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

Midterm 3 – November 30, 2023

- **You will have 75 minutes (1.25 hours) to solve 7 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 *use a dark-colored pen* unless you are *absolutely* sure your pencil writing is forceful enough to be legible when scanned. We reserve the right to take off points if we have difficulty reading the uploaded document.
- 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.
- You can only refer to the cheat sheet content as a black box.
- ***Don't cheat.*** If we catch you, you will get an F in the course.
- ***Good luck!***

Name: ____________________________

NetID: ____________________________

Date: ____________________________

1 Short Answer I (8 questions) - 16 points

For each of the problems circle *true* if the statement is *always* true, circle *false* otherwise. There is no partial credit for these questions.

(a) If A is a NP-Complete language and B is a NP-Hard language, then $A \leq_P B$.

True False

(b) If A is a NP-Complete language then $A \leq_P SAT$ and $SAT \leq_P A$.

True False

(c) If $A \leq_P B$ and B is NP-Complete, then A is NP-Complete.

True False

(d) If $A \leq_P B$ and A is NP-Complete, then B is NP-Complete.

True False

(e) Every decidable language is in NP.

True False

(f) All NP problems are recursively enumerable.

True False

(g) Every regular language is in P

True False

(h) If A and B are both in NP, then $A \leq_P B$

True False

2 Short Answer II (3 questions) - 9 points

For each of the problems circle all the answers that apply. There is no partial credit for these questions. Points are not necessarily divided evenly among all possible choices.

(a) Assume X is the problem that finds the *hamiltonian cycle of minimum length* given a directed, weighted graph G. Circle the complexity classes this problem belongs to:

P NP NP-hard NP-complete

decidable undecidable

(b) The Tautology problem is the problem of determining if a 3SAT evaluates to true under every possible assignment to its variables. Tautology belongs to:

P NP NP-hard NP-complete

decidable undecidable

(c) Recall the primality problem is problem of determining if a number (n) is prime (has factors $< n$). Primality belongs to:

P NP NP-hard NP-complete

decidable undecidable

3 Classification I (P/NP) - 15 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!

The 374 path problem (374P) asks given an undirected graph G, does G contain a path that visits atleast 374 vertices.

- INPUT: A graph G.
- OUTPUT: TRUE if there exists a path that is atleast 374 vertices long. FALSE otherwise.

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

P NP NP-hard NP-complete

4 Classification II (P/NP) - 15 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!

The multi-solution SAT (**MultiSAT**) problem asks whether a SAT problem has multiple satisfiable truth assignments.

- Input: A SAT formula ϕ.
- Output: True if there exists atleast two distinct variable assignments that satisfy this formula. False otherwise.

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

P NP NP-hard NP-complete

5 Classification III (P/NP) - 15 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!

$HALT_{TM}$ you are given a turing machine $\langle M \rangle$ and must determine if it halts on a empty input.

- INPUT: A TM $\langle M \rangle$.
- OUTPUT: TRUE if the will halt on an empty input. FALSE otherwise.

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

P NP NP-hard NP-complete

You must justify (prove) your answer!

6 Classification I (Decidability) - 15 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.

$$\text{ReachQ}_{TM} = \left\{\langle M, w, q\rangle \mid M \text{ is a } \textbf{\textit{TM}}\left(= \langle Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}\rangle\right) \text{ and will enter state } q \in Q, \text{ on input } w\right\}$$

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

decidable undecidable

7 Classification II (Decidability) - 15 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.

$$\textsc{Accept374}_{TM} = \{\langle M, w\rangle \mid M \text{ is a } \textbf{\textit{TM}} \text{ and accepts } w \text{ in 374 steps. }\}$$

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

decidable undecidable

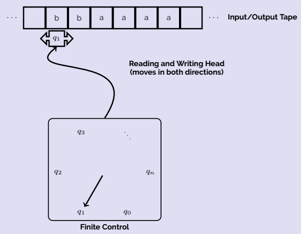
This page is for additional scratch work!

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

Turing Machines

Turing machine is the simplest model of computation.

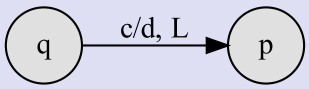
- Input written on (infinite) one sided tape.
- Special blank characters.
- Finite state control (similar to DFA).
- Ever step: Read character under head, write character out, move the head right or left (or stay).
- Every TM **M** can be encoded as a string $\langle M \rangle$



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

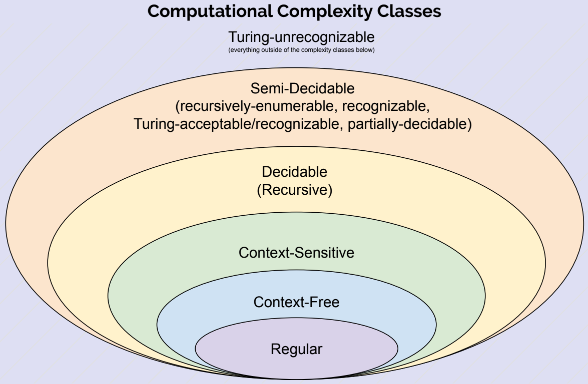
$\delta(q, c) = (p, d, \leftarrow)$

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

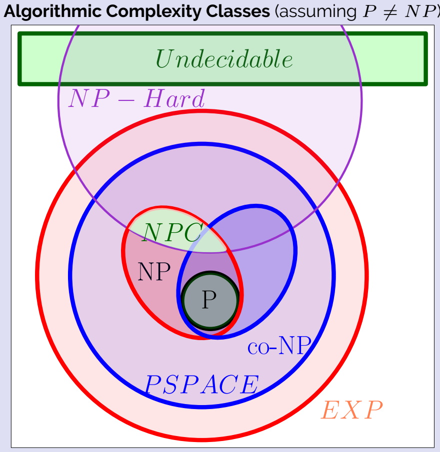


Complexity Classes

Computational Complexity Classes



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



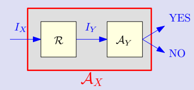
Reductions

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 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:	$A_{TM} = \{\langle M, w \rangle \mid M$ is a TM and M accepts on $w\}$
HaltsOnInput:	$Halt_{TM} = \{\langle M, w \rangle \mid M$ is a TM and halts on input $w\}$
HaltOnBlank:	$HaltB_{TM} = \{\langle M \rangle \mid M$ is a TM & M halts on blank input$\}$
Emptiness:	$E_{TM} = \{\langle M \rangle \mid M$ is a TM and $L(M) = \varnothing\}$
Equality:	$EQ_{TM} = \left\{\langle M_A, M_B \rangle \;\middle|\; M_A \text{ and } M_B \text{ are TM's and } L(M_A) = L(M_B)\right\}$