Q1 a) *Proof.* ($\longleftarrow$) Suppose $A \leq_m A_{TM}$ via a reduction map f, we show that A is semi-decidable. Note that we know from lecture that A_{TM} is recognizable by some TM M. Consider the following recognizer M_A for A.
On input w:

- Compute $f(w)$, which can be done because f is computable.
- Run M on $f(w)$, and accept or reject accordingly to M.

Note that M_A recognizes A:

$$\begin{aligned} & w \in \mathcal{L}(M_A) \\ \iff & f(w) \in \mathcal{L}(M) \\ \iff & f(w) \in A_{TM} \\ \iff & w \in A \quad \text{by reduction } f. \end{aligned}$$

Thus, A is semi-decidable. Note in addition that the above proof works if A_{TM} is replaced by any other semi-decidable language. So the following statement is true: if languages $C, D \subseteq \Sigma^*$ are such that $C \leq_m D$, then D being semi-decidable implies C is too.

($\longrightarrow$) Suppose that A is semi-decidable by some TM M. Define $f : \Sigma^* \to \Sigma^*$ by $f(a) = \langle M, a \rangle$. This is obviously computable since we are copying a constant $\langle M \rangle$ and a. This is also a reduction because $a \in A$ if and only if M accepts a if and only if $\langle M, a \rangle \in A_{TM}$. Hence, $A \leq_m A_{TM}$. □

b) *Proof.* Since $A \leq_m A^c$ (say via a reduction f), we know $A^c \leq_m A$ via f as well:

$$a \in A \iff f(a) \in A^c$$

if and only if

$$a \notin A \iff f(a) \notin A^c$$

if and only if

$$a \in A^c \iff f(a) \in A.$$

By result we proved in part a), we can conclude from $A^c \leq_m A$ that A^c is semi-decidable. Because both A and A^c are semi-decidable, A is decidable[1]. □

Q2 Preliminary: A key idea is that given any $A, B \subseteq \Sigma^*$ and a map f on Σ^*, the following two statements are equivalent.

- $s \in B \iff f(s) \in A$ for all $s \in \Sigma^*$;
- $f^{-1}(A) = B$, where $f^{-1} : \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$ is the preimage map.

The second statement is a better notation as it makes clear that given any such A and f, there is exactly one B that can reduce to A via f.

a) *Proof.* Observe that any computable function $f : \Sigma^* \to \Sigma^*$ is a computed by some Turing Machine, and since there are countably many Turing Machine encodings, we can conclude the set of computable functions is countable.

[1]The decider simply combines the recognizers of A and A^c and accept/reject accordingly.

We show that there are at most countably many languages that reduce to A, then, because the set of languages is uncountable, there must be some language B that cannot be reduced to A.

For any language B', B' is reducible to A exactly when there is a computable $f : \Sigma^* \to \Sigma^*$ such that $f^{-1}(A) = B'$. Furthermore, each computable $f : \Sigma^* \to \Sigma^*$ reduces exactly one language, $f^{-1}(A)$, to A. Since there are countably many computable functions, there are at most countably many languages that reduce to A.[2] By the reasoning in second paragraph, we are done. □

b) *Proof.* Our strategy is to prove that there are uncountably many languages that A can reduce to but only countably many languages that reduce to A. So some language C that A reduces to does not reduce to A.

There are only countably many languages that reduce to A: This is proved in part a).

There are uncountably many languages that A reduces to:

First, we will prove the following result.

Lemma. *If $f : \Sigma^* \to \Sigma^*$ is such that $f(A) \cap f(A^c) = \emptyset$,[3] then A reduces to any language C such that $f(A) \subseteq C \subseteq \Sigma^* \setminus f(A^c)$ via f.*

Proof of Lemma. This is a simple check: $a \in A$ implies $f(a) \in C$ since $f(A) \subseteq C$; and $f(a) \in C$ implies $a \in A$ since C does not intersect $f(A^c)$. Thus, $a \in A$ if and only if $f(a) \in C$. So $A \leq_m C$ via f. □

Note the "flexibility" of C in our lemma—C can be as small as $f(A)$ and as large as $\Sigma^* \setminus f(A^c)$. If we choose f so that $\Sigma^* \setminus f(\Sigma^*)$ is infinite, then we have shown there are uncountably many languages A reduces to via f.

Indeed, if $|\Sigma| \geq 2$, define $f : \Sigma^* \to \Sigma^*$ by $f(w) = ws$, where $s \in \Sigma$ is a fixed symbol. We check that $f(A) \cap f(A^c) = \emptyset$: $ws \in f(A) \cap f(A^c)$ implies $w \in A \cap A^c$, which is a contradiction. Furthermore, $\Sigma^* \setminus f(\Sigma^*)$ is the set of strings that do not end in s, which is clearly infinite.

On the other hand, if $\Sigma = \{s\}$, then define f by $w \mapsto ww$. This is clearly injective. So $f(A) \cap f(A^c) = \emptyset$. And $\Sigma^* \setminus f(\Sigma^*)$ is the infinite set of odd length strings.

