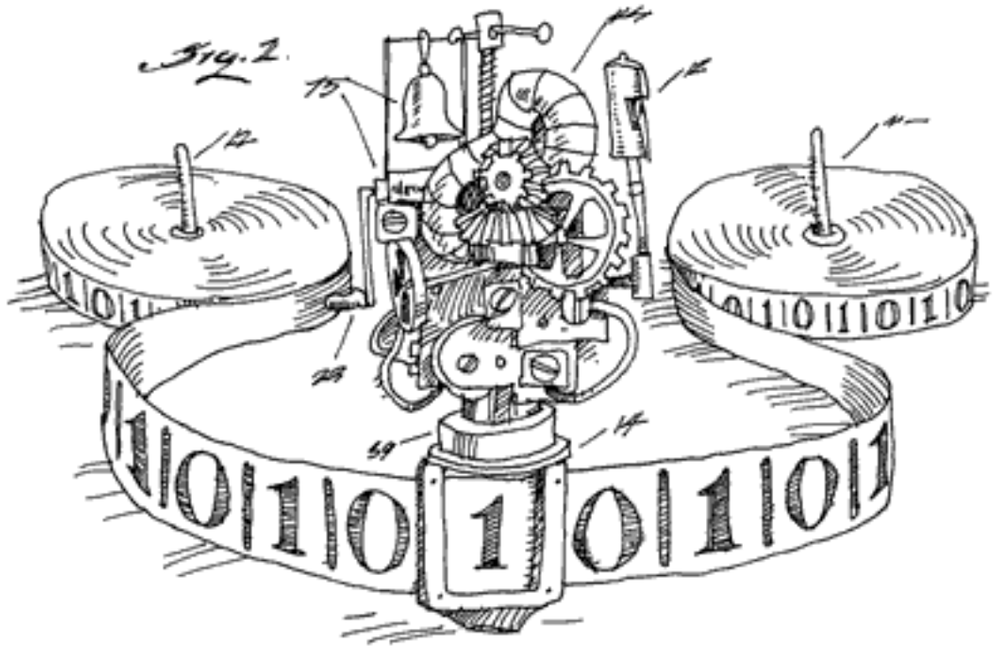
CSC4010: Introduction to Theoretical Computer Science



Lecture 11

Announcements

- Tomorrow: HW4 discussion.

- HW3 solutions are online. (Wording of Q7 was ambiguous, it has been changed.)

- HW2 solutions will be online today.

- Quiz 2 on 27/10 in tutorial session for the first 20 minutes. Format same as Quiz 1, but based on lectures 6-10.

- After Quiz 2, discussion on HW5 (based on lectures 11, 12). No more problem sets after HW5 until midterm.

- Mock midterm will be released on 23rd.

- The midterm is in class (8:30-9:50 am in AB E206) on 2/11

Midterm Instructions

- The midterm is in class (8:30-9:50am in AB E206) on 2/11.

- Worth 30% of your grade (36 points).

- Duration is 1 hour.

- Based on lectures 1-12. (So make sure you do HW5.)

- You are allowed to bring cheatsheet (one-sided, a4-size, handwritten by you).

- No phones allowed on your person! Bring a bag to keep your phone in.

Recap

Deciding a Language in Time T

- Let $Y \subseteq \{0,1\}^*$ and let $T : \mathbb{N} \to [0,\infty)$ be a function.
- **Definition:** We say that Y is can be decided in time T if there exists a one-tape Turing Machine M such that
  - M decides Y,
  - For every $n \in \mathbb{N}$ and $w \in \{0,1\}^n$, the running time of M on w is at most $T(n)$.

The Complexity Class P

- **Definition:** For any function $T : \mathbb{N} \to [0, \infty)$, we define

$$\text{TIME}(T) = \{ Y \subseteq \{0,1\}^* \mid Y \text{ can be decided in time } T \}.$$

- **Definition:**

$$P = \{ Y \subseteq \{0,1\}^* \mid Y \text{ can be decided in time poly}(n) \}$$
$$= \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- "Polynomial Time"

P: Our model of Tractability

- Let $Y \subseteq \{0,1\}^*$
- If $Y \in P$, then Y is “tractable”.
- If $Y \notin P$, then Y is “intractable”.

P: Our model of Tractability

Lemma: If there exists a k-tape Turing machine that decides Y with time complexity $T(n)$, then there exists a 1-tape Turing machine that decides Y with time complexity $O(T(n)^2)$.

