What problems are beyond our capacity to solve? In this last assignment, you’ll explore the limits of computation.

Problem 1

We’ve introduced several terms and definitions related to Turing machines, languages, and what it means to solve a problem. Some of these terms can only describe Turing machines, while others can only describe languages. A statement that uses terms incorrectly has no meaning; it cannot be evaluated as being true or false.

Consider, by analogy, the statement “the set \( \mathbb{N} \) is even”. This statement doesn’t type-check because “even” only applies to individual numbers, and \( \mathbb{N} \) isn’t a number. Similarly, the statement “\( 1 \in 5 \)” doesn’t type-check because 5 isn’t a set.

On the other hand, the statement \( \mathbb{Z} \subseteq \mathbb{N} \) type-checks, but it isn’t true – the integers aren’t a subset of the natural numbers.\(^1\)

For each of the following statements, decide whether

- the statement type-checks and is true,
- the statement type-checks and is false, or
- the statement does not type-check.

Briefly explain your answers.

a. If \( M \) is a Turing machine, \( w \) is a string, and \( M \) accepts \( w \), then \( A_{TM} \) accepts \( \langle M, w \rangle \).

b. If \( M \) is a Turing machine, \( w \) is a string, and \( M \) loops on \( w \), then \( \langle M, w \rangle \notin L(U) \).

---

\(^1\) This is the mathematical equivalent of a grammatically correct sentence that's false, like "Otters aren't cute". You understand what the sentence means, but it's objectively not true. They're adorable.
c. $U$ is decidable.

d. $\langle U \rangle$ is decidable.

e. $\{\langle U \rangle\}$ is decidable.
Problem 2

In class, we saw that $A_{TM}$ is undecidable but Turing-recognizable. From a programming perspective, this means that it's possible to write a procedure `will_accept(program, input)` that takes as its arguments string representations of a program and that program's input and has the following return values:

- If the program would accept that input, `will_accept` must return true.
- If the program would not accept the input, `will_accept` may return false or it may go into an infinite loop.

Now consider the following program, `ow-my-brain.py`:

```python
def will_accept(program, input):
    # An implementation of a recognizer for ATM: Does
    # the program accept the input?
    ...

def main(my_input):
    # Read our own source code
    my_code = open("ow-my-brain.py").read()

    if will_accept(my_code, my_input):
        return False  # i.e., reject
    else:
        return True   # i.e., accept
```

Use proof by contradiction to show that this program must loop infinitely on all inputs. It's fine for this to be a casual "proof sketch" as long as your thinking is clear.
Problem 3

Prove: The collection of decidable languages (R) is closed under the intersection operation.

For Problems 3 and 4, follow the style of proof seen in the textbook, in sections 4.1 and 4.2. E.g., you may want to review the solutions to problems 4.10, 4.12, and 4.14 on pages 213–4.

Problem 4

Prove or disprove: The following language is decidable:

\[ L = \{ (M, n) \mid M \text{ is a Turing machine with } n \text{ states, and there exists a string } w \in \Sigma^* \text{ of length at most } n^2 \text{ such that } M \text{ accepts } w \text{ in at most } n^3 \text{ transitions} \} \]
Problem 5

Determine which one of the following languages is decidable and which one is undecidable, where $M$ is a Turing machine. Briefly justify your answer, either sketching how a Turing machine could decide the language or how you could solve a known undecidable problem if a decider existed for the language.

a. $\{\langle M, a \rangle \mid M$ writes the character a at some point when started on the empty tape}$

b. $\{\langle M \rangle \mid M$ writes a non-blank character at some point when started on the empty tape}$
Problem 6

Consider the language

\[ A_{\varepsilon_{TM}} = \{ \langle M \rangle \mid M \text{ is a TM that accepts } \varepsilon \} . \]

Use proof by reduction to show that \( A_{\varepsilon_{TM}} \) is undecidable.
Problem 7

For each of the following languages, explain whether or not Rice's Theorem applies to those languages.

For those languages for which Rice's Theorem does not apply, state whether the language is decidable or undecidable and give a one-sentence justification for your answer.

a. \( L = \{ \langle M \rangle \mid M \text{ accepts at least one even-length string} \} \)

b. \( L = \{ \langle M \rangle \mid M \text{ never prints the tape symbol under the tape head on a transition}, \text{i.e., it never has a transition where it writes the same symbol it read.} \} \)

c. \( L = \{ \langle M \rangle \mid M \text{ rejects all descriptions of Turing machines} \} \)