Proof of Time Hierarchy Theorem

Definition 1. *Let $f(n) \geq n$ be a proper complexity function, and define H_f to be the following time-bounded version of the HALTING language H:*

$$H_f = \{M; x : M \text{ accepts input } x \text{ after at most } f(|x|) \text{ steps }\}$$

where M ranges over all descriptions of multi-string TMs.

Lemma 1. $H_f \in TIME((f(n))^3)$.

Proof of Lemma

We give a 4-string TM U_f which decides H_f in time $(f(n))^3$:

1. First U_f uses M_f on x to initialize its 4th string by $\#^{f(|x|)}$ — "clock".

U_f copies the description of M on it 3rd string and second string is initialized to initial state s and first string is initialized to $\triangleright x$.

Total time is $O(f(|x|) + n) = O(f(n))$.

2. U_f simulates M on x step by step. It also advances the clock by one.

Each step of M takes $O(\ell_M k_M^2 f(|x|))$, where k_M is the number of strings of M and ℓ_M is the length of the description of each state and symbol of M.

k_M and ℓ_M are bounded above by $\log n$.

Thus time to simulate a step of M is $O((f(n))^2)$.

3. U_f accepts/rejects its input based on the clock. Total time is $O((f(n))^3)$.

The constant can be removed by modifying U_f to treat several symbols as one.

Lemma 2. $H_f \notin TIME(f(\lfloor \frac{n}{2} \rfloor))$.

Proof: By contradiction. Assume that there is a TM M_{H_f} that decides H_f in time $f(\lfloor \frac{n}{2} \rfloor)$.

Construct the following machine D_f:

$D_f(M)$: if $M_{H_f}(M, M) = yes$ then no else yes

D_f on input M runs in $f(\lfloor \frac{2n+1}{2} \rfloor) = f(n)$.

What about $D_f(D_f)$?

If $D_f(D_f) = yes$ then $M_{H_f}(D_f, D_f) = no$, i.e., $(D_f; D_f) \notin H_f$.

Thus D_f fails to accept its own description in $f(n)$ steps — implies $D_f(D_f) = no$.

Similarly, $D_f(D_f) = no$ implies $D_f(D_f) = yes$.

Contradiction in both cases.

The Time Hierarchy Theorem follows from the two previous lemmas.

Relations between Important Complexity Classes

Theorem 1. *Let $f(n)$ be a proper function. Then*

(a) $SPACE(f(n)) \subseteq NSPACE(f(n))$ *and* $TIME(f(n)) \subseteq NTIME(f(n))$.

(b) $NTIME(f(n)) \subseteq SPACE(f(n))$

(c) $NSPACE(f(n)) \subseteq TIME(k^{\log n + f(n)})$

Corollary 1. $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE$.

Proof of Theorem

(b) Simulate the NTM that runs in $f(n)$ time using a TM in exponential time.

The computation tree has height $f(n)$, i.e., each simulation can be carried in $f(n)$ space.

But there are exponentially many simulations (a path from root to leaf).

Each simulation can be carried out one-by-one by reusing the same space.

We can keep track of the sequence of choices currently simulated and generate the next in space $O(f(n))$.

Note that a sequence of choices is a $f(n)$-long sequence of integers between 0 and $d-1$ (where d is the degree of nondeterminism).

The first sequence $(0^{f(n)})$ can be generated since f is proper.