CIS 511: Theory of Computation	**Feb 23, 2017**

Lecture 13: Wrapping up NP and Beginning Space Complexity

Professor Sampath Kannan — *Zach Schutzman*

NB: *These notes are from CIS511 at Penn. The course followed Michael Sipser's Introduction to the Theory of Computation (3ed) text.*

More NP-$Complete$ness

Other problems are NP-$Complete$, the reduction for $3COLOR$ creates gadgets from each clause and each variable (see any Algorithms book).

There is an obvious reduction from $3COLOR$ to $4COLOR$ (and beyond). Add a new vertex, connect it to every vertex in your original graph.

$SUBSETSUM$, the question of whether a set of numbers contains a subset that sums to a particular value k is NP-$Complete$. The reduction is from $3SAT$. Given a formula ϕ with m clauses and n variables, we create $2n$ numbers with $n+m$ base 7 digits. Each variable gets a 1 in positions indicating the index of the variable and the clauses it appears in. ϕ is satisfiable if and only if there is a subset that sums to $11\ldots111$ in the first n positions, and $4\ldots44$ in the remaining m positions, by introducing some dummy numbers which are all zeros in the first n positions and 1 *or* 2 in the corresponding clause position.

We know that $P \subseteq NP$, NP-$Complete \subset NP$

Definition 13.1 *A language L is in Co-NP if its complement is in NP. Equivalently, given $x \notin L$, a verifier can check non-membership given the right certificate (easy to check that something is not a solution).*

Example: $TAUT = \{\phi|\phi \text{ is } satistfied \text{ by every assignment}\}$ *is in Co-NP. If a ϕ is not in L, then any non-satisfying assignment can be quickly checked.*

We don't know if $NP =$ Co-NP.

If $P = NP$, then $NP =$ Co-NP.

If $NP \neq$ Co-NP, then $P \neq NP$.

Space Complexity

Definition 13.2 Space complexity *refers to the number of cells of tape scanned by the head of a Turing machine in running an input.*

Definition 13.3 *$DSPACE(s(n))$ is the set of languages recongized by a deterministic TM in $O(s(n))$ space.*

Definition 13.4 *$NSPACE(s(n))$ is the set of languages recognized by a non-deterministic TM in $O(s(n))$ space.*

Let's look at $SAT \in NP$. *What is its space complexity? Let's say* n *is the length of the input and* k *is the number of variables. If we don't care about time, we can represent the assignment as a* k*-bit number and try one assignment at a time. We only need* $n + k + c$ *(where* c *is some small fixed workspace). Therefore,* $SAT \in DSPACE(n)$.

Let $L_{NANFA} = \{\langle M \rangle | M \text{ is an } NFA,\ L(M) \neq \Sigma^*\}$.