CS 221: Computational Complexity	**Prof. Salil Vadhan**
Lecture Notes 6	
Feb 10, 2010	**Scribe: Rebecca A. Resnick**

Agenda:
PSPACE
Alternation
PH
Time-Space tradeoffs for SAT.

1 TQBF

Consider the language of TRUE QUANTIFIED BOOLEAN FORMULAS (TQBF), i.e. TQBF = {true statements $\exists x_1 \in \{0,1\}^{n_1} \forall x_2 \in \{0,1\}^{n_2} \ldots \mathbb{Q}_m x_m \in \{0,1\}^{n_m} \phi(x_1, \ldots x_m)$ where ϕ is a 3-CNF formula}.

Theorem 1 TQBF *is* **PSPACE**-*complete wrt* $\leq_{Cook}$

Proof: By the following 3 lemmas:

Lemma 2 TQBF $\in$ **PSPACE**

Proof: 2-word proof: Recursive evaluation. Try $x_1 = 0$ and $x_1 = 1$, see if remaining formula is satisfiable. Actually a **SPACE**(n) algorithm, because depth of recursion is number of elements you have to try. ■

Lemma 3 **PSPACE** $\in$ **AP**, *"alternating* **PSPACE***"*

First, a few definitions: An **alternating TM (ATM)** is like an NTM except each state is labeled as either $\exists$ or $\forall$.

Some notes:

1. Running time: maximum length computation path as a function of the input length (just like NTM)
2. Acceptance condition on x (see figure below). Nodes of the computation tree are defined to be accepting or rejecting recursively:
   (a) leaf: is configuration accepting or rejecting?
   (b) $\exists$: accepting if at least one child is accepting
   (c) $\forall$: accepting if all children are accepting
   (d) Computation accepts $\iff$ start configuration on x is accepting.

3. (NTMs = ATMs with only $\exists$ nodes)
4. $\mathbf{ATIME}(t(n))$ and $\mathbf{AP}$ are defined in natural way, i,e:

$$\mathbf{AP} = \bigcup_k \mathbf{ATIME}(n^k)$$

Comment from the audience: This seems like a boolean circuit with AND/OR gates. Answer: You never have construct entire tree with an ATM; you only care about the runtime of a particular path (whereas in Boolean circuits the main complexity measure is size of the entire circuit, though depth is also considered). Also note that the "leaves" here are not bits of the ATM's inputs, but whether or not the ATM accepts or rejects at the end of a particular computation path. Compare with standard non-determinism: an ATM with ONLY existential nodes. Nevertheless, the similarity between the two models can be exploited - people have used results about boolean circuits to prove results about the power of ATM's "relative to oracles."

Now proceed with pf of lemma.

Proof of Lemma 3: Given L decided by a PSPACE algorithm M, we will give an AP algorithm for $\text{Reach}_{G_{M,x}}(u, v, i)$. $G_{M,x}$ is the configuration graph of M on x. (Since M is deterministic, $G_{M,x}$ is really just a path.) $i \in \{1, \dots 2^{\text{poly}(n)}$. $\text{Reach}_{G_{M,x}}(u, v, i) = 1$ if there exists a path $u \to v$ in $G_{M,x}$ of $\leq i$ steps.

AP algorithm to compute $\text{Reach}_{G_{M,x}}(u, v, i)$: Base cases: left to the reader.
$\exists$: nondeterministically guess a configuration w.
$\forall$: Check both $\text{Reach}_{G_{M,x}}(u, w, \lceil i/2 \rceil)$ and $\text{Reach}_{G_{M,x}}(w, v, \lfloor i/2 \rfloor)$.
Running time is depth of tree, which is polynomial because i is shrinking by half at each step ■

Lemma 4 TQBF *is* $\mathbf{AP}$*-hard.*

Proof: $L \in \mathbf{AP} \iff \exists$ poly-time M, polynomials q, r, such that

$$x \in L \iff \exists u_1 \forall u_2 \dots Q_r u_r (M(x, u_1, \dots, u_r)), u_i \in \{0, 1\}^{q(|r|)}.$$

By Cook-Levin, we can convert $M(x, u_1, \dots u_r)$ to a 3-CNF, $\exists z \phi_{M,x}(u_1, \dots, u_r, z)$, where $\phi_{M,x}$ is constructed from M, x in logspace ■

This concludes the proof that TQBF is $\mathbf{PSPACE}$-complete wrt $\leq_l$ ■

Corollary 5 ("alternating time equals space") PSPACE = AP

Fact 6 ("alternating space equals exponentially more time") AL = P *and,* **APSPACE = EXP**

Corollary 7 $\exists \epsilon > 0$ *such that* TQBF $\notin \mathbf{SPACE}(n^\epsilon)$ *(suffices to take $\epsilon \leq .49$)*

