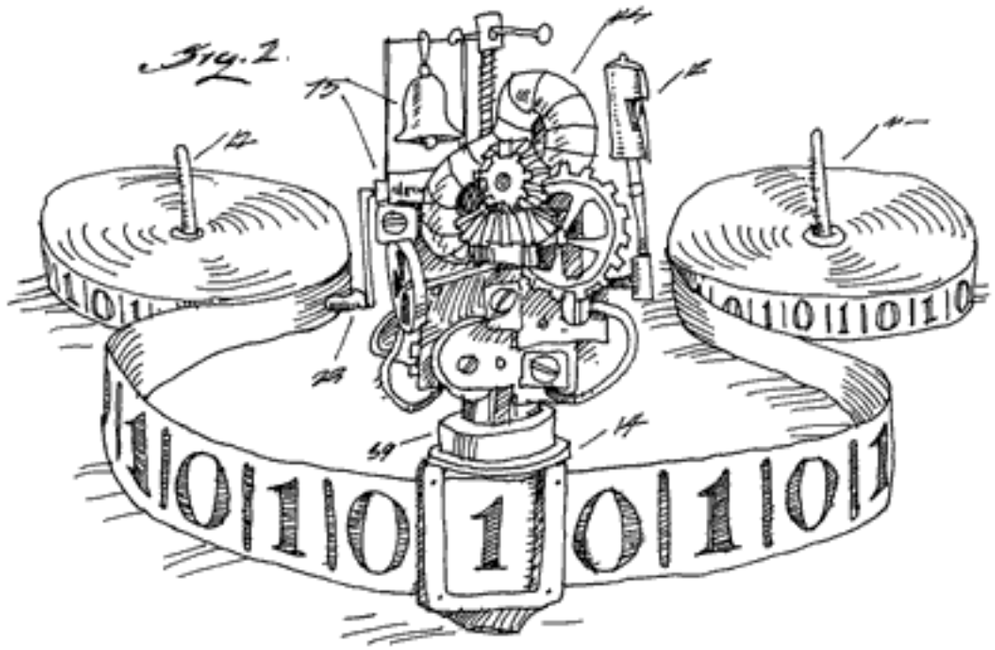
CSC4010: Introduction to Theoretical Computer Science



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 $T(n)$.

$$\text{TIME}(n^c) \subseteq \text{TIME}(n^{c+1})$$

$$c = 0$$

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"

exactly time $\Theta(n^c)$ then P doesn't contain TIME($\log n$), TIME($n^{1.1}$) ...

There is a decider for Y which on i/p any string of length n, halts in time poly(n).

$2^n < n^k$ for some fixed k.

k = 2

$k > n/\log n$.

$k \geq 2$.

k is fixed

$n = k, 2k, \ldots$

2^k	k^k
2^{2k}	2

$\{ 2^n < n^2$ & $n = 3, 4$

2^4	16

P: Our model of Tractability

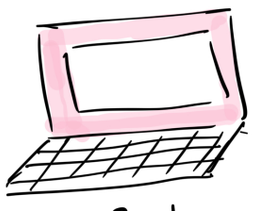
- Let $Y \subseteq \{0,1\}^*$
- If $Y \in P$, then Y is "tractable". / efficiently solvable.
- If $Y \notin P$, then Y is "intractable". no efficient algorithm for Y.

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.

"Robust" to addition of tapes.



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 <1sec	133 days	12 million years

↳ 2^{85}

* Turing machines are quite different from how modern computers work.?

