

Coms 4236 Intro to Complexity Theory

Lecture 1 (Friday September 11, 2026)

Instructor: Toniann (Toni) Pitassi toni@cs.columbia.edu

TAs: Conlan Olson co2583@columbia.edu

Rain Yang ry2553@columbia.edu

Office hrs: Wed 4-6 (+ more later)

/Course & Grading/

① Check webpage every week - esp lectures link
for readings, notes, info on Hws, test, etc.

② Grading:

- 4 Hws (10 % total)

- 3 tests (in class) Oct 9, Oct 23, Dec 4
(20% each)

- Final Dec 18

Part 1	}	chance to improve scores on Tests 1-3
Part 2		
Part 3		
Part 4	}	40 30%

③ Pre-req: CS theory - TMs, asymptotic notation,
probability, discrete math

see chapter 0 of Sipser

④ See Syllabus for policies:

Collaboration (IMPT), AI, Grading details, missing things

/ Course Intro /

1930s: Computability theory

Formalized computation (general model)

what problems / fns are computable / uncomputable

1970s: What problems are realistically computable given limited resources?

Axes: runtime (steps)

space / memory usage

randomness

approx vs exact solutions

Examples

① Matrix multiplication

INPUT: 2 $n \times n$ matrices w/ integer entries

(assume entries bounded in $[-c, c]$)

$$n \left\{ \underbrace{\begin{bmatrix} A_1 \end{bmatrix}}_n \begin{bmatrix} A_2 \end{bmatrix} \right.$$

OUTPUT: $A_1 \cdot A_2$

Brute force alg: n^3 time (how?)

Surprising: can solve in time $n^{2.4}$

Can we get to n^2 ?

(Why is n^2 a lower bound?)

②

Primality testing

INPUT: x (integer represented in binary)

OUTPUT: 1 if x prime, 0 otherwise

Brute force: $2^{O(|x|)}$ $\leftarrow |x|$: length of binary rep. of x

Surprisingly: Can solve in time $\text{poly}(|x|)$!

③

Factoring

INPUT: x (int. in binary)

OUTPUT: prime factorization of x

Brute force: $2^{O(|x|)}$

Best known: still exponential in $|x|$

(some) Crypto primitives are based on the hardness of factoring

④

Reachability

INPUT: (G, s, t) G is a directed graph $G = (V, E)$
 $s, t \in V$

{ Note: can represent G as $n \times n$ adjacency matrix
 or by list of edges & nodes

OUTPUT: 1 iff $\exists$ path $s \rightsquigarrow t$

Polytime (BFS)

⑤

3-SAT

INPUT: 3 CNF formula

e.g. $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_2 \vee x_4 \vee x_7) \wedge \dots \wedge (\dots)$

OUTPUT: 1 iff $\exists$ assignment $\alpha \in \{0, 1\}^n$ s.t. $f(\alpha) = 1$

Input representation $\leq \binom{n}{\leq 3} \cdot 2^3 = \text{poly}(n)$

Brute force solution: $2^n \cdot \text{poly}(n)$

NP-complete

We will also go beyond runtime complexity to space, randomness
& cover relationships between them

Topics

- I. Time complexity: P , NP , EXP , NP -completeness, time hierarchy thms, randomized time complexity, derandomization
- II. Space complexity: L , NL , $PSPACE$, surprising results! time / space relationships, catalytic space
- III. Interactive proofs, approx. algs, PCP thm
- IV. Lower bounds: circuit complexity, communication complexity, proof complexity

/ TM Basics /

Def A multitape TM is a tuple $(Q, \Gamma, q_0, q_1, \delta)$

where:

