Instructions:

- Please write your name and username in the space provided.

- You have the entirety of lab time to finish this exam.

- This is a closed book, closed notes exam. However, you may have access to a “reminder” sheet restricted to one side of a US Letter size paper (8.5” × 11”).

- Please turn off all cell phones and electronic devices. Headphones, computers, wireless devices, notebooks, textbooks, and other external objects that are or appear to be capable of communication, computation, or storage must be stowed out of reach during the exam.

- Please write your solutions in the space provided. If you run out of room for an answer, continue on the back of the page.

- Provide a brief description of any machines you construct in your proofs.

- Write legibly! In extreme cases, lack of readability may result in a points penalty.

- Unless otherwise instructed, show enough work to justify your answer.

- When giving proofs, you may use any results (including closure properties, constructions, and theorems) shown in class, in the text, in the practice problems or homework without proving those results.

- You should attempt all problems. Partial solutions will receive partial credit.

- Recall that

\[ A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a Turing Machine that accepts the string } w \} \]

\[ \text{HALT}_{TM} = \{ \langle M, w \rangle \mid M \text{ is a Turing Machine that halts on the input string } w \} \]

- Good luck!
It is ok to use results from:

- lecture notes
- the text
- homework and practice problems and...
- previous problems
[5.23] A is decidable iff $A \leq_m 0^*1^*$

Prove both directions!
Mapping Reducibility

A is decidable $\Rightarrow A \leq_m 0^*1^*$

**Proof:**
- Suppose A is decidable
- Let D be a decider for A
- Need to define computable $f: \Sigma^* \rightarrow \Sigma^*$ st
  $w \in A \iff f(w) \in 0^*1^*$
- Define TM F that computes $f(w)$ as follows:
  - On input $w$
    - Simulate D on $w$
    - If D accepts, output $01$ else output $10$
- IF $w \in A$ THEN $f(w) = 01 \in 0^*1^*$
- IF $w \notin A$ THEN $f(w) = 10 \notin 0^*1^*$
Mapping Reducibility

$A \leq_m 0^*1^* \Rightarrow A$ is decidable

**Proof:**
- Suppose $A \leq_m 0^*1^*$
- Let TM $F$ compute $f: \Sigma^* \rightarrow \Sigma^*$ st
  $w \in A \iff f(w) \in 0^*1^*$
- $0^*1^*$ is a regular $\Rightarrow$ There is a decider $R$ that decides $0^*1^*$
- Define $D$ a decider for $A$ as follows:
  - On input $w$
    - Simulate $F$ on $w$. Call the output $f(w)$.
    - Simulate $R$ on $f(w)$ and do what $R$ does.

- $w \in A \Rightarrow f(w) \in 0^*1^* \Rightarrow R$ accepts $f(w) \Rightarrow D$ accepts $w$
- $w \notin A \Rightarrow f(w) \notin 0^*1^* \Rightarrow R$ rejects $f(w) \Rightarrow D$ rejects $w$
Efficient Algorithms

Show that $E_{DFA} \in P$

We know $PATH \in P$.
Let M be the polytime decider for PATH. Define polytime decider A for $E_{DFA}$ as follows
On input $<D>$:
• Let G be the graphical representation of D.
• For every state f in set of accept states
  • Simulate M on $<G, s, f>$
  • If M accepts, reject
• No path from s to any accept state was found so accept.
NP-completeness

[7.26] $\neg$SAT is NP-complete
NP-completeness

$\neg \text{SAT} = \{<\phi> \mid \phi \text{ is a satisfiable 3CNF formula where no clause has all 3 literals evaluate to True}\}$
\#SAT \in \text{NP}

\#SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 3CNF formula where no clause has all 3 literals evaluate to True}\}
$\#\text{SAT}$ is NP hard

$\#\text{SAT} = \{<\phi> \mid \phi$ is a satisfiable 3CNF formula where no clause has all 3 literals evaluate to True$\}$

**Claim:**
For a given 3CNF formula negation of a satisfying $\neq$-assignment is also a satisfying $\neq$-assignment
$\#\text{SAT}$ is NP hard

$\text{3SAT} \leq_{P} \#\text{SAT}$

**Construction:** Replace each clause 

\[(y_1 \lor y_2 \lor y_3)\]

with 

\[(y_1 \lor y_2 \lor z_i) \land (\bar{z}_i \lor y_3 \lor b)\]
\( \neg \text{SAT} \text{ is NP hard} \)

3SAT \( \leq_P \neg \text{SAT} \)