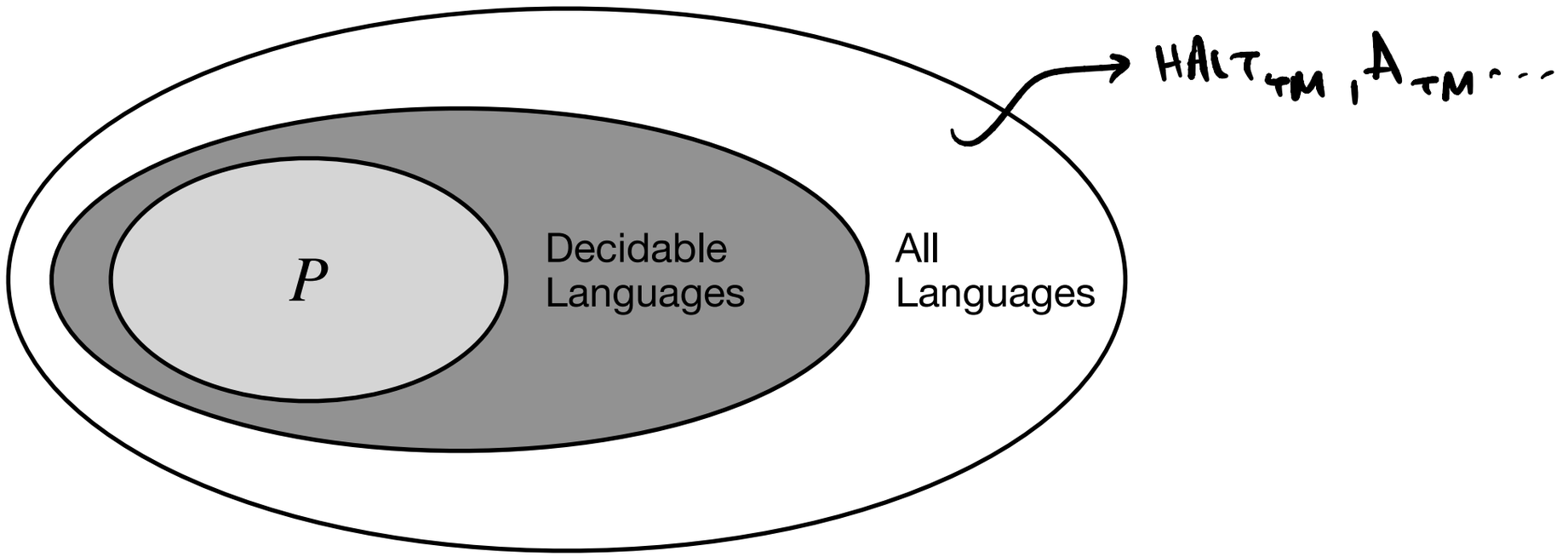
* $TIME(n^{100}) \subseteq P$

↳ $(85)^{100}$

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.

$|\langle M \rangle| = n \Rightarrow 2^n$ steps.

↳ $M(\langle M \rangle)$ then, M halts and rejects within $2^{|\langle M \rangle|}$ steps.

The decider takes as input $\langle M \rangle$

a) Runs $M(\langle M \rangle)$ for $2^{|\langle M \rangle|}$ steps

b) If M accepts then, reject

M rejects then, accept

M hasn't halted, then, reject.

↳ M hasn't rejected $\langle M \rangle$ in $2^{|\langle M \rangle|}$ steps

$\Rightarrow \langle M \rangle \in Y \Rightarrow$ reject.

Proof that $Y \notin P$

for $n = |\langle R \rangle| > 1$, $T(n) > 2^n \cdot n = 100$

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

We haven't shown $T(n) \neq \text{poly}(n)$ ← $2^n < n^{100}$

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

The time complexity of R is T: on strings of length n, R halts in time $T(n)$ and $x \in Y \Leftrightarrow R(x) = 1$.

R takes as i/p $\langle M \rangle$ and accepts it $\Leftrightarrow$ $M(\langle M \rangle)$ rejects within $2^{|\langle M \rangle|}$ steps.

Run R on $\langle R \rangle$ and accept $\Leftrightarrow$ $R(\langle R \rangle)$ rejects within $2^{|\langle R \rangle|}$ steps.

↳ R accepts $\langle x \rangle$ in $2^{|\langle R \rangle|}$ steps then R will reject $\langle R \rangle$.

↳ R rejects $\langle R \rangle$ in $2^{|\langle R \rangle|}$ steps then R will accept $\langle R \rangle$.

↳ R doesn't halt on $\langle x \rangle$ in $2^{|\langle R \rangle|}$ steps.

$T(|\langle R \rangle|) > 2^{|\langle R \rangle|}$.

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. for $n = |\langle R \rangle|$, $T(n) > 2^n$.

a) TM R_n, s.t. $|\langle R_n \rangle| = n$, for $n \geq r$.

b) By adding dummy steps to R.

