In this lecture, we will discuss Nondeterministic Time Hierarchy Theorem. Recall the Deterministic Time Hierarchy Theorem.

**Theorem:** (Deterministic Time Hierarchy Theorem) If \( t \) is a time-constructible function, then \( \text{DTIME}(t(n)) \subset \text{DTIME}(t^2(n)) \) (also \( \text{DTIME}(t(n)) \subset \text{DTIME}(t(n) \log^2 t(n)) \)).

Note that \( \subset \) is used for proper inclusion.

For the proof of this theorem, we used the diagonalization argument. The basic idea is to reject when the deterministic TM accepts, and to accept otherwise. However, we cannot apply the diagonalization argument directly in the case of nondeterministic machines. There are some accepting paths and some rejecting paths in a computation with nondeterministic Turing machines.

**Theorem:** (Nondeterministic Time Hierarchy Theorem) Let \( T \) be a time constructible function.

If \( t(n+1) = o(T(n)) \), then \( \text{NTIME}(t(n)) \subset \text{NTIME}(T(n)) \).

**Corollary:** \( \text{NTIME}(n) \subset \text{NTIME}(n \log^* n) \).

**Open Question:** \( \text{DTIME}(n) =? \text{DTIME}(n \log^* n) \).

Note that \( \text{DTIME}(2^{2^n}) \subset \text{DTIME}((2^{2^n})^2) \).

On the other hand, the following equality is still an open question:

\[ \text{NTIME}(2^{2^n}) =? \text{NTIME}((2^{2^n})^2) \]

Thus, for “small” time bounds the nondeterministic time hierarchy is tighter, and for “large” time bounds the deterministic time hierarchy is tighter.

**FACT:** If \( A \in \text{NTIME}(t(n)) \), then \( A \) is accepted by a Nondeterministic Turing Machine with 2 tapes in time \( t(n) \).

**Proof of the Nondeterministic Hierarchy Theorem:**

Let \( M_1, M_2, \cdots \) be an enumeration of two-tape Turing machines of time complexity \( t(n) \).

Let \( f \) be a rapidly growing function (such as \( f(i, n, s) = 2^{2^{i+s+s+s}} \)) such that the predicate

\[
(i, n, s) \mapsto \begin{cases} 
1 & \text{if } M_i \text{ accepts } 1^n \text{ in } \leq s \text{ steps} \\
0 & \text{otherwise}
\end{cases}
\]

is computable deterministically in time \( f(i, n, s) \).

Now let us divide the natural numbers into regions by defining \textit{start} and \textit{end} of each region as follows:

\[
\begin{align*}
\text{start}(1) &= 1 \\
\text{start}(i + 1) &= \text{end}(i) + 1 \\
\text{end}(j) &= f(i, \text{start}(j), T(\text{start}(j))) \text{ where } j = (i, y)
\end{align*}
\]
Note that in region \( j = (i, y) \), we try to fool machine \( M_i \), and on input \( 1^{\text{end}(j)} \), a deterministic machine can, in time \( T(\text{end}(j)) \), determine whether \( M_i \) accepts \( 1^{\text{start}(j)} \) in at most \( T(\text{start}(j) - 1) \) steps.

Let us look at the code.

on input \( 1^n \)

Find \( j = (i, y) \) such that \( \text{start}(j) \leq n \leq \text{end}(j) \) (I)

If \( n = \text{end}(j) \)

then

accept iff (II) \( M_i \) does not accept \( 1^{\text{start}(j)} \) in \( \leq T(\text{start}(j)) \) steps.

else

accept iff (III) our universal NTM accepts \( (i, 1^{n+1}) \) in \( T(n) \) steps.

Step (I) takes \( O(n) \) time, step (II) takes deterministically \( T(n) \) steps, step (III) takes \( T(n) \) steps.

Let \( A \) be the language accepted by the above algorithm. Clearly, \( A \in \text{NTIME}(T(n)) \).

We claim that \( A \notin \text{NTIME}(t(n)) \).

Assume that \( A \in \text{NTIME}(t(n)) \). Let \( M_i \) be the nondeterministic machine accepting \( A \) in time \( t(n) \), namely, \( A = L(M_i) \) where \( t_i(n) \leq t(n) \).

Consider \( j = (i, y) \) where \( y \) is so big that

\[
\forall n \geq j, t(n + 1) \leq T(n)/i^3.
\]

Consider \( n \) such that \( \text{start}(j) \leq n \leq \text{end}(j) \).

\[
1^n \in A \tag{1}
\]

iff \( U \) accepts \( (i, 1^{n+1}) \) in \( \leq T(n) \) steps \( \tag{2} \)

iff \( M_i \) accepts \( (i, 1^{n+1}) \) in \( T(n)/i^3 \) steps \( \tag{3} \)

iff \( M_i \) accepts \( 1^{n+1} \) in \( T(n)/i^3 \) steps, \( M_i \) accepts \( 1^{n+1} \) in \( t(n + 1) \) steps \( \tag{4} \)

iff \( 1^{n+1} \in A \). \( \tag{5} \)

Therefore, \( 1^n \in A \) iff \( 1^{n+1} \in A \). This contradicts the fact that \( 1^{\text{start}(j)} \in A \) if and only if \( 1^{\text{end}(j)} \notin A \). \( \square \)

Let us check both directions for (2) iff (3) above.

\((\Rightarrow) \) If \( M_i \) does not accept in \( T(n)/i^3 \) steps, then \( M_i \) does not accept it at all, since \( t(n + 1) < T(n)/i^3 \), so \( U \) does not accept \( (i, 1^{n+1}) \).

\((\Leftarrow) \) If \( M_i \) does accept in \( T(n)/i^3 \) steps, then \( U \) has time to compute the simulation. \( \square \)

Next time, we will talk about Immerman-Szelepczeny and Savitch Theorems.