**Construction:** Replace each clause
\((y_1 \lor y_2 \lor y_3)\)
with
\((y_1 \lor y_2 \lor z_i) \land (\bar{z}_i \lor y_3 \lor b)\)

\(\phi \in \text{3SAT} \implies \phi' \in \neg \text{SAT} \)
¬SAT is NP hard

\[ \phi \in \text{3SAT} \Rightarrow \phi' \notin \text{SAT} \]

**Proof:**

- If \( y_1 \) and \( y_2 \) are False
  - assign \( z_i \) to True.
  - Otherwise assign \( z_i \) to False.
- Assign \( b \) to False

Replace each clause

\[ (y_1 \lor y_2 \lor y_3) \]

with

\[ (y_1 \lor y_2 \lor z_i) \land (\overline{z_i} \lor y_3 \lor b) \]
\#SAT is NP hard

3SAT \leq_p \#SAT

**Idea:** Replace each clause
\((y_1 \lor y_2 \lor y_3)\)
with
\((y_1 \lor y_2 \lor z_i) \land (\bar{Z}_i \lor y_3 \lor b)\)

\[ \phi' \in \#SAT \implies \phi \in 3SAT \]
\[ \phi \in \neg \text{SAT} \Rightarrow \phi \in 3\text{SAT} \]

**Proof:**
- If \( b \) is assigned True
- consider negation of assignment
- In the satisfying \( \neg \)-assignment with \( b \) assigned False, \( y_1, y_2 \) and \( y_3 \) cannot all be assigned False

Replace each clause
\((y_1 \lor y_2 \lor y_3)\)
with
\((y_1 \lor y_2 \lor z_i) \land (\overline{z_i} \lor y_3 \lor b)\)
All languages

Turing recognizable

Turing decidable

CFL

RL
All languages

\[ P \subseteq \text{NP-complete} \subseteq \text{NP-hard} \]

17
Define TM $M_1$ and $M_2$ as follows
\begin{itemize}
  \item $M_1$: On any input loop
  \item $M_2$: On any input accept
\end{itemize}

Define TM $M$ as follows

On input $w$
\begin{itemize}
  \item If $w$ is not a valid 3CNF encoding
  \item then output $\langle M_1, w \rangle$
\end{itemize}

If $w$ is a valid 3CNF encoding
\begin{itemize}
  \item then check if $w$ evaluates to True for any possible assignment to variables in $w$
  \item If yes then output $\langle M_2, w \rangle$ else output $\langle M_1, w \rangle$
\end{itemize}
Define TM $M_1$ and $M_2$ as follows
\[ M_1: \text{On any input loop} \]
\[ M_2: \text{On any input accept} \]

Define TM $M$ that computes $f(w)$ as follows
On input $w$
\begin{enumerate}
\item If $w$ is not a valid 3CNF encoding
then output $\langle M_1, w \rangle$
\item If $w$ is a valid 3CNF encoding
then
\begin{enumerate}
\item check if $w$ evaluates to True for any possible assignment to variables in $w$
\item If yes then output $\langle M_2, w \rangle$ else output $\langle M_1, w \rangle$
\end{enumerate}
\end{enumerate}

$M$ computes $f: \Sigma^* \rightarrow \Sigma^*$ such that
\[ \forall w \in \Sigma^* \quad w \in 3\text{SAT} \iff f(w) \in \text{HALT}_{TM} \]
Define TM $M_1$ and $M_2$ as follows

$M_1$: On any input loop
$M_2$: On any input accept

Define TM $M$ that computes $f(w)$ as follows
On input $w$

If $w$ is not a valid 3CNF encoding
then output $\langle M_1, w \rangle$

If $w$ is a valid 3CNF encoding
then

check if $w$ evaluates to True for any possible assignment to variables in $w$
If yes then output $\langle M_2, w \rangle$ else output $\langle M_1, w \rangle$

This proves that $\text{HALT}_{TM}$ is NP-hard
True or False

If \( \text{SAT} \in \mathbf{P} \) then \( \mathbf{P} = \mathbf{NP} \)

True
True or False

If \( \text{FACTOR} \in \mathbb{P} \) then \( \mathbb{P} = \mathbb{NP} \)

False
It is possible that $3\text{SAT} \in \mathsf{P}$ and $\text{VERTEX-COVER} \notin \mathsf{P}$

False
True or False

It is possible that FACTOR $\in \mathbf{P}$ and SAT $\notin \mathbf{P}$

True
If $A \leq_m \overline{A}$ and $A$ is Turing recognizable then $A$ is decidable.

True