ECE 374 B: Algorithms and Models of Computation, Fall 2022

Midterm 3 – December 01, 2022

- **You will have 75 minutes (1.25 hours) to solve 4 problems. Most have multiple parts.** Don't spend too much time on questions you don't understand and focus on answering as much as you can!
- *No* resources are allowed for use during the exam except a multi-page cheatsheet and scratch paper on the back of the exam. ***Do not tear out the cheatsheet or the scratch paper!*** It messes with the auto-scanner.
- You should write your answers *completely* in the space given for the question. We will not grade parts of any answer written outside of the designated space.
- Please bring (sharpened) ***pencils and an eraser*** to take your exam with, unless you are *absolutely sure* you will not need to erase. We will *not* provide any additional scratch paper if you write in pen and make mistakes, nor will we provide pencils and erasers.
- Unless otherwise stated, assume $\mathsf{P} \neq \mathsf{NP}$.
- Assume that whenever the word "reduction" is used, we mean a (not necessarily polynomial-time) *mapping/many-one* reduction.
- ***Don't cheat.*** If we catch you, you will get an F in the course.
- ***Good luck!***

Name: ______________________________

NetID: ______________________________

Date: ______________________________

1 Short Answer I (3 questions) - 20 points

For each of the problems provide a brief and concise solution. These are short answer questions and partial credit will be limited.

(a) (8 POINTS) Consider the problem NEGATIVEHAMCYCLE, which asks if a graph contains a Hamiltonian cycle with negative total weight. Is NEGATIVEHAMCYCLE in NP?

(b) (6 POINTS) Suppose we knew that NEGATIVEHAMCYCLE could *NOT* be decided in polynomial time. Would this imply $P \neq NP$? Why or why not? (You may build off your solution in (a))

(c) (6 POINTS) Consider the problem TRIVIALHAMCYCLE, which asks if all of the edges in a graph form a Hamiltonian cycle. In other words, a YES instance of this problem is a graph (V, E) with $|V| = |E|$ such that E forms a Hamiltonian cycle. This problem is in P since it can be decided in polynomial time. Does this imply $P = NP$? Why or why not?

2 Short Answer II (3 questions) - 20 points

For each of the problems provide a brief and concise solution. These are short answer questions and partial credit will be limited.

(a) (8 POINTS) Is the Halting Problem in NP? Why or why not?

(b) (6 POINTS) Consider the following $2SAT$ expression:

$$(a \vee b) \wedge (\neg d \vee e)$$

Give an equivalent $3SAT$ expression.

(c) (6 POINTS) Consider the following $4SAT$ expression:

$$(a \vee \neg a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c \vee d)$$

Give an equivalent $3SAT$ expression.

3 Classification I (P/NP) - 30 points

Are the following problems in P, NP, or some combinations of complexity classes? For each of the following problems, circle all the complexity classes that problem belongs to. Whatever class it is in, prove it!

(a) Let's define a new problem: WellItsSomething-SAT (you can abbreviate to WIS_{SAT}) defined as follows:

- Input: A 3SAT formula. You may assume variables appear at most once in each clause, and that all clauses are distinct.
- Output: True if there exists a variable assignment that satisfies at least 4 clauses.

Which of the following complexity classes does WellItsSomething-SAT belong to? Circle ***all*** that apply:

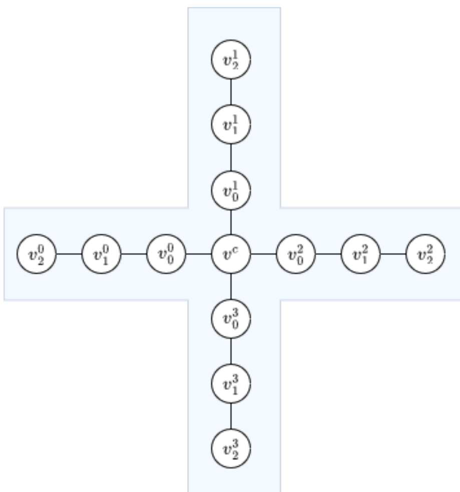
P NP co-NP NP-hard NP-complete

(b) For any integer $k > 1$, a (k)-*plus* is a graph formed by 4 *paths* meeting at a particular vertex , each one on k vertices, plus another vertex linking all the graphs together (i.e., it's a graph with $4k + 1$ vertices). See the figure below for an example of a (3)-plus. The PLUS problem is defined as follows:

- INPUT: An ***undirected*** graph G ***and*** an integer k.
- OUTPUT: TRUE if G contains a (k)-plus as a subgraph, FALSE otherwise.

Which of the following complexity classes does PLUS belong to? Circle ***all*** that apply:

P NP co-NP NP-hard NP-complete



4 Classification II (Decidability) - 30 points

Are the following languages decidable? For each of the following languages,

- Circle one of "decidable" or "undecidable" to indicate your choice.
- If you choose "decidable", prove your choice correct by describing an algorithm that decides that language. If you choose "undecidable", prove your choice correct by giving a reduction proving its correctness.
- Regardless of your choice, explain *briefly* (i.e., in 3 sentences maximum, diagrams, *clear* pseudo-code) why the proof of the choice you gave is valid.