(i) each $|\langle R_n \rangle| = n$ (ii) R_n decides Y for $n \geq r$.
(iii) Time complexity of each R_n is still T.

a) Run R_n on $\langle R_n \rangle$.
(i) If R_n accepts $\langle R_n \rangle$ in $2^{|\langle R_n \rangle|}$ steps $\Rightarrow$ R_n rejects $\langle R_n \rangle$
(as $\langle R_n \rangle \notin Y$, R_n is a decider for Y)

(ii) If R_n rejects $\langle R_n \rangle$ in $2^{|\langle R_n \rangle|}$ steps $\Rightarrow$ R_n accepts $\langle R_n \rangle$.

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.

$\Rightarrow$ R_n on i/p $\langle R_n \rangle$ should halt in $2^{k*n>1}$ steps.

$T(n) > 2^n$ $\forall n \geq r$.

$n = |\langle R_n \rangle|$.

$\Rightarrow$ $T(n) \neq poly(n)$.

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 $Y \notin \text{TIME}(o(T))$.

$T = n^2$ $\quad Y \in \text{TIME}(n^8)$ $\quad Y \notin \text{TIME}(o(n^2))$

- "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

1-tape TM and 2-tape TM
$T(n)^2$ ⇐ $T(n)$

Proof of The Time Hierarchy Theorem

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

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

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

Decider: on i/p $\langle M \rangle$, run M on $\langle M \rangle$ for $T(|\langle M \rangle|)$ steps.

(i) If M rejects, then accept

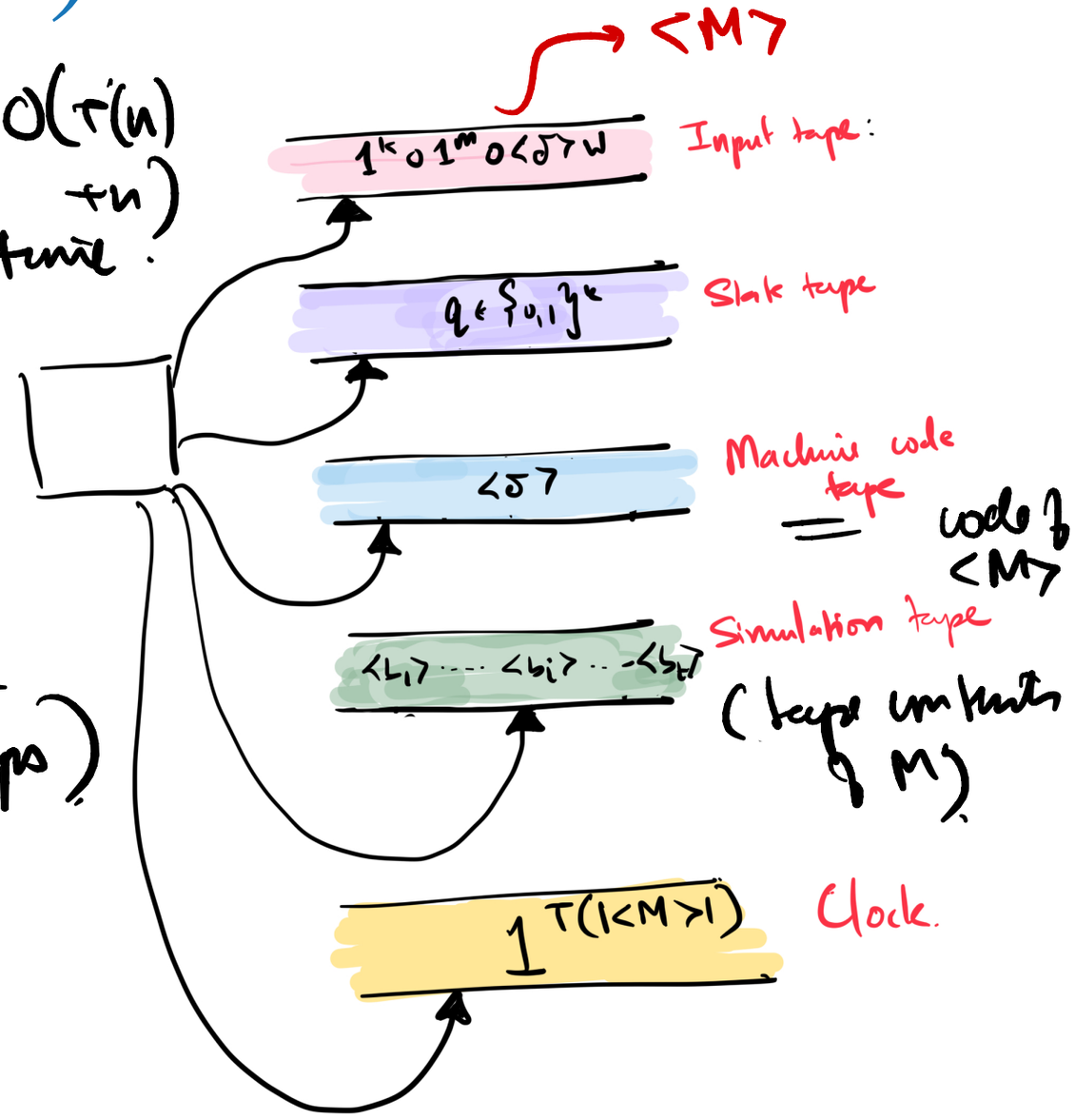
(ii) o/w reject.

