ECE 374 B: Algorithms and Models of Computation, Spring 2023

Midterm 3 – April 27, 2023

- **You will have 75 minutes (1.25 hours) to solve 5 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 (2 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. Also assume $P \neq NP$.

(a) (12 POINTS) Assuming the reductions below can be proven,circle all the classes (6 choices) that the problem X *may* belong to:

- $3SAT \leq_P X$

P NP NP-hard NP-complete

decidable undecidable

- $X \leq_P CLIQUE$

P NP NP-hard NP-complete

decidable undecidable

- $X \implies A_{TM}$

P NP NP-hard NP-complete

decidable undecidable

(b) (8 POINTS) *Briefly* describe a reduction that shows:

$$SAT \implies A_{TM}$$

2 Classification I (P/NP) - 20 points

Is the following problem 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!

General shortest-**simple**-path (GSP) problem. Given a graph G, assuming every edge can be taken only once (recall that, that's what simplicity means, every edge can only be used once), **does there exist a path from s to t that is less than k length.** The graph **may have negative cycles**, but that doesn't mean there isn't a shortest **simple** path because every edge can only be taken once.

- INPUT: A graph G and vertices s, t, and integer k.
- OUTPUT: TRUE if there exists a simple path $\leq k$ length. FALSE otherwise.

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

P NP NP-hard NP-complete

3 Classification II(P/NP) - 20 points

Is the following problem 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!

Specialized (DAG) shortest-**simple**-path (SSP) problem. Same problem as before, but this time G is a directed-acyclic graph (DAG). Given a graph G, assuming every edge can be taken only once, **does there exist a path from s to t that is less than k length.**

- INPUT: A DAG G ***and*** an integer k.
- OUTPUT: TRUE if there exists a simple path $\leq k$ length. FALSE otherwise.

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

P NP NP-hard NP-complete

4 Classification I (Decidability) - 20 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.

$$\text{EndWith0}_{DFA} = \{\langle D \rangle \mid D \text{ is a } \textbf{\textit{DFA}} \text{ and all } w \in L(D) \text{ end with the character } \mathtt{0}\}$$

$$\Sigma = \{\mathtt{0}, \mathtt{1}\}$$

decidable undecidable

5 Classification II (Decidability) - 20 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.

$$\text{INF}_{TM} = \{\langle M \rangle \mid M \text{ is a } \textbf{\textit{TM}} \text{ and } |L(M)| = \infty\}$$

$$\Sigma = \{\mathtt{0}, \mathtt{1}\}$$

decidable   undecidable

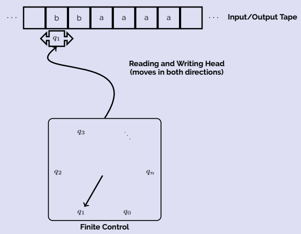
This page is for additional scratch work!

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

Turing Machines

Turing machine is the simplest model of computation.

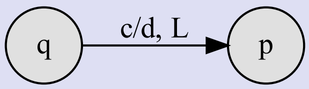
- Input written on (infinite) one sided tape.
- Special blank characters.
- Finite state control (similar to DFA).
- Ever step: Read character under head, write character out, move the head right or left (or stay).
- Every TM **M** can be encoded as a string $\langle M \rangle$



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.



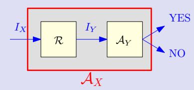
Reductions

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.



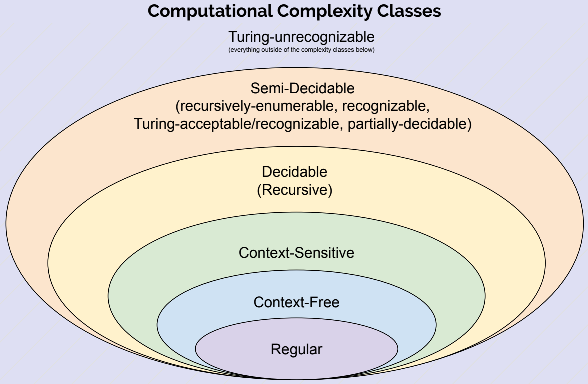
Assuming

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

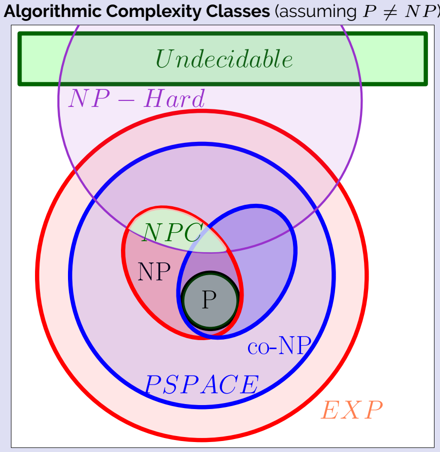
Running time of $\mathcal{A}_X$ is $O(Q(R(n)))$

Complexity Classes

Computational Complexity Classes



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



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