$Q = \{q_0, q_1, \dots, q_{|Q|}\}$ is a set of states

Γ is a symbol alphabet containing $\sqcup$

q_0 is the start state

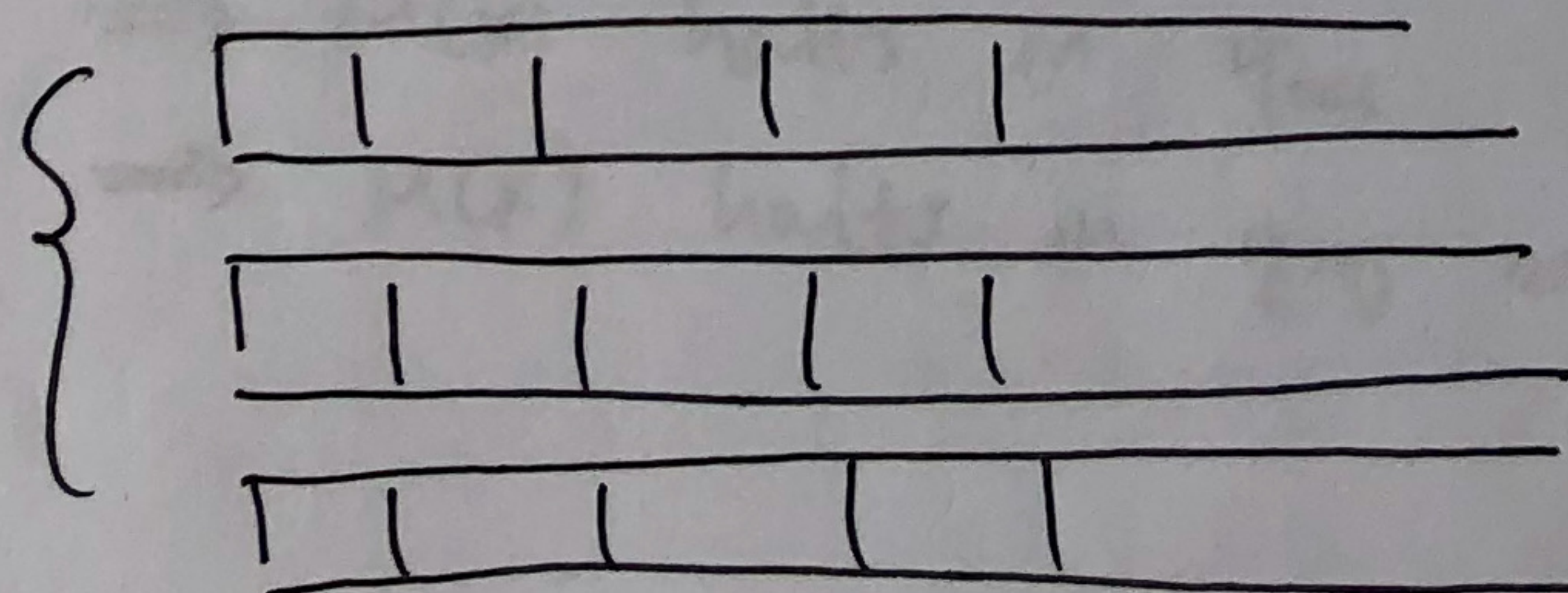
q_1 is the halt state

$\delta: Q \times \Gamma^{k+2} \rightarrow Q \times \Gamma^{k+2} \times \{L, R\}^{k+2}$

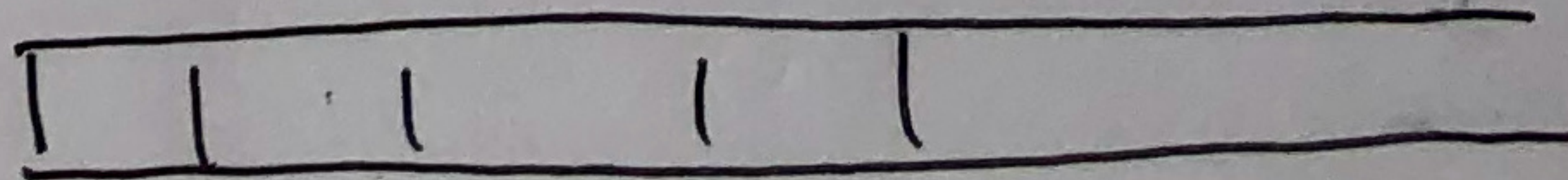
Input tape

x_1	x_2	$\dots$	x_n	$\sqcup$	$\sqcup$	$\sqcup$	$\dots$
-------	-------	---------	-------	----------	----------	----------	---------

k work tapes



Output tape



Start config: Input tape contains input $x_1, \dots, x_n$ followed by all $\perp$
All other tapes contain all $\perp$
All pointers start at the furthest left cell
 q_0 state

Operation: Read all pointers to obtain $\sigma_1, \dots, \sigma_{k+2}$
Let $(q', (\sigma'_2, \dots, \sigma'_{k+2}), (z_1, \dots, z_{k+1})) = \delta(q, (\sigma_1, \dots, \sigma_{k+2}))$
Write $\sigma'_2, \dots, \sigma'_{k+2}$ on the (non-input) tapes
Move each head according to z_i
 q' is the new state

End config: state is q .
Output ~~tape~~ is the output ^{tape} up to first $\perp$