In either case, we have constructed a desired map f such that $f(A) \cap f(A^c) = \emptyset$ and $\Sigma^* \setminus f(\Sigma^*)$ is infinite. Then, any language in the form of $C' \cup f(A)$, where $C' \subseteq \Sigma^* \setminus f(\Sigma^*)$, is reducible from A. But since $\Sigma^* \setminus f(\Sigma^*)$ is infinite, there are uncountably many choices for C', so A reduces to uncountably many languages. □

[2] What we have done here is constructing a surjective map from the set of computable functions to languages reducible to A.

[3] $f(A)$ denotes the image of A under f

Q3

Theorem. *Every infinite semi-decidable language A has an infinite decidable subset B.*

Proof. Given that A is semi-decidable, we can find an enumerator E that enumerates A. Name the strings, in order they are printed by E, $w_1, w_2, \ldots$ (repetition is allowed). Let $\preceq$ be the lexicographic order on Σ^*. Define $B = \{w_i : w_i \succ w_j, \forall j < i\}$. Intuitively, this is the elements in A that are printed in lexicograhic order by E.

Clearly, $B \subseteq A$. Also, B is infinite because A is infinite and thus for any $w_j \in \mathcal{L}(A)$, some w_i, will be longer than w_j where $i > j$. Furthermore, B is decided by the following TM M.

On input w: run E

- If no previous strings printed is greater than w lexicographically and the current string printed is w, accept.
- If a string greater than w lexicographically is printed, reject.

Our TM obviously terminates because eventually E prints a string greater than w lexicographically. Moreover, $b \in \mathcal{L}(M)$ if and only if b is printed by E and is greater than every string printed before if and only if $b \in B$.

Thus, we have found the desired B! □

Q4 *Proof.*
$\underline{U \in coSD}$. This is true if and only if $U^c \in SD$. Notice that U^c is the set of strings that are not turing machine encodings union the set of TM encodings without a useless state. Let $w_1, w_2, \ldots$ be the list of Σ-strings in lexicographic order. Consider the following recognizer M_{U^c} for U^c.

On input w:

- If w is not a TM encoding, accept;
- If $w = \langle M \rangle$ is a TM encoding, then, for $i = 1, 2, \ldots$, run M for i steps on each of $w_1, \ldots, w_i$. If a state of M is visited during any computation, then it is marked visited (recall that state q_i is encoded as 1^i, we can mark it as visited by changing it to $\bar{1}^i$). If—after running M for i steps on $w_1, \ldots, w_i$—all states are marked, then accept.

This is a recognizer for U^c because $w \in \mathcal{L}(M_{U^c})$ if and only if [w is not a TM encoding or w encodes a TM and all of its states are used], which is what U^c means.

$\underline{U \notin D}$. We will show this by reduction from A_{TM} to U^c. This would show that U^c is undecidable, and therefore U is too. Define a reduction map $f : \Sigma^* \to \Sigma^*$

by

$$\langle M, w\rangle \mapsto \langle M_{\langle M,w\rangle}\rangle,$$
$$a \mapsto \langle M_{reject}{}^4\rangle,$$

where $M_{\langle M,w\rangle}$ is a TM with no useless non-final state that on input y:

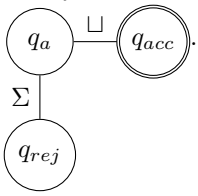
- Runs M on w and halts if and only if M halts on w;
- When M accepts w, it accepts y if and only if y is the empty string;
- When M rejects w, it rejects y.

Indeed, for all $y \in \Sigma^*$,

$y \in A_{TM}$
$\iff y = \langle M, w\rangle$ and M accepts w
$\iff M_{\langle M,w\rangle}$ accepts the empty string but nothing else
$\iff M_{\langle M,w\rangle}$ has no useless state (since $M_{\langle M,w\rangle}$ has no useless non-final states)
$f(y) = f\langle M, w\rangle \in U^c$.

It remains to show that f is a computable function. Given a universal TM W that simulates any $\langle M, w\rangle$ with no unused non-final state (this means every non-final state in W is visited regardless of what it simulates)[5], we can always compute f.

Indeed, we can easily write down $\langle W\rangle$ and load $\langle M, w\rangle$ into W by embedding $\langle M, w\rangle$ into $\langle W\rangle$. The embedding is done by adding states and transitions to $\langle W\rangle$ so that W writes $\langle M, w\rangle$ on its tapes initially. To handle y after M accepts w, we can change the encoding of 

to 

. The case where M rejects w is similar.

Thus, we have shown that $A_{TM} \leq_m U^c$ via f. But A_{TM} being undecidable implies U^c is undecidable too. Decidable languages are closed under complement, so U cannot be decidable. We are done! □

[5]It is not hard to construct such a universal TM since the execution of any TM is just a series of transitions. So it is enough that W parses and executes each transition with no extra state, which is simple to do given the fixed structure of the encoding.