Corollary: If a language L can be decided by a multi-tape Turing machine in time $p(n)$ for some polynomial p. Then L can also be decided by a one-tape TM in time $q(n)$ for some polynomial q.



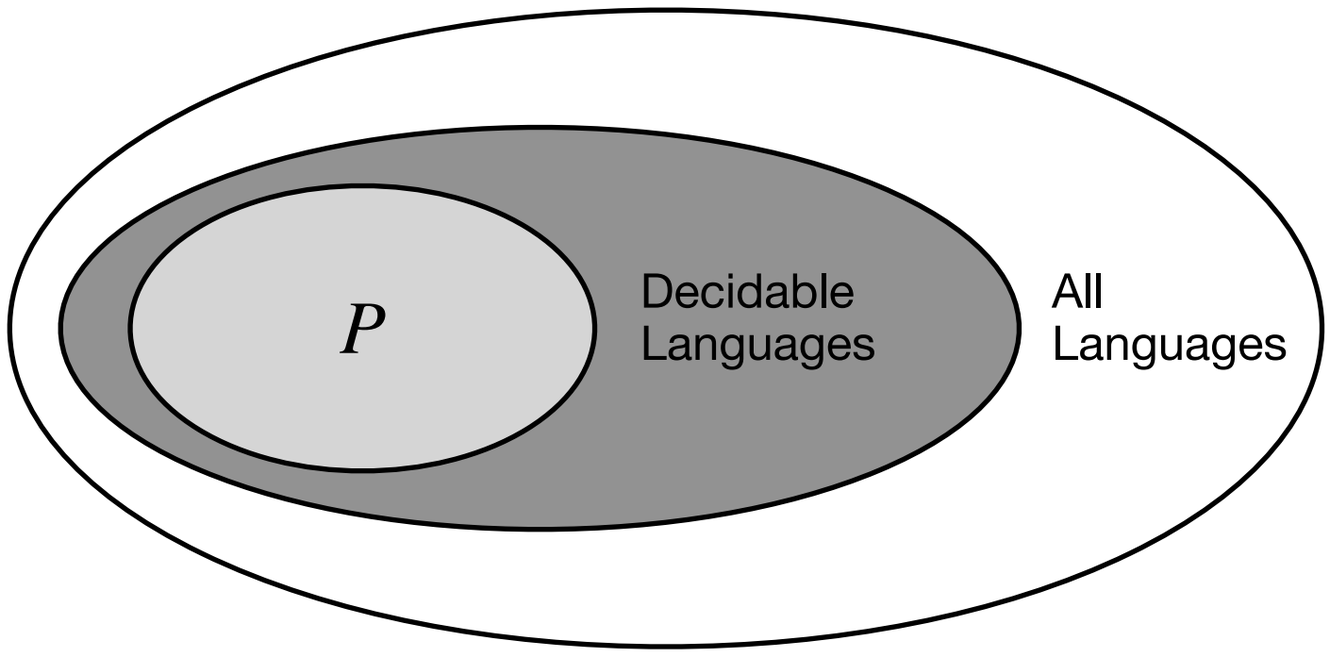
Your laptop: Performs 10^{11} operations/sec

	$n=10$	$n=35$	$n=60$	$n=85$
n^2	100 <1sec	1225 <1sec	3600 <1sec	7225 <1sec
n^3	1000 <1 sec	43K <1sec	216K <1sec	614K <1sec
2^n	1024 <1sec	34×10^9 <1 sec	133 days	12 million years

Today: “Tractability” vs “Decidability”

Tractability vs Decidability

- Perhaps every decidable language is in P?
  - Can every algorithm be modified to make it run in polynomial time?



Tractability vs Decidability

Theorem: There exists $Y \subseteq \{0,1\}^*$ such that Y is decidable, but $Y \notin P$

- **Proof:** Let $Y = \{\langle M \rangle \mid M$ rejects $\langle M \rangle$ within $2^{|\langle M \rangle|}$ steps$\}$. Language Y is decidable.

Proof that $Y \notin P$

$Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } 2^{|\langle M \rangle|} \text{ steps}\}$

- **Proof Strategy:** Diagonalization with a clock.
  - Let R be a Turing Machine that decides. Suppose $T : \mathbb{N} \to \mathbb{N}$ is the time complexity of R.

