Assignment 8

1. Prove using a reduction to a known undecidable problem that the following problem is undecidable: given a turing machine T and a word w, does T accept ww? Recall the steps: assume the problem is solveable, and that there must be a T' machine that takes as input T and ww and *decides* whether T accepts ww. Use this T' as part of an algorithm that can consequently *decide* a known undecidable problem to reach a contradiction.

   Proof. Assume T accept ww is decidable. Therefore, there exists a T' that halts and decides it.
   T' takes input T and w, the machine will decide whether T accepts ww.
   Use T' as a subroutine to solve $T(E)$, a known undecidable problem T accepts ϵ. $T(E)$ takes $e_T(T)$ as input.
   Preprocessor takes in $e_T(T)$ and produce $e_T(T)$ and feed ϵ (as second input) for T'.
   So we have T accepts ϵ if and only if T' accepts $\epsilon\epsilon$.
   Machine T' accepts if T accepts ϵ but this is a known unsolvable problem.
   Therefore, T'cannot exist and we conclude that T accepts ww is unsolvable.

   □

2. Give an algorithm to decide the following *decidable* problem: given a CFG G, does $G \Rightarrow^\star \epsilon$? That is, given a grammar can it generate the empty word? Be sure that your algorithm is decidable in that it *always halts* on *all inputs* and correctly answers the question. You can use any constructions or algorithms we've developed in class by naming them as a black-box part of your algorithm.

   A Turing machine that decides the problem of CFG G:
   1. Convert G into an equivalent CFG $G = (V, \Sigma, R, S)$ in Chomsky normal form.
   2. If G includes the rule $S \Rightarrow \epsilon$, accept. Otherwise, reject.