

Two "Handwired" Facts:

Pippenger-Fischer 1978 after Herlihy-Stearns 1966.

Theorem: Given any k -tape DTM M we can build a 2-tape DTM M' such that on any input x and for any t , $M'(x)$ simulates t steps of $M(x)$ using $O(t \log t)$ steps of its own. Moreover, the sequences of cells $C_0, C_1, C_2, \dots, C_t$ of each of the two heads of M' depends only on the length $|x|$ of x .

Corollary: We can further build circuits $\{C_n\}_{n=1}^\infty$ of size $O(t(n) \log t(n))$ rather than $O(t(n)^2)$ shown in class and notes such that $\forall x \in \{0,1\}^n, C_n(x) = M(x)$.

Corollary 2: The reduction from any A in $\text{NTIME}(n)$ to 3SAT can be made to run in $O(n \log n)$ time.

Theorem: Let $f(n)$ and $g(n)$ be any functions built up from $+$, \cdot , $-$, $/$, exp and log. Then either $f(n) = o(g(n))$, $f(n) = \Theta(g(n))$, or $g(n) = o(f(n))$. "Trichotomy of O ".

Moreover, if $f(n) \geq nt$ then f is "fully time constructible" and if $g(n) \geq \log(n)$ then g is "fully space constructible".

Even "jags" simulates 1 step of M and doing t jags takes total moves.

G.H. Hardy 1910s "Folklore"

$x \sin(\frac{1}{x})$

Deterministic Time Hierarchy Theorem: Suppose $t_1(n) \log t_1(n) = o(t_2(n))$ with $t_2(n)$ being "fully time constructible". Then $\text{DTIME}[t_1(n)]$ is properly contained in $\text{DTIME}[t_2(n)]$.

DSPACE Hierarchy Theorem: If $S_1(n) = o(S_2(n))$ and $S_2(n)$ is fully space constructible (or ...?), then $\text{DSPACE}(S_1(n)) \subsetneq \text{DSPACE}(S_2(n))$.

Examples:

$\text{DTIME}(O(n)) \subsetneq \text{DTIME}(O(n^2)) \subsetneq \text{DTIME}(O(n^3)) \subsetneq \dots \subsetneq P$ (needs $S_1(n) \geq \log(t_2(n))$)

$\text{DSPACE}(O(\log n)) \subsetneq \text{DSPACE}(O((\log n)^2)) \subsetneq \dots \subsetneq P$ (needs $2^{O((\log n)^2)} = n^{O(\log n)}$ not polynomial)

$L \subsetneq \text{DLBA} \subsetneq \text{DSPACE}(n^2)$

Note: We still don't know whether $L \subsetneq P$ (or even NP!)

Jags

Even "jag" simulates 1 step of M and doing t jags takes $O(t \log t)$ total moves.

Proof - We design a DTM M'' that is clocked to run in time $t_2(n)$, so $L(M'') \in \text{DTIME}(t_2(n))$
 such that $L(M'')$ differs on some string x from $L(M)$ for every M that runs in time $t_1(n)$.

M'' :
 input x $n=|x|$
 Initialize a tape to $t_2(n)$
 fully time constructible means this can be done within $O(t_2(n))$ steps.
 Attempt to decode $x =: M' \# y$ where M' is recognizable a 2-tape TM obtained from some M by the Theorem.
 on success if fail, reject x .

Code of M' , length = same c # $y_1 y_2 \dots$ length = same d independent of c
 clock $t_2(n)$ ← clock register which "ticks down"
 = the two tapes of M' .
 Tapes of M' derived from M by the first Theorem.
 The time overhead for M'' to simulate each step by M' is just the factor of c needed to find and execute the next instruction.

Enter a simulation of M' on input $x = M' \# y$, which is a padded version of the own code of M' , padded by y . Meanwhile, decrement the clock at each step of M'' .
 • If and when the clock goes to 0, reject.
 • Else, this means $M'(x)$ finishes under your simulation before the clock goes to 0. If M' accepts x , reject x . If M' rejects x , accept x .

Hardy 910s
 "Klone"

forced to run in time $t_2(n)$ by the clock, so $L(M'') \in \text{DTIME}(t_2(n))$.

The cut-off part at left is the same as before and does not matter here. Next adds the end of the proof:

M'' that is clocked to run in time $t_2(n)$, so $L(M'') \in \text{DTIME}(t_2(n))$
 differs on some string x from $L(M)$ for every M that runs in time $t_1(n)$.

Code of M' , length = same c # $y_1 y_2 \dots$ length = same d independent of c
 clock $t_2(n)$ ← clock register which "ticks down"
 = the two tapes of M' .
 Tapes of M' derived from M by the first Theorem.
 The time overhead for M'' to simulate each step by M' is just the factor of c needed to find and execute the next instruction.

simulation of M' on input $x = M' \# y$, padded version of the own code of M' , padded by y . Meanwhile, decrement the clock at each step of M'' .
 • when the clock goes to 0, reject.
 • means $M'(x)$ finishes under your simulation before the clock goes to 0. If M' accepts x , reject x . If M' rejects x , accept x .

forced to run in time $t_2(n)$ by the clock, so $L(M'') \in \text{DTIME}(t_2(n))$.

Let any $A \in \text{DTIME}(t_1(n))$ be given.
 Take M to accept A in time $t_1(n)$.
 Take M' to accept A in time $O(t_1(n) \log t_1(n))$ by the Theorem.
 By $t_1(n) \log t_1(n) = o(t_2(n))$,
 for any c there is a d such that for any $n \geq c+d$,
 $O(t_1(n) \log t_1(n)) \leq t_2(n)$.
 Hence with $x = \langle M \rangle \# 0^d$ the simulation of $M'(x)$ finishes before the clock rings and gives a different answer.
 Thus $M''(x) \neq M'(x) = M(x) = A(x)$. Note: We
 So $L(M'') \neq A$.

Examples:
 $\text{DTIME}(O(n))$
 $\text{DTIME}(O(\log n))$