**Def.** A Quantified Boolean Formula (QBF) \( \phi \) has quantifier 

\[ \forall x_1 \text{ and } \exists x_2, \text{ in addition to } \forall, \exists, \overline{x}, \overline{x} \text{ over variables.} \]

Usually we suppose all quantifiers are in front: \( \exists x_1 (\exists x_2)(\exists x_3)(\exists x_4)\cdots(\exists x_n)\phi \)

Where \( \phi \) is an ordinary Boolean formula in \( x_1 \rightarrow x_n \) without quantifiers.

Define \( \text{TQBF} = \exists^* \text{QBF} \). \( \psi \) is true.

(Usually people drop the \( T \) and just call it QBF)

We can kind of consider SAT and TAUT to be subsets of QBF, because:

\( \phi(x_1, \ldots, x_n) \) is satisfiable if the QBF \( (\exists x_1)(\exists x_2)\cdots(\exists x_n)\phi \) is true.

\( \phi(x_1, \ldots, x_n) \) is a tautology, if the QBF \( (\forall x_1)(\forall x_2)\cdots(\forall x_n)\phi \) is true.

Precisely, \( \text{SAT} \leq^T \text{QBF} \) and \( \text{TAUT} \leq^T \text{QBF} \).

In fact, \( \leq^T \text{QBF} \) is true.

In fact, \( \leq^T \text{QBF} \) is true.

**Theorem:** TQBF is complete for PSPACE under \( \leq^T \text{QBF} \) and hence under \( \leq^T \).

**Proof:** In PSPACE:

\[ (\forall x_n)(\phi) \]

Use O(n) steps to track the current branch in \( O(n) \) and its status of the quantified nodes along current branch.

But from ALR it's 25 s.c. Accept if the root becomes good.

All is done in \( O(n) \) space.
Completeness: Let any \( A \in \text{DTIME}[2^{O(n)}] \) be given, where \( S(n) = 2^{O(n)} \). Then not only is \( A \in \text{DTIME}[2^{O(n)}] \)

but the maximum length \( R(n) \) of any computation path that accepts (without checking) is \( 2^{O(n)} \) where \( \text{run}(A) = O(S(n)) \).

Idea: "A journey of a thousand miles... has a step that is exactly 500 miles from each side."

Take a DTM \( M \) such that \( L(M) = A \) and on \( x \in \Sigma^* \) any accepting (comp) has length at most \( 2^{cn} \) steps.

We can write a Boolean formula \( \phi_M \) with variables representing bit encoding \( D_k, I, J \) such that an assignment to these variables satisfies \( \phi_M \) if and only if the corresponding \( D_k, I, J \) come from some assignment to those variables.

We will say \( \phi_M(I, J) \equiv \text{I goes to } J \) in at most \( k \) steps.

Now for \( k \geq 1 \) define a formula \( \phi_k(I, J) \equiv \text{I goes to } J \) in at most \( 2^k \) steps recursively by:

\[
\phi_1(I, J) = (\exists k)(\forall k') (k' = k \land \phi_k(I, J))
\]

Problems: The number of recursive branches grows \( \phi_{k+1} \) up to \( 2^k \times 2^k = 2^{2k} \) which is too big.

Imagine: \( 2 \) marble balls, \( \exists k, k' \) to pick.

\[
(\exists k)(\forall k') (k' = k \land \phi_k(I, J)) \rightarrow \phi_k(I, J)
\]

Now, we define \( \text{HP} \) where \( \text{run}(A) = O(S(n)) \).

Not only does this have only one recursive branch, but the recursion to produce the final output formula

\[
\phi(I_0, I_f)
\]

\( \text{HP} = \text{NP} \) if and only if \( \text{TATU} \leq_m \text{SAT} \).

\( \text{TATU} \leq_m \text{NP} \) if \( A \leq_m \text{NP} \) then \( \text{TATU} \leq_m \text{SAT} \).

In fact, \( M \) can be an NFA, so \( \text{NLFA} \leq_m \text{TATU} \) in \( O(n^2) \) time.