Proof that $Y \notin P$

$Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } 2^{|\langle M \rangle|} \text{ steps}\}$

- **Proof Strategy:** Diagonalization with a clock.
  - Let R be a Turing Machine that decides. Suppose $T : \mathbb{N} \to \mathbb{N}$ is the time complexity of R.

The Time Hierarchy Theorem

- Using the same proof idea, we can prove a more general theorem.

Time Hierarchy Theorem: For every function $T : \mathbb{N} \to \mathbb{N}$ such that $T(n) \geq n$, there is a language $Y \in \text{TIME}(T^4)$ such that $Y \notin \text{TIME}(o(T))$.

- "TIME$(o(T))$" refers to the set of languages that are decidable in time $o(T)$.

- "Given more time, we can solve more problems."



Stearns and Hartmanis

Proof of The Time Hierarchy Theorem

- Let $Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$.
- We will show:
  - $Y \in \text{TIME}(T^4)$
  - $Y \notin \text{TIME}(o(T))$.

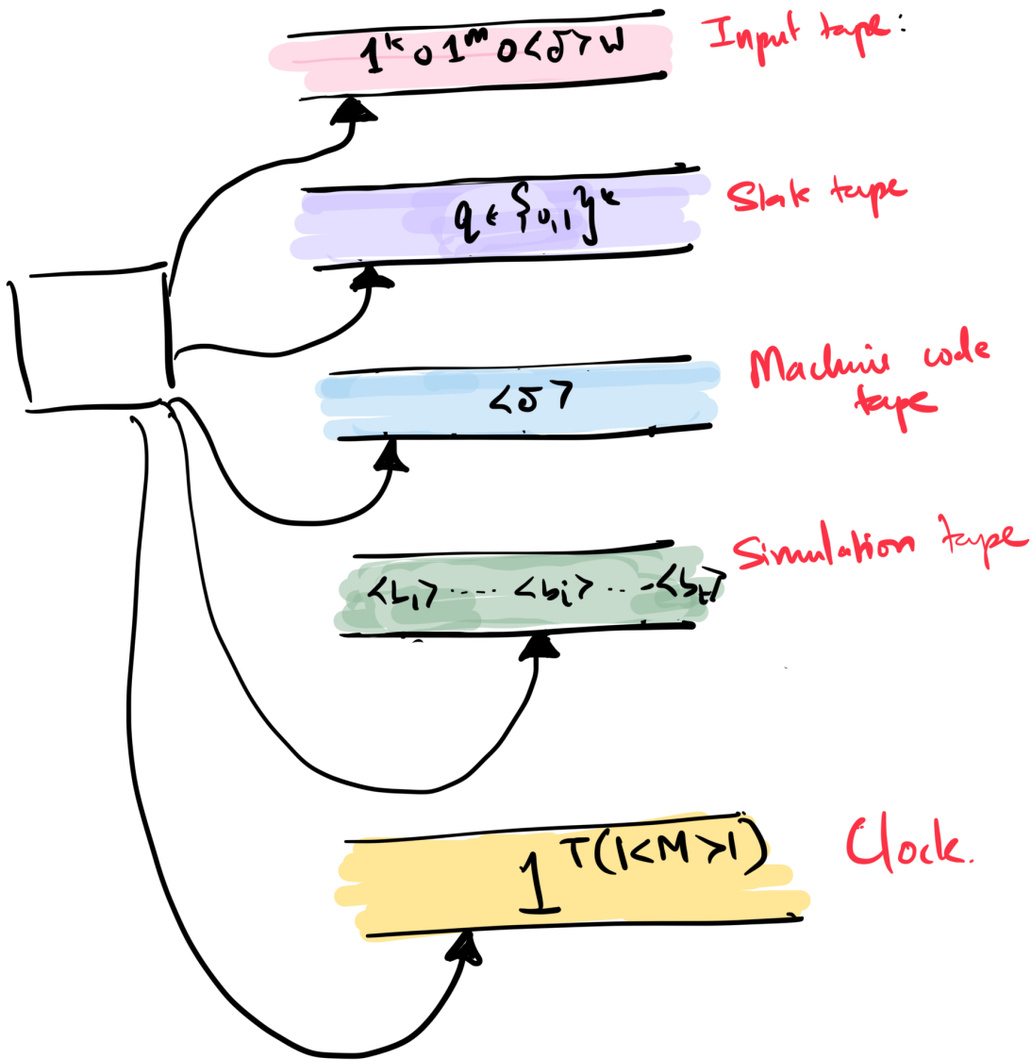
$Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$