Languages Problems where output is 0 or 1

i.e. $L \subseteq \{0, 1\}^*$
 $\uparrow$ all binary strings

$(L = \{x \mid x \text{ is accepted}\})$

e.g. $L = \{0111x \mid x \in \{0, 1\}^*\}$

Convention for TMs to decide languages: no output tape,

Two halt states q_{accept} and q_{reject}

See Sipser for all formal defs

Def A language $L \subseteq \{0, 1\}^*$ is decided by a TM M iff

- ① $x \in L \Rightarrow M(x)$ halts in q_{acc}
- ② $x \notin L \Rightarrow M(x)$ halts in q_{rej} or never halts

Time Complexity / (sipser ch. 3, A-B ch. 0, 1.1-1.6)

Def A TM M has time complexity $T(n)$ if
 $\forall x \text{ s.t. } |x| = n, M(x) \text{ halts within } \leq T(n) \text{ steps}$

$L \subseteq \{0,1\}^*$ is in $T(n)$ time if $\exists$ multitape TM M
that decides L in time $T(n)$

$TIME(T(n)) := \{L \subseteq \{0,1\}^* \mid L \text{ has time complexity } T(n)\}$

Def $P = \bigcup_{k \geq 0} TIME(n^k)$

"languages decidable in polynomial time"

Exs: Reachability
Matrix multiplication
2-SAT

/Nondeterminism / (sipser 7.3, A-B ch. 2)

Normal TMs always follow one transition

Nondeterministic TMs are a generalization

NTMs are not realizable / realistic but do help us understand
problems without known polytime algs, but easy polytime
verification

Def (informal) A NTM is like a TM but

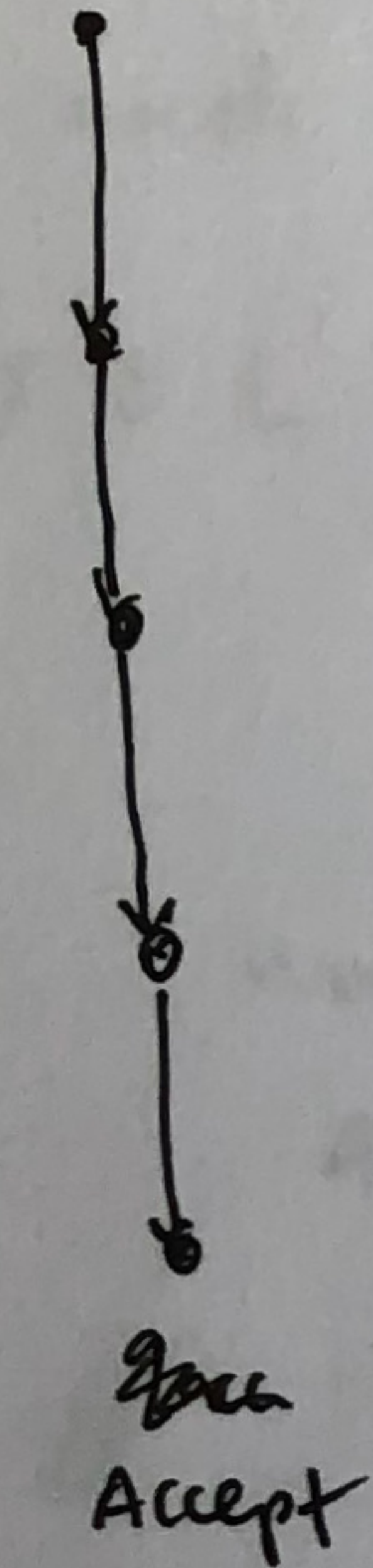
δ maps to a set of configurations:

$$\delta: Q \times \Gamma^{k+2} \rightarrow \mathcal{P}(Q \times \Gamma^{k+1} \times \{L, R\}^{k+2})$$

ex. $\delta(q_2, (a, b, b)) = \left\{ \begin{array}{l} (q_3, (a, c), (LLL)) \\ (q_{rej}, (b, c), (RL, L)) \end{array} \right\}$

