Undergraduate Complexity Theory
Lecture 4: Time Complexity and Universal Turing Machines

Marcythm

July 9, 2022

1 Lecture Notes

Recap: simulate 3-tape TM with 1-tape TM. 3-tape TM M_3 use $T(n)$ time and at most $T(n)$ symbols, then 1-tape TM M_1 use $3T(n) = O(T(n))$ symbols, and thus $O(T(n))$ time to simulate each step of M_3, so total time is $O(T(n))T(n) = O(T(n)^2)$.

Theorem 1.1 (Hennie '65)**.** *Any 1-tape TM solving* $\mathsf{PALINDROME}$ *needs* $\Omega(n^2)$ *time.*

Definition 1.2. Let $t : \mathbb{N} \to \mathbb{R}^+$, e.g. $t(n) = n^2$, define

$$\mathsf{TIME}(t(n)) = \{L : \text{exists a TM deciding language } L \text{ in } O(t(n)) \text{ time}\}$$

e.g. $\mathsf{PALINDROME} \in \mathsf{TIME}(n^2)$

Remark 1.3. complexity class = set of languages.

Remark 1.4. Big O notation is built into the definition of complexity class.

Fact 1.5 (Speedup Theorem)**.** *Say exists a TM deciding language L in $5n^3$ time, then exists ...in n^3 time, also $\frac{1}{100}n^3$, $\frac{1}{1000}n^3$ ...*

Definition 1.6. The complexity class $\mathsf{P} = \bigcup_{k\in\mathbb{N}} \mathsf{TIME}(n^k)$, i.e. all languages decidable in polynomial time by a TM.

Remark 1.7. The definition of P is very robust, since it doesn't depend on the model anymore.

Any problems provably not in P? $\mathsf{HALTING}$ problem: no decider solving this problem.

Definition 1.8.

$$\mathsf{ACCEPTS} = \{\langle M, w\rangle : M \text{ is a TM with input alphabet } \Sigma,\ w \in \Sigma^*,\ M(w) \text{ accepts}\}$$

Fact 1.9. *There is a "Universal Turing Machine" U, that takes as input $\langle M, w\rangle$ and simulates $M(w)$.*

Use **diagonalization method** to prove that $\mathsf{ACCEPTS}$ is not decidable.
The next lecture: Time Hierarchy Theorem

Theorem 1.10 (Time Hierarchy Theorem, Informally)**.** *Given more time, a Turing machine can solve more problems.*

2 Reading

2.1 Sipser 7.2 (The Class P)

1. Exponential time algorithms typically arise when we solve problems by exhaustively searching through a space of solutions, called brute-force search

2. All reasonable deterministic computational models are polynomially equivalent.
3. P is a mathematically robust class: invariant for all models of computation that are polynomially equivalent to the deterministic single-tape TM
4. use DP and CNF to prove every CFL is in P

2.2 Sipser 4.2 (Undecidablity)

1. diagonalization method: prove A_{TM} is not decidable
2. decidable = recognizable + co-recognizable