CSCI 432/532, Spring 2025

Homework 5

Due Monday, February 24, 2025 at 11:59pm Mountain Time

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF to the appropriate Canvas dropbox.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document.
- When possible, the homework will include at least one fully solved problem, similar to that week's assigned problems, together with the rubric we would use to grade this problem if it appeared in an actual homework or exam. These model solutions show our recommendations for structure, presentation, and level of detail in your homework solutions. (Obviously, the actual *content* of your solutions won't match the model solutions, because your problems are different!)

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you must

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on problem 2. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve problem 1 part (c); "I put my solution for problem 1 part (c) into ChatGPT to check that it was correct and it caught a missing case and suggested some grammar fixes.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading Rubrics

For the undecidability problem:

Undecidability proof rubric. 10 points =

+ 4 for a correct wrapper Turing machine.

+ 6 for a correct proof by contradiction.

Undecidability by reduction.
+ 4 for correct reduction
+ 3 for “if” proof
+ 3 for “only if” proof

1. Consider the language $\textsc{NeverReject} = \{\langle M \rangle : \textsc{Reject}(M) = \emptyset\}$; that is, the set of all (encodings of) Turing machines that do not reject any input. Prove that $\textsc{NeverReject}$ is undecidable.

Solved Problem

1. Consider the language SometimesHalt $= \{\langle M \rangle \mid M \text{ halts on at least one input string}\}$. Note that $\langle M \rangle \in$ SometimesHalt does not imply that M *accepts* any strings; it is enough that M *halts* on (and possibly rejects) some string.

   (a) Prove that SometimesHalt is undecidable.

Solution: We can reduce the standard halting problem to SometimesHalt as follows:

```
DecideHalt(⟨M, w⟩):
  Encode the following Turing machine M′:
    M′(x):
      (ignore x)
      run M on input w
  return DecideSometimesHalt(⟨M′⟩)
```

We prove this reduction correct as follows:

$\Longrightarrow$ Suppose M halts on input w.
Then M' halts on *every* input string x.
So DecideSometimesHalt must accept the encoding $\langle M' \rangle$.
We conclude that DecideHalt correctly accepts the encoding $\langle M, w \rangle$.

$\Longleftarrow$ Suppose M does not halt on input w.
Then M' diverges on *every* input string x.
So DecideSometimesHalt must reject the encoding $\langle M' \rangle$.
We conclude that DecideHalt correctly rejects the encoding $\langle M, w \rangle$.

■