TM

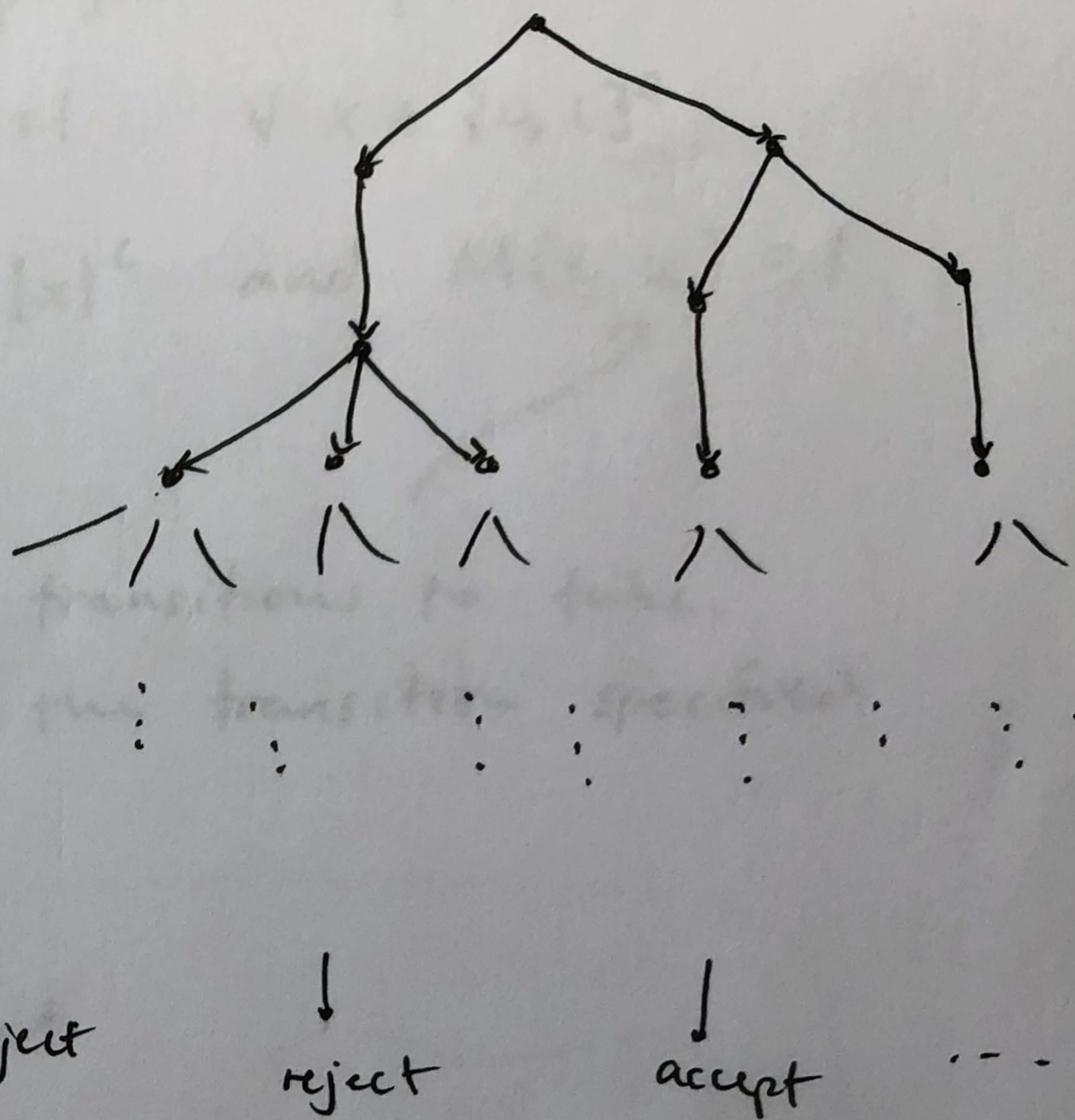
start config



RUNTIME

NTM

start config



Def An NTM decides $L \subseteq \{0,1\}^*$ if:

$x \in L \Rightarrow \exists$ sequence of transitions s.t. $M(x)$ halts & accepts

$x \notin L \Rightarrow \forall$ sequences of transitions, $M(x)$ does not halt & accept

Def An NTM M runs in time $T(n)$ if

$\forall x$ s.t. $|x| = n$, the maximum runtime over all possible computation paths before $M(x)$ halts is $\leq T(n)$

Def $\text{NTIME}(f(n)) = \{ L \subseteq \{0,1\}^* \mid \exists \text{ NTM } M \text{ that decides } L \text{ in time } f(n) \}$

Def 1 $\text{NP} := \bigcup_{k \geq 0} \text{NTIME}(n^k)$

"Verifier Definition" of NP

Def 2 $L \in NP$ iff $\exists$ ~~polynomial~~ polynomial $p(n) = n^c$
and a polytime TM M s.t. $\forall x \in \{0, 1\}^*$
 $x \in L \iff \exists u$ s.t. $|u| \leq |x|^c$ and $M(x, u) = 1$

"witness" : tells ~~NTM~~ TM which transitions to take.