(a)

$$\text{AtleastOne}_{TM} = \{\langle M\rangle \mid M \text{ is a } \textbf{\textit{TM}} \text{ and } L(M) \text{ accepts at least one string}\}$$

decidable   undecidable

(b)

$$\textsc{Empty}_{NFA} = \{\langle A \rangle \mid A \text{ is a } \textbf{\textit{NFA}} \text{ and } L(A) \text{ is empty}\}$$

decidable undecidable

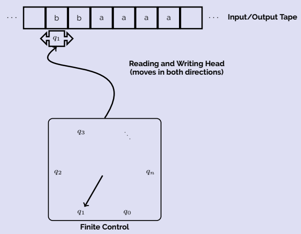
This page is for additional scratch work!

ECE 374 B Reductions, P/NP, and Decidability: Cheatsheet

Turing Machines

Turing machine is the simplest model of computation.

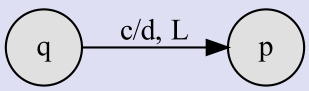
- Input written on (infinite) one sided tape.
- Special blank characters.
- Finite state control (similar to DFA).
- Ever step: Read character under head, write character out, move the head right or left (or stay).
- Every TM **M** can be encoded as a string $\langle M \rangle$



Transition Function: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to, \square\}$

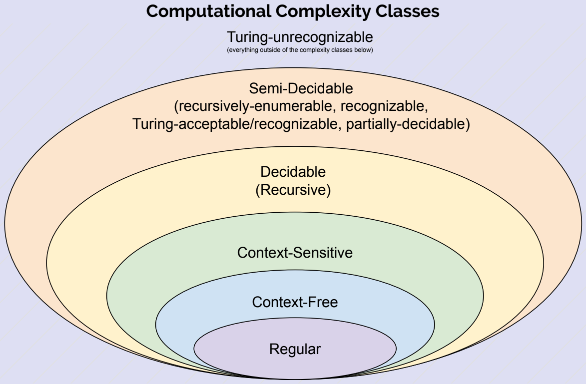
$\delta(q, c) = (p, d, \leftarrow)$

- q: current state.
- c: character under tape head.
- p: new state.
- d: character to write under tape head
- $\leftarrow$: Move tape head left.

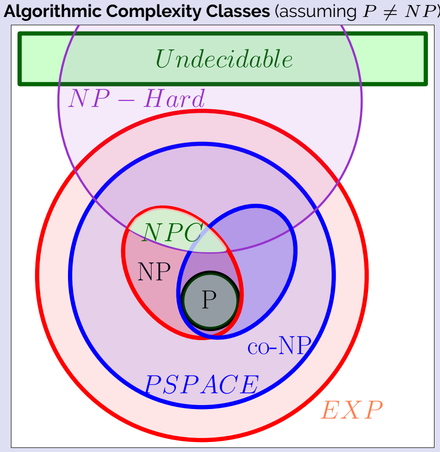


Complexity Classes

Computational Complexity Classes



Algorithmic Complexity Classes (assuming $P \neq NP$)



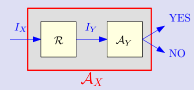
Reductions

A general methodology to prove impossibility results.

- Start with some *known* hard problem X
- *Reduce* X to your favorite problem Y

If Y can be solved then so can $X \implies Y$. But we know X is hard so Y has to be hard too. On the other hand if we know Y is easy, then X has to be easy too.

The Karp reduction, $X \leq_P Y$ suggests that there is a polynomial time reduction from X to Y.



Assuming

- $R(n)$: running time of $\mathcal{R}$
- $Q(n)$: running time of $\mathcal{A}_Y$

Running time of $\mathcal{A}_X$ is $O(Q(R(n))$

Sample NP-complete problems

CircuitSat: Given a boolean circuit, are there any input values that make the circuit output True?

3Sat: Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?

IndependentSet: Given an undirected graph G and integer k, what is there a subset of vertices $\geq k$ in G that have no edges among them?

Clique: Given an undirected graph G and integer k, is there a complete complete subgraph of G with more than k vertices?

KPartition: Given a set X of kn positive integers and an integer k, can X be partitioned into n, k-element subsets, all with the same sum?

3Color: Given an undirected graph G, can its vertices be colored with three colors, so that every edge touches vertices with two different colors?

HamiltonianPath: Given graph G (either directed or undirected), is there a path in G that visits every vertex exactly once?

HamiltonianCycle: Given a graph G (either directed or undirected), is there a cycle in G that visits every vertex exactly once?

LongestPath: Given a graph G (either directed or undirected, possibly with weighted edges) and an integer k, does G have a path $\geq k$ length?

- Remember a **path** is a sequence of distinct vertices $[v_1, v_2, \ldots v_k]$ such that an edge exists between any two consecutive vertices in the sequence. A **cycle** is the same with the addition of a edge $(v_k, v_1) \in E$. A **walk** is a path except the vertices can be repeated.
- A formula is in conjunction normal form if variables are or'ed together inside a clause and then clauses are and'ed together: $((x_1 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee x_4 \vee x_5))$. Disjunctive normal form is the opposite $((x_1 \wedge x_2 \wedge x_3) \vee (\overline{x_2} \wedge x_4 \wedge x_5))$.

Sample undecidable problems

AcceptOnInput: $A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts on } w\}$

HaltsOnInput: $Halt_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and halts on input } w\}$

HaltOnBlank: $HaltB_{TM} = \{\langle M\rangle \mid M \text{ is a TM \& } M \text{ halts on blank input}\}$

Emptiness: $E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \varnothing\}$

Equality: $EQ_{TM} = \left\{\langle M_A, M_B\rangle \,\middle|\, \begin{array}{l} M_A \text{ and } M_B \text{ are TM's} \\ \text{and } L(M_A) = L(M_B) \end{array}\right\}$