Undergraduate Complexity Theory
Lecture 5: The Time Hierarchy Theorem

Marcythm

July 10, 2022

1 Lecture Notes

1. A language not in P? $\mathsf{HALTING}$.
2. A <u>decidable</u> language not in P?
   idea: task. simulate a TM for 2^n steps
   (a) should be doable in poly(2^n) time
   (b) shouldn't be doable in $poly(n)$ time
3. extended task: not doable in $O(n^2)$ time, but doable in …say $O(n^8)$ time.
   idea: simulating a TM for n^3 steps

Definition 1.1. Define a TM D: on input $\langle M, w\rangle$, use a universal TM to simulate $M(\langle M, w\rangle)$ for n^3 steps. If it accepts, then D rejects. Conversely, if it rejects or times out, D accepts.

Fact 1.2. *D is a decider, therefore D defines a language.*

Fact 1.3. *D runs in poly time. (depends on efficiency of UTM, maybe $O(|\langle M\rangle|^4 \cdot n^3 \cdot \log n) \le O(n^8)$.)*

Fact 1.4. *Let L be the language decided by D, therefore $L \in \mathsf{TIME}(n^8)$.*

Claim 1.5. *$L \notin \mathsf{TIME}(n^2)$.*

Proof. AFSOC (Assume for the sake of contradiction) that S is a decider TM for L with time $t(n)$, which is $O(n^2)$. Consider running S on inputs of form $\langle S, 0^l\rangle$. Let $c = \langle S\rangle$. S halts on such inputs in $\le t(l + c)$ time, which is $O((l + c)^2)$, strictly less than $(l + c)^3$ for large enough l, say $l \ge l^*$. Since S decides L, $S(\langle S, 0^{l^*}\rangle) = D(\langle S, 0^{l^*}\rangle)$, where the things D does is using UTM to simulate $S(\langle S, 0^{l^*}\rangle)$ for $(l^* + c)^3$ steps, then it does the opposite. Here S doesn't timeout, then $S(\langle S, 0^{l^*}\rangle) = D(\langle S, 0^{l^*}\rangle) = \neg S(\langle S, 0^{l^*}\rangle)$. □

Definition 1.6. Bounded Halt/Accepts problem:

$$BA_{n^3} = \{\langle M, w\rangle : M(w) \text{ accepts w/i} \le |w|^3 \text{ steps}\}$$

Claim 1.7. *$BA_{n^3} \in \mathsf{TIME}(n^8)$.*

Claim 1.8. *$BA_{n^3} \notin \mathsf{TIME}(n^2)$.*

Proof. AFSOC, B is a TM deciding BA_{n^3} in $O(n^2)$ time, we'll show $L \in \mathsf{TIME}(n^2)$. To decide L on input $\langle M, w\rangle$, we need to check if $M(\langle M, w\rangle)$ accs w/i n^3 steps. Run B on the string $\langle M, \langle M, w\rangle\rangle$ (which can be prepared in $O(n^2)$), and do opposite, which also takes $O(|\langle M, \langle M, w\rangle\rangle|^2) = O(n^2)$ time. □

UTM with a clock: given $\langle M, w\rangle$, simulate $M(w)$ for $t(|w|)$ steps. For $U_t(\langle M, w\rangle)$:

1. Count $|w|$: i.e. get $n = |w|$ written on tape.
2. Compute $T = t(n)$ and write it on tape.

3. Sim $M(w)$, decrement T at each step till finishes or $T = 0$.

Time analysis:

1. Doable in $O(n^2)$ time, also in fact in $O(n \log n)$ time. (Bring the counter together with the pointer, n moves with each time $\log n$ symbols to shift, and n symbols to shift when the length increments, but only $\log n$ times).
2. Depends on $t(n)$! $t(n)$ could be uncomputable. If $t(n)$ is simple, maybe polylog(n) time.

Definition 1.9. A function $t : \mathbb{N} \to \mathbb{N}$ is "time constructible" if:

1. $t(n) \geq n \log n$
2. given a length-n string, can compute $t(n)$ in time $O(t(n))$.
3. Per 1 step of sim, takes $O(|\langle M \rangle|^3)$ time.

Remark 1.10. All "normal" functions $\geq n \log n$ are "time-constructible". e.g. $n \log n, n^2, n^{3.5}, 2^n, \ldots$

Deficiency:

1. Gotta handle the clock: keep the clock with M's state, thus additional $\log t(n)$ bits slowdown per sim step.
2. Only handles alphabets $\{0, 1, b\}$: encode with $\log |\Sigma|$ bits, thus a constant factor slowdown.

Theorem 1.11. *If $t(n)$ is time-constructible, exists TM T_t s.t. given $\langle M, w \rangle$, U_t sims $M(w)$ for $t(|w|)$ steps in time $O(|\langle M \rangle|^4 t(n) \log t(n))$.*

2 Reading

2.1 Sipser 9.1 (Hierarchy Theorems)

1. space constructible
2. space hierarchy theorem and corollaries
3. time constructible
4. time hierarchy theorem and corollaries
5. EQ_{REX} is $\mathsf{EXPSPACE}$-complete (TODO)