At i^{th} transition, take the transition specified

by u_i

Claim Defs 1 & 2 are equivalent

Fact $P \subseteq NP$

Example 3-SAT : Input 3 CNF formula f over $x_1, \dots, x_n$

witness is $\alpha \in \{0, 1\}^n$ (the assignment).

$V(f, \alpha)$: check if $f(\alpha) = 1$
if yes, accept
o/w, reject

/ Complexity Classes /

Open $P \stackrel{?}{=} NP$ (\$\$\$)

if yes :
- lots of stuff is easy
- replace mathematics
- break a lot of crypto (OWFs)

/NP-Completeness/

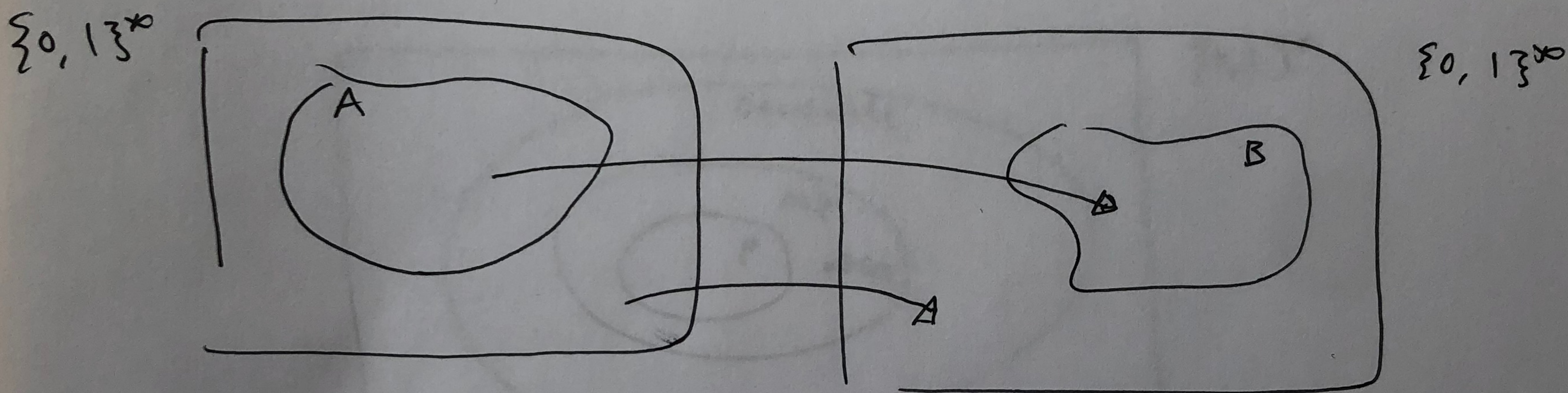
(Sipser 7.4, 7.5, A-B 2.2-2.4)

let $A \subseteq \{0, 1\}^*$, $B \subseteq \{0, 1\}^*$ be languages

Def A is polynomially reducible to B ($A \leq_p B$)

if $\exists$ polynomial $f: \Sigma^* \rightarrow \Sigma^*$

s.t. $\forall x \in \{0, 1\}^*$, $x \in A \iff f(x) \in B$



- i.e.
- ① If we can efficiently solve B then we can efficiently solve A
 - ② If A is hard (not in P) then B is also hard (not in P)

Claim $L_1 \leq_p L_2$ and $L_2 \leq_p L_3 \implies L_1 \leq_p L_3$
(why?)