Proof that $Y \in \mathbf{TIME}(T^4)$

$Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$

Proof that $Y \in \text{TIME}(T^4)$

- Let $n = |\langle M \rangle|$
- Each simulated step takes $O(n)$ actual steps.
- Total time complexity of multi-tape machine: $O(T(n) \cdot n)$
- After converting to a one-tape machine: $O(T(n)^2 n^2) = O(T(n)^4)$



UTM with a clock

Time Constructible Functions

- **Definition:** A function $T : \mathbb{N} \to \mathbb{N}$ is time-constructible if there exists a Turing machine M such that:
  - Given input 1^n as input, M halts with $1^{T(n)}$ written on tape.
  - M has time complexity $O(T(n))$.

- Our proof that $Y \in \text{TIME}(T^4)$ works assuming T is time-constructible.
- All “reasonable” time complexity bounds are time-constructible.

Proof of The Time Hierarchy Theorem

- Let $Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$.
- We will show:
  - $Y \in \text{TIME}(T^4)$
  - $Y \notin \text{TIME}(o(T))$.

$Y = \{\langle M \rangle \mid M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$

Proof that $Y \notin \textbf{TIME}(o(T))$

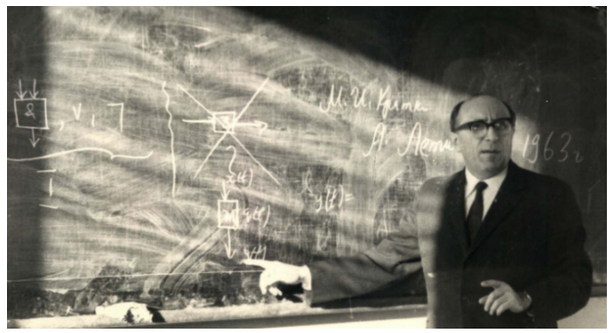
- Let R be a TM that decides Y with time complexity $T' : \mathbb{N} \to \mathbb{N}$.

Thoughts about the proof

Strong Time Hierarchy Theorem: For every pair of functions $f, g : \mathbb{N} \to \mathbb{N}$ where $f \log f = o(g)$ and g is time-constructible, TIME(f) $\subsetneq$ TIME(g).

Thoughts about the proof

Time Hierarchy Gap Theorem: There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $\text{TIME}(f(n)) = \text{TIME}(2^{f(n)})$.



Boris Trakhtenbrot in 1964



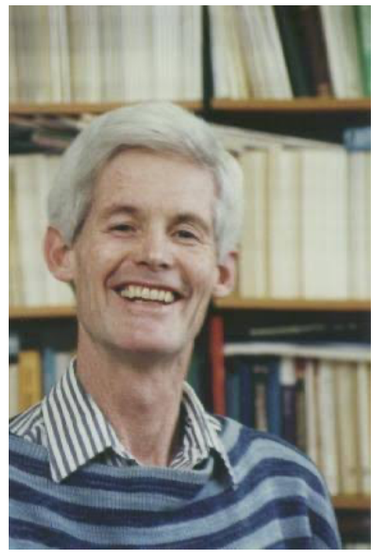
Alan Borodin in 1972

Extended Church-Turing Thesis

Everybody's intuitive notion of efficient algorithms.

=

Polynomial-time algorithms.



Stephen Cook



Robert Reckhow

Objection: "Practical computers are very different from TMs"

Response: RAM model, a theoretical model closer in spirit to the design of contemporary, von Neumann architecture computers.

Controversial thesis! Are n^{100} time algorithms really efficient? What about quantum computing? Randomised algorithms?

Summary

Theorem: There exists a language Y that is decidable but $Y \notin P$.

Strong Time Hierarchy Theorem: For every pair of functions $f, g : \mathbb{N} \to \mathbb{N}$ where $f \log f = o(g)$ and g is time-constructible, TIME(f) $\subsetneq$ TIME(g).

$$\text{TIME}(n) \subsetneq \text{TIME}(n^2) \subsetneq \text{TIME}(n^3) \subsetneq \cdots$$

- There is an infinite hierarchy of increasingly more time-consuming problems.
- **Question:** Are there important everyday problems that are high up in this time hierarchy? A natural problem that needs n^{10} time? This is open!