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 \exists x_m \in \{0,1\}^{n_m} \phi(x_1,\ldots,x_m) \} where \phi is a 3-CNF formula.

**Theorem 1** TQBF is PSPACE-complete wrt \( \leq_{\text{Cook}} \)

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

\[
\text{AP} = \bigcup_k \text{ATIME}(n^k)
\]

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.) \( i \in \{1, \ldots, 2^{\text{poly}(n)}\} \). \( \text{Reach}_{G_{M,x}}(u, v, i) = 1 \) if there exists a path \( u \rightarrow v \) in \( G_{M,x} \) of \( \leq i \) steps.

**AP algorithm to compute 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 \( \text{AP}-\text{hard} \).

**Proof:** \( L \in \text{AP} \iff \exists \) poly-time \( M \), polynomials \( q, r \), such that

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

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

This concludes the proof that TQBF is \( \text{PSPACE}\)-complete wrt \( \leq_l \)

**Corollary 5** ("alternating time equals space") \( \text{PSPACE} = \text{AP} \)

**Fact 6** ("alternating space equals exponentially more time") \( \text{AL} = \text{P} \) and \( \text{APSPACE} = \text{EXP} \)

**Corollary 7** \( \exists \epsilon > 0 \) such that \( \text{TQBF} \not\in \text{SPACE}(n^\epsilon) \) (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 \( \text{PSPACE} \) complete, \( L \) reduces to TQBF in logspace, so if TQBF \( \in \text{SPACE}(n^\epsilon) \), then \( L \in \text{SPACE}((n^\epsilon)^\epsilon + \log(n)) \). If \( \epsilon < 1/c \), we have a contradiction.
2 Polynomial Hierarchy

Define $\Sigma_k \text{TIME}(t(n)) = \{ \text{languages decided by ATMs with } \leq k - 1 \text{ alternations between } \exists \text{ and } \forall \text{ on each computation path, time } \leq t(n), \text{ starting with } \exists \}$, and $\Pi_k \text{TIME}(t(n)) = \{ \text{languages decided by ATMs with } \leq k - 1 \text{ alternations between } \exists \text{ and } \forall \text{ on each computation path, time } \leq t(n), \text{ starting with } \forall \}$. Also define $\Sigma^p_k = \bigcup_c \Sigma_k \text{TIME}(n^c)$, $\Sigma^p_k = \bigcup_c \Pi_k \text{TIME}(n^c)$.

2.1 Motivation

1. Natural Problems (at low levels):

   \[
   \text{Circuit Minimization} = \{\langle C, k \rangle : \exists C' | |C'| \leq k \land \forall C C(x) = C'(x)\}.
   \]

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

3. Useful for lower bounds on $\text{NP}$, e.g $\text{PH} \neq P \Rightarrow \text{NP} \neq P$ (later), also, known: $\Sigma_4 \text{TIME}(n) \neq \text{DTIME}(n)$ and thus, $\text{NTIME}(n) \neq \text{DTIME}(n)$.

3 Alternating Characterizations

1. $L \in \Sigma^p_k \iff \exists$ polynomial $q$, poly-time $M$ such that $x \in L \iff \exists u_1 \forall u_2 \ldots \forall k u_k M(x, u_1, \ldots u_k), \quad u_i \in \{0, 1\}^{q(|x|)}$

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

Theorem 8

1. $P = \text{NP} \Rightarrow \text{PH} = P$

2. $\Sigma^p_k = \Pi^p_k \Rightarrow P = \Sigma^p_k = \Pi^p_k$ (conjectured to be false)

Proof:

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

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

Remark 9  

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

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