Def L is complete for a class $\mathcal{C}$ of languages if

① $L \in \mathcal{C}$

② $\forall L' \in \mathcal{C}$, $L' \leq_p L$

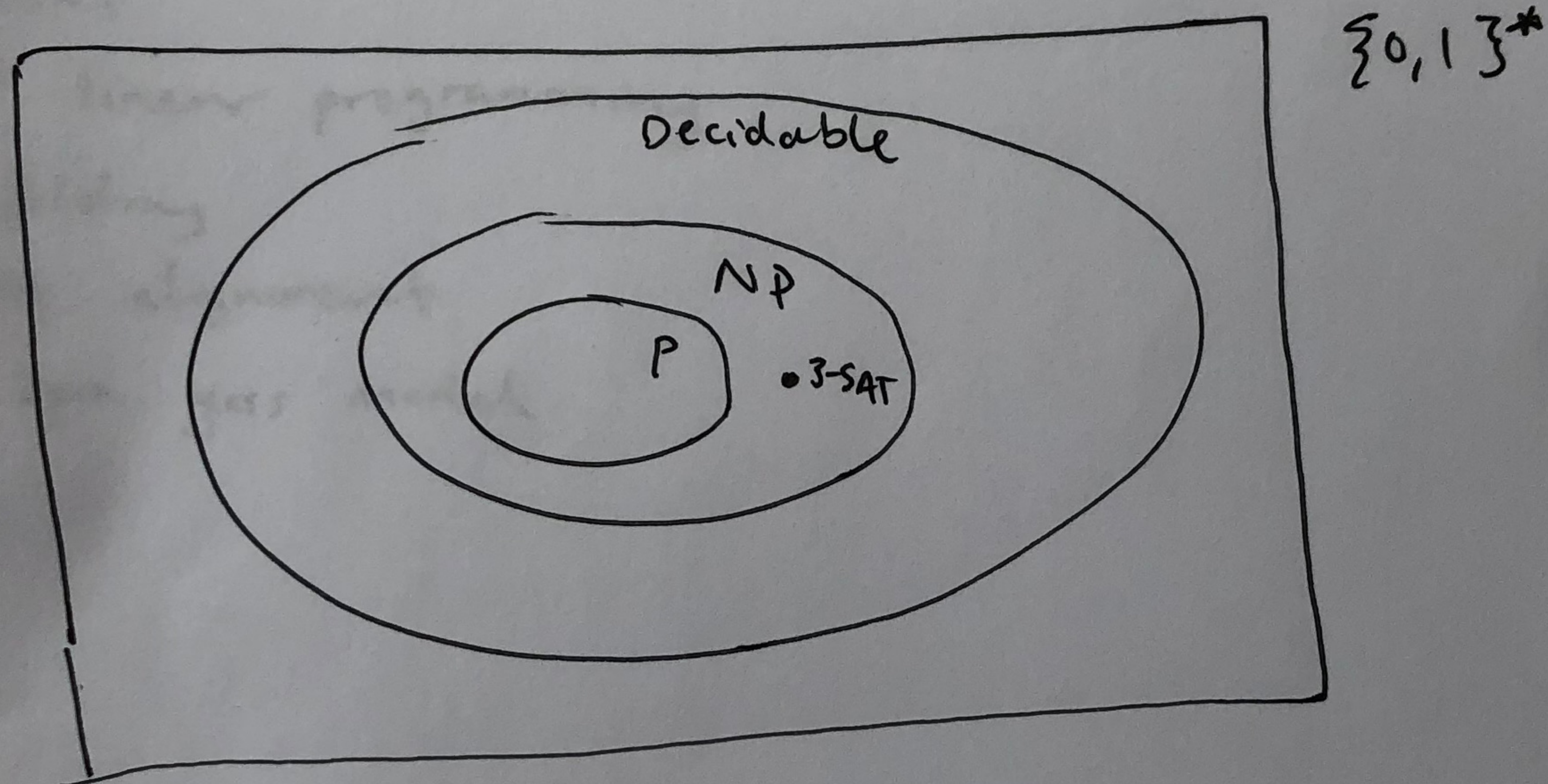
(just ②: $\mathcal{C}$ -hard)

Turing reductions

$L_1 \leq_T^P L_2$ if we can decide L_1 with poly-many calls to L_2

Thm (Cook-Levin)

3-SAT is NP-complete



Note: $3\text{-SAT} \in P \Rightarrow P = NP$ (why?)

Many NP-complete problems. All easy to check $x \in L$ given a "solution" but seemingly hard to find a solution.

Remark All $L \in NP$ are decidable in time 2^{n^c} (why?)

Remark $\exists L$ decidable and $L \notin NP$

Ex. $L = \{ \langle M, x \rangle \mid M \text{ accepts } x \text{ within } 2^{2^{|x|}} \text{ steps} \}$

$$3\text{-UNSAT} = \{ 3 \text{ CNF } f \mid \forall \alpha \in \{0,1\}^n, f(\alpha) = 1 \}$$

More NP-complete problems (Garey - Johnson)

3-SAT

CLIQUE

VERTEX-COVER

INDEP-SET

TSP

Scheduling

Knapsack

Integer linear programming

Protein folding

DNA seq. alignment

Ising spin glass model