CIS 511: Theory of Computation	**Feb 30, 2017**

Lecture 15: PSPACE Completeness

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.*

The Class *PSPACE-Complete*

Definition 15.1 *A language L is PSPACE-Complete if L is in PSPACE and for all $A \in PSPACE$, there exists a mapping reduction from A to L in polynomial time.*

Definition 15.2 *A* ***quantified boolean formula*** *is a formula of the form $\forall x_1 \exists x_2 \forall x_3 \ldots \phi(x_1, x_2, \ldots)$, where ϕ is a boolean formula over the x_i.*

Claim 15.3 *The language of True Quantified Boolean Formulas (TQBF) is PSPACE-Complete.*

Proof:

We'll think of a machine that decides this language as taking a quantified boolean formula as input and accepting if it is true, rejecting if it is false.

This language is obviously in PSPACE, we can simply check each allowable assignment and accept if the formula is true.

To show that it is PSPACE-Complete, we need to show that every other PSPACE language can be reduced to it in polynomial time. Let L be a language in PSPACE. Without loss of generality, we can say that a decider for L takes $O(2^{n^k})$ time.

We saw in the Cook-Levin Theorem how to think of states of computation as boolean formulas, and we will do something similar here. Let C_{init} the initial configuration and C_{accept} as the canonical accepting configuration. We write $\Phi(C_a, C_b, t)$ to denote the quantified boolean formula corresponding to 'there exists a sequence of computation steps of length at most t such that the machine moves from configuration C_a to C_b. We want to find formula equivalent to $\Phi(C_{init}, C_{accept}, 2^{n^k})$ which is true if and only if the input x to the decider for L is in L.

We start by observing that if $x \in L$, then there must be some intermediate configuration C_m which the machine passes through in its computation. We can then write

$$\Phi(C_{init}, C_{accept}, 2^{n^k}) \iff \Phi(C_{init}, C_m, \frac{2^{n^k}}{2}) \wedge \Phi(C_m, C_{accept}, \frac{2^{n^k}}{2})$$

We can also consider quarter-way and eighth-way points, and so on and do the same thing, however, we get an exponentially large formula by the time we get down to step sizes of $t = 1$.

We get around this by introducing two new variables C_3, C_4 and universally quantifying them over our configuration pairs $(C_1, m), (m, C_2)$ such that we have

$$\Phi(C_1, C_2, t) = \exists m \forall (C_3, C_4) \in \{(C_1, m), (m, C_2)\}(\Phi(C_3, C_4, \frac{t}{2}))$$

We only need polynomial space for this, hence L is reducible to $TQBF$, and $TQBF$ is $PSPACE$-Complete.
■

Example: *The language Generalized Geography (GG) is $PSPACE$-Complete. GG is a two player game on a directed graph. One player picks a node, then the other picks a node for which there exists an arc from the previous player's choice into that node. A player loses if there are no nodes for her to legally choose (i.e. the previous player picked a node with out degree zero). Player One's first choice is a designated start node. The language GG is the decision problem of whether, given a directed graph with a start node, Player One can always win given perfect play.*

We can think of this as a $TQBF$ in the following way: Player One has a winning strategy if there exists a node with out-degree zero such that for any previous choice made by Player Two there exists a previous choice by Player One, and so on.