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

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: $O(T(n)^2 n^2) = O(T(n)^4)$

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



UTM with a clock

Will simulate M on ⟨M⟩ upto $T(\langle M \rangle)$

$\exists Y \in \text{TIME}(T^4)$, $Y \notin \text{TIME}(o(T))$ then, T hard to Time constructible.

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 $\underline{1^n}$ as input, M halts with $\underline{1^{T(n)}}$ written on tape.
  - M has time complexity $O(T(n))$.

in unary o/p.

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

"polynomial", 2^n ...

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 $Y \notin \text{TIME}(o(T))$

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

Run R on <R>

(a) Within R accepts <R> in time $T(|\langle R \rangle|)$

→ R rejects <R>.

(b) Within R rejects <R> in time $T(|\langle R \rangle|)$

→ R accepts <R>

→ ∴ the only possibility is that R does not halt on <R> in time $T(|\langle R \rangle|)$

→ $T'(|\langle R \rangle|) > T(|\langle R \rangle|)$

→ $Y \neq \text{TIME}(o(T))$ as any decider has complexity $T'(n) > T(n)$ for $n \geq n_0$.

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).

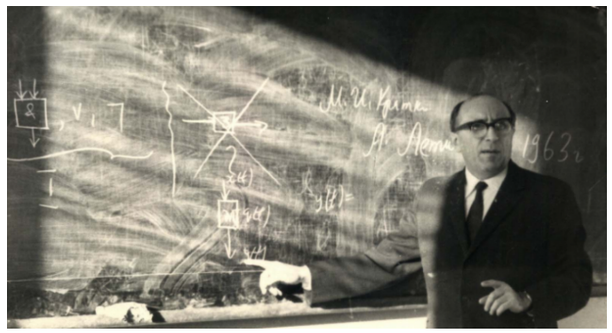
$\text{TIME}(o(T)) \subsetneq \text{TIME}(T^2)$

$\text{TIME}(T) \subsetneq \text{TIME}(T \log T)$

* need T to be time constructible.
* need a better implementation of UTM that has overhead of $\log n$ instead of n.

Thoughts about the proof

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



Boris Trakhtenbrot in 1964



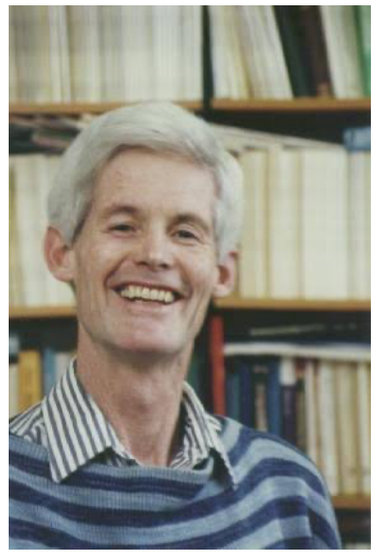
Alan Borodin in 1972

Extended Church-Turing Thesis

Everybody's intuitive notion of efficient algorithms.

=

Polynomial-time algorithms.



Stephen Cook

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.

If RAM takes time $T(n)$ ⇒ 1-tape TM that runs in $T(n)^2$.



Robert Reckhow

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

↘ ∗ fast prime factorization.

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!