18.404/6.840 Lecture 12

Last time:

- Self-reproducing machines and The Recursion Theorem
- Applications:
  a) New proof that A_{TM} is undecidable
  b) MIN_{TM} is T-unrecognizable (and so is any infinite subset of MIN_{TM})
  c) True but unprovable statements

Today: (Sipser §7.1)

- Introduction to Complexity Theory
- Complexity classes; the Class P

Intro to Complexity Theory

Computability theory (1930s - 1950s):
Is A decidable?

Complexity theory (1960s - present):
Is A decidable with restricted resources?
(time/memory/…)

Example: Let $A = \{\mathrm{a}^k\mathrm{b}^k \mid k \geq 0\}$.
Q: How many steps are needed to decide A?
Depends on the input.

We give an upper bound for all inputs of length n.
Called "worst-case complexity".

steps to decide $A = \{\mathrm{a}^k\mathrm{b}^k \mid k \geq 0\}$

Theorem: A 1-tape TM M can decide A where, on inputs of length n, M uses at most cn^2 steps, for some fixed constant c.

Terminology: M uses $O(n^2)$ steps.

Proof: $M =$ "On input w

1. Scan input to check if $w \in \mathrm{a}^*\mathrm{b}^*$, *reject* if not.
2. Repeat until all crossed off.
   Scan tape, crossing off one a and one b.
   Reject if only a's or only b's remain.
3. Accept if all crossed off. "

Analysis:

$O(n)$ steps
$+O(n)$ iterations
$\times O(n)$ steps

$O(n) + O(n^2)$ steps
$= O(n^2)$ steps

Check-in 12.1

How much improvement is possible in the bound for this theorem about 1-tape TMs deciding A?

(a) $O(n^2)$ is best possible.

(b) $O(n \log n)$ is possible.

(c) $O(n)$ is possible.

Check-in 12.1

Deciding $A = \{\mathrm{a}^k\mathrm{b}^k \mid k \geq 0\}$ faster

Theorem: A 1-tape TM M can decide A by using $O(n \log n)$ steps.

Proof:

$M =$ "On input w

1. Scan tape to check if $w \in \mathrm{a}^*\mathrm{b}^*$. *Reject* if not.
2. Repeat until all crossed off.
   Scan tape, crossing off every other a and b.
   Reject if even/odd parities disagree.
3. Accept if all crossed off. "

Analysis:

$O(n)$ steps

$+O(\log n)$ iterations

$\times O(n)$ steps

$O(n) + O(n \log n)$ steps

$= O(n \log n)$ steps

	Parities
a's	
b's	

Further improvement? Not possible.

Theorem: A 1-tape TM M cannot decide A by using $o(n \log n)$ steps.

You are not responsible for knowing the proof.

Deciding $A = \{a^k b^k \mid k \geq 0\}$ even faster

Theorem: A multi-tape TM M can decide A using $O(n)$ steps.

$M =$ "On input w
1. Scan input to check if $w \in a^*b^*$, *reject* if not.
2. Copy a's to second tape.
3. Match b's with a's on second tape.
4. *Accept* if match, else *reject*."

Analysis:
$O(n)$ steps
$+O(n)$ steps
$+O(n)$ steps

$= O(n)$ steps

Model Dependence

Number of steps to decide $A = \{\mathrm{a}^k\mathrm{b}^k \mid k \geq 0\}$ depends on the model.

- **1-tape TM:** $O(n \log n)$
- **Multi-tape TM:** $O(n)$

Computability theory: model independence (Church-Turing Thesis)
Therefore model choice doesn't matter. Mathematically nice.

Complexity Theory: model dependence
But dependence is low (polynomial) for reasonable deterministic models.
We will focus on questions that do not depend on the model choice.

So… we will continue to use the 1-tape TM as the basic model for complexity.