Proof: By the Space Hierarchy Thm there exists some language L solvable in linear space, but NOT solvable in sub-linear space. Since TQBF is **PSPACE** complete, L reduces to TQBF in logspace, so if TQBF $\in \mathbf{SPACE}(n^\epsilon)$, then $L \in \mathbf{SPACE}((n^c)^\epsilon + \log(n))$. If $\epsilon < 1/c$, we have a contradiction. ■

2 Polynomial Hierarchy

Define $\mathbf{\Sigma_k TIME}(t(n)) = \{$ languages decided by ATMs with $\leq k-1$ alternations between $\exists$ and $\forall$ on each computation path, time $\leq t(n)$, starting with $\exists\}$, and

$\mathbf{\Pi_k TIME}(t(n)) = \{$ languages decided by ATMs with $\leq k-1$ alternations between $\exists$ and $\forall$ on each computation path, time $\leq t(n)$, starting with $\forall\}$. Also define

$$\mathbf{\Sigma_k^p} = \bigcup_c \mathbf{\Sigma_k TIME}(n^c),$$

$$\mathbf{\Sigma_k^p} = \bigcup_c \mathbf{\Pi_k TIME}(n^c).$$

2.1 Motivation

1. Natural Problems (at low levels):

$$\textsc{Circuit Minimization} = \{\langle C, k\rangle : \exists C'(|C'| \leq k \wedge \forall x C'(x) = C(x)\}.$$

2. $\mathbf{PH}$ = class of languages L for which we currently know how to prove that if $\mathbf{P} = \mathbf{NP}$, then $L \in \mathbf{P}$.

3. Useful for lower bounds on $\mathbf{NP}$. e.g $\mathbf{PH} \neq \mathbf{P} \Rightarrow \mathbf{NP} \neq \mathbf{P}$ (later),

   also, known: $\mathbf{\Sigma_4 TIME}(n) \neq \mathbf{DTIME}(n)$ and thus, $\mathbf{NTIME}(n) \neq \mathbf{DTIME}(n)$.

3 Alternating Characterizations

1. $L \in \mathbf{\Sigma_k^p} \iff \exists$ polynomial q, poly-time M such that $x \in L \iff \exists u_1 \forall u_2 \ldots Q_k u_k M(x, u_1, \ldots u_k)$, $u_i \in \{0,1\}^{q(|x|)}$

2. $\Sigma_k\text{SAT} = \{\exists x_1 \forall x_2 \ldots Q_k x_k \phi(x_1, \ldots x_k)\}$ is Σ_k^p-complete. ϕ is a 3-CNF if $Q_k = \exists$, 3-DNF if $Q_k = \forall$.

Theorem 8 *1.* $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$

2. $\mathbf{\Sigma_k^p} = \mathbf{\Pi_k^p} \rightarrow \mathbf{P} = \mathbf{\Sigma_k^p} = \mathbf{\Pi_k^p}$ *(conjectured to be false)*

Proof:

1. Assume $\mathbf{P} = \mathbf{NP}$. Let $L \in \mathbf{\Sigma_k^p}$. By first characterization, there exists a poly-time M_0 such that $x \in L \iff \exists u_1 \forall u_2 \ldots \forall_k u_k M_0(x, u_1, \ldots u_k)$, $u_i \in \{0,1\}^{q(|x|)}$.

   Since $\mathbf{P} = \mathbf{NP}$, intuitively, we can replace statements that have 1 quantifier with quantifierless statements. In particular since $\forall_k u_k M_0(x, u_1, \ldots u_k) \in \mathbf{co\text{-}NP} = \mathbf{P}$ we can replace $\forall_k u_k M_0(x, u_1, \ldots u_k)$ with $M_1(x, u_1, \ldots u_{k-1})$, where M_1 is an $\mathbf{NP}$-time algorithm. Repeating this step $k-1$ more times, we get $x \in L \iff M_k(x) = 1$

2. On problem set 2.

$\blacksquare$

Remark 9 1. Suppose $\mathbf{NTIME}(n) \subseteq \mathbf{DTIME}(f(n)) \rightarrow \mathbf{\Sigma_k TIME}(t(n)) \subseteq \mathbf{TIME}(f(f \ldots f(t(n)) \ldots))$. Eg $f(n) = n^2 \Rightarrow \mathbf{DTIME}(t(n)^{2^k})$ but if $f(n) = n^{1+o(1)} \Rightarrow \mathbf{DTIME}(t(n)^{1+o(1)})$.

2. Above shows that SAT $\in \mathbf{P} \Rightarrow \Sigma_k$SAT $\in \mathbf{P}$. But we haven't given a *reduction* from Σ_kSAT to SAT! Indeed, it can be shown that if there were a Cook reduction from Σ_kSAT to SAT, then the $\mathbf{PH}$ collapses to $\mathbf{P^{NP}} = \mathbf{\Delta_1^P}$.