG).

Create the variables t_{ijl} which is true if and only if symbol l is on tape square i at time j. Likewise, we create variables h_{ij} indicating the position of the machine head and variables s_{kj} indicating the state of the machine.

Now just get your hands dirty and write a humongous formula relating all these variables according to the computation rules of M. We claim that we can build this formula in polynomial time, and the satisfying assignments correspond exactly to the accepting computations of M.

(So, we glossed over the fact that we should be able to figure out what the nth character of the string in polylog time, so that the whole thing can be parallelized efficiently. But we'll save that for another day.) □

Definition 1.9. $\oplus$SAT is the set of formulas such that the number of satisfying assignments is odd.

Corollary 1.10. *$\oplus$SAT is many-one complete for $\oplus\mathbb{P}$.*

Definition 1.11. UniqueSAT is the following problem: On input φ, a formula, output 1 if φ has exactly one satisfying assignment. If φ has none, then output 0. Otherwise, you can output whatever.

Remark 1.12. Some authors use USAT to denote the the problem where you must output 0 if there is more than 1 satisfying assignment.

Theorem 1.13. *(Valiant-Vazirani.) There is a probabilistic polytime (PPT) algorithm f such that for all $\varphi(x_1, \ldots, x_n)$, if $\varphi \in SAT$ then $f(\phi) \in USAT$ with probability at least $\frac{1}{8n}$. On the other hand, if $\varphi \notin SAT$ then $f(\varphi)$ is never in SAT.*