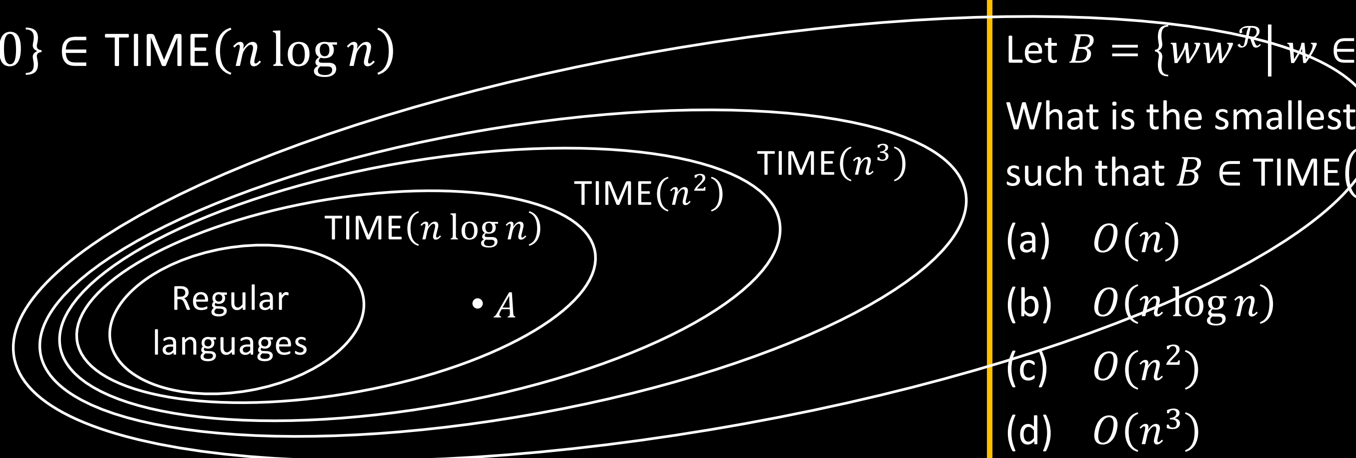
TIME Complexity Classes

Defn: Let $t\colon \mathbb{N} \to \mathbb{N}$. Say TM M runs in time $t(n)$ if M always halts within $t(n)$ steps on all inputs of length n.

Defn: $\text{TIME}\big(t(n)\big) = \{B \mid$ some deterministic 1-tape TM M decides B and M runs in time $O\big(t(n)\big)\}$

Example:

$A = \{\text{a}^k\text{b}^k \mid k \geq 0\} \in \text{TIME}(n \log n)$



Check-in 12.2

Let $B = \{ww^{\mathcal{R}} \mid w \in \{\text{a},\text{b}\}^*\}$.

What is the smallest function t such that $B \in \text{TIME}\big(t(n)\big)$?

(a) $O(n)$

(b) $O(n \log n)$

(c) $O(n^2)$

(d) $O(n^3)$

Check-in 12.2

Multi-tape vs 1-tape time

Theorem: Let $t(n) \geq n$.

If a multi-tape TM decides B in time $t(n)$, then $B \in \text{TIME}\big(t^2(n)\big)$.

Proof: Analyze conversion of multi-tape to 1-tape TMs.

To simulate 1 step of M's computation, S uses $O\big(t(n)\big)$ steps.

So total simulation time is $O\big(t(n) \times t(n)\big) = O\big(t^2(n)\big)$.

Similar results can be shown for other reasonable deterministic models.

Relationships among models

Informal Defn: Two models of computation are polynomially related if each can simulate the other with a polynomial overhead:
So $t(n)$ time $\rightarrow t^k(n)$ time on the other model, for some k.

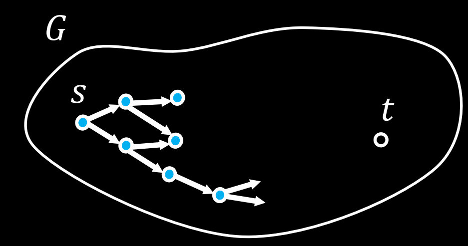
All reasonable deterministic models are polynomially related.

- 1-tape TMs
- multi-tape TMs
- multi-dimensional TMs
- random access machine (RAM)
- cellular automata

The Class P

Defn: P $= \bigcup_k$ TIME(n^k)
$=$ polynomial time decidable languages

- Invariant for all reasonable deterministic models
- Corresponds roughly to realistically solvable problems



Example: $PATH = \{\langle G, s, t\rangle |$ G is a directed graph with a path from s to t $\}$

Theorem: $PATH \in$ P

Proof: $M =$ "On input $\langle G, s, t\rangle$
1. Mark s
2. Repeat until nothing new is marked:
   For each marked node x:
   Scan G to mark all y where (x, y) is an edge
3. *Accept* if t is marked. *Reject* if not.

$\leq n$ iterations
$\times \leq n$ iterations
$\times O(n^2)$ steps

$O(n^4)$ steps

To show polynomial time: Each stage should be clearly polynomial and the total number of steps polynomial.

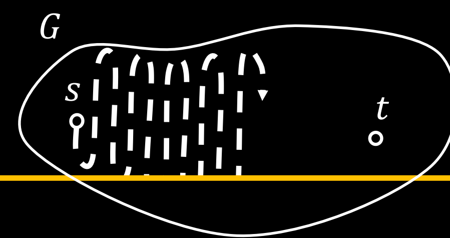
$PATH$ and $HAMPATH$

Example: $HAMPATH = \{\langle G, s, t\rangle |$ G is a directed graph with a path from s to t and the path goes through every node of G $\}$

Called a Hamiltonian path

Recall Theorem: $PATH \in$ P

Question: $HAMPATH \in$ P ?



"On input $\langle G, s, t\rangle$

1. Let m be the number of nodes in G.
2. For each path of length m in G:
   test if m is a Hamiltonian path from s to t.
   Accept if yes.
3. *Reject* if all paths fail."

May be $m! > 2^m$ paths of length m
so algorithm is exponential time
not polynomial time.

Check-in 12.3

Is $HAMPATH \in$ P ?

(a) Definitely Yes. You have a polynomial-time algorithm.
(b) Probably Yes. It should be similar to showing $PATH \in$ P.
(c) Toss up.
(d) Probably No. Hard to beat the exponential algorithm.
(e) Definitely No. You can prove it!

Check-in 12.3

Quick review of today

1. Introduction to Complexity Theory
2. Which model to use? 1-tape-TMs
3. $\text{TIME}\big(t(n)\big)$ complexity classes
4. The class P
5. $PATH$ ∈ P