Please print single-sided with each problem on its own pages and your name on every page.

List any collaborators or sources (including yourself) at the end of your submission.

1 Populating the Arithmetic Hierarchy

Prove that for every $n \in \mathbb{N}$, $\textsc{Superhalt}^n$ is m-complete for Σ^0_{n+1} and $\overline{\textsc{Superhalt}^n}$ is m-complete for Π^0_{n+1}, and neither is in Δ^0_{n+1}.

2 The Non-collapse of the Arithmetic Hierarchy

Let $n \in \mathbb{N}^+$. Let A be m-complete for Σ^0_n, A' m-complete for Π^0_n, and $B \in \Delta^0_n$. Prove that A and A' are both not Turing-reducible to B.

3 More About the Arithmetic Hierarchy

(It's a big topic this week.) For languages $A = \{\langle M \rangle \mid M \text{ is a Turing machine with property } P\}$ and B, M^B denotes an oracle Turing machine with an oracle for B and
$A^B = \{\langle M^B \rangle \mid M^B \text{ is an oracle Turing machine (with an oracle for } B\text{) with property } P\}$. For example, $\textsc{Superhalt}$ could be written as $\textsc{Halts}^{\textsc{Halts}}$.

a Prove that $\textsc{Fin}^{A_{TM}}$ is m-hard for Σ^0_3.

Optional: This statement can be generalized. For example, $\textsc{Fin}^{\textsc{Superhalt}^n}$ is m-hard for Σ^0_{n+3}. Please see a staff member (preferably Asa) if you want to explore such generalizations!

b Prove that $\textsc{Cof}$ is m-complete for Σ^0_3. Use the following mapping reduction from $\textsc{Fin}^{A_{TM}}$ to $\textsc{Cof}$. (You will need to prove that it is correct! That will require studying it carefully—it is subtle.) $M^{A_{TM}}$ maps to the following Turing machine:

```
def N(⟨h⟩): #h is a computation history of K^{A_TM}
    def M̃^{A_TM}(s):
        Simulate M^{A_TM} on s, but every time M^{A_TM} receives a YES answer from query_oracle(⟨P,x⟩),
        run P on x and reject if it rejects. Finally, answer as M^{A_TM} on s.
    def K^{A_TM}(s):
        dovetail over x with |x| > |s|:
            if M̃^{A_TM} on x accepts:
                accept
    if h is not a valid and accepting computational history (assuming all its oracle answers are correct):
        accept
    #we can use timesharing rather than dovetailing for the following loop,
    #since there can only be finitely many occurrences
    parallelfor query_oracle(⟨P,x⟩) returning NO in h:
        if P on x accepts:
            accept
    reject
```