Turing Machines and Decidability

1. Give implementation level descriptions for Turing Machines that decides the following two languages.

   - $\{w|w\text{contains an equal number of 0s and 1s}\}$
   - $\{w|w\text{contains an twice as many 0s as 1s}\}$

Both of these involve simply scanning the input and marking off 0s and 1s one at a time (or one 1 for each two 0s in the second TM).

2. A *2-PDA* is a pushdown automata that has two stacks instead of just one. Prove that a 2-PDA is more powerful than a traditional PDA with one stack. *Hint: Show how to simulate a Turing Machine's tape with the 2-PDA.*

This is quite simple. When we need to move the head left, simply pop from stack 1 and push that item to stack 2.

When we want to move the head right, simply pop from stack 2 and push the item to stack 1.

If we want to write to the tape, simply pop the item off and push on the new data.

3. For this question, you will do three separate proofs. Prove that the class of *Decidable Languages* is closed under union, concatenation, and star.

First, we will do union. if A and B are decidable, then $A \cup B$ is decidable because we can simply run both deciders one after the other. Accept iff one of the two deciders accepts.

Concatenation. The challenge here is only the input we want to consider should be on the tape. If A and B are both decidable, then on input w, we non-deterministacally guess where to divide the input. For each guess, we put the first part into decider for A and second part into decider for B. Accept if any branch gets two YES results.

For star, we do something similar. Non-deterministically guess where to put all the dividing points in the input strinng w. For each branch, put each substring into the decider for A to see if that division of w is in A^*

4. Prove the following claim: Let C be a language. Prove that C is Turing-recognizable if and only if a decidable language D exists such that $C = \{x|\exists y((x,y) \in D)\}$

Direction 1: If C is recognizable then there exists a decidable language D as described in the problem.

- C is recognizable so a machine $M_C(x)$ exists that recognizes C

- let $f(n)$ be the maximum number of steps M_C takes to say YES on any string in C of length n. *Note that $f(n)$ is calculable because there are always a finite number of strings in C of any specific length n*
- Define the language D as follows: For every x in C such that $|x| = n$, let $(x, f(n))$ be in D
- The decider for D is: Given (x, y), run $M_C(x)$ for at most y steps. If M_C says YES, return YES. If it takes y steps or more, return NO

Direction 2: If D is decidable, then C is recognizable

- D is decidable, so a decider $M_D(x, y)$ exists for D
- To recognize C, simply do the following: Given x as input, For each possible y in "", Σ^1, Σ^2,etc. if $M_D(x, y)$ says YES, return YES else continue to next possible y
- This is a recognizer because if a y exists that pairs with the given x, we will find it (but we might loop forever).

5. In the *Harry Potter* books / movies, the Weasley family owns a magical clock that can tell when the Weasley children are (among other things) in *mortal danger!*. See this link for details. Is it theoretically possible to build a clock with this functionality? In this problem you will show that building this function is undecidable.

   First, let's change the function of the clock to make this question less grim. Suppose instead of detecting whether Ron Weasley is in mortal danger, the clock detects whether Ron Weasley is in danger of getting his hair cut off. If the clock registers this danger setting, then Ron is going to get his hair cut off unless influenced by an outside force (the clock just knows!!). If the clock does not register this danger setting, then Ron's hair is safe from the clippers!

   Your question is this: Show that this clock's functionality is undecidable through reduction. Specifically, show that if this clock existed, you could construct a machine that decides the halting problem. Describe how to construct this machine. Your solution will likely require Turing Machines, physical components, razors, and Ron Weasley himself!

We want to build a decider for the halting problem given that this clock exists. We do the following:

- Take as input, M and w
- Simulate M on w, tie Ron to a chair and connect a razor to an automated arm mechanism aimed at his hair.
- If M halts on w, engage the arm and chop off Ron's hair. Otherwise do nothing.
- If the clock says Ron's hair is in danger, output "halts", otherwise output "loops forever"