Ella, Chidera, Marco

Worksheet #10
Decidability

1. What does it mean for a language to be decidable?

   A language is decidable if there exists a TM, M, that terminates correctly on every input string w.

2. Given computational model **M** and string **w**, what does ⟨M, w⟩ mean?

   <M> is the binary encoding of the TM M followed by the binary encoding of an input string w, so ⟨M, w⟩ is then the binary encoding of the TM M having been given the string w.

3. For A_{TM}, when given an input ⟨M, w⟩ why can't we create a decider for this language by simulating the execution of M on w and then accepting if M accepts w and rejecting if M rejects w?

   The input for the decider would need the output of the simulation of the execution of M on w, but if this simulation never terminates, there would never be an output (we are expecting the output to be the empty language). This means we are essentially trying to create a decider for the Halting Problem, which we know is not possible.

4. Prove that the set of Cartesian products of natural numbers is countable:

   $\mathbb{N} \times \mathbb{N} = \{(x, y) : x \in \mathbb{N}, y \in \mathbb{N}\}$

   Similar to how $\mathbb{Q}$ is countable, we need to create a bijection between $\mathbb{N}$ and $\mathbb{N}$. To do this we can create a table to arrange the Cartesian products of natural numbers.

	1	2	3	4	…
1	(1,1) 1	(2,1) 2	(3, 1) 6	(4, 1) 7	…
2	(1,2) 3	(2,2) 5	(3, 2) 8	(4, 2)	…
3	(1,3) 4	(2,3) 9	(3, 3)	(4, 3)	…
4	(1,4)	(2,4)	(3, 4)	(4, 4)	…
…	…	…	…	…	…

The order of ℕ x ℕ can be diagonalized, just as with the ℚ table. This means that each “step,” or cell, has a corresponding natural number, like an index. Since natural numbers are countable, this means the rows and columns are both countable, so we could extend this pattern for an arbitrarily large table.

f(1) = (1,1)
f(2) = (2,1)
f(3) = (1,2)
f(4) = (1,3)
f(5) = (2,2)
f(6) = (3,1)
f(7) = (4,1)
f(8) = (3,2)

5. Is the following language decidable?

P = { ⟨M⟩ : M is a Turing machine and L(M) = {00, 11, 000, 111}}

Let’s assume that P is decidable.

By Rice’s Theorem, if we assume that P is decidable, we are stating that the language HALT = {<M,w>, where M is a machine that accepts w} is also decidable, because a decider for P would need to know whether or not a simulation of the execution of M on w halts. This is a contradiction since the language HALT is not decidable by theorem 5.1.7 in the textbook. Therefore, since HALT is not decidable, P cannot be either.