Problem 1: Fun with Turing machines

Design a TM with tape alphabet $\Gamma = \{0, 1, \square\}$ that on input $x = b_1 \ldots b_n$ (where $b_i \in \{0, 1\}$), changes the content of the tape to $f(x) = b_10b_20\ldots b_n0$, and halts with the tape head positioned at the beginning of the string. In other words, when started in configuration $(q_s, \epsilon, x)$, the TM terminates in configuration $(q_h, \epsilon, f(x))$ (or, more precisely, a configuration of type $(q_h, \square^*, f(x)\square^*)$).

Your solution should be structured in a way similar to the TM presented in class for the problem of reversing the input string. Specifically, your solution should consist of:

1. A high level description of the design ideas underlying your TM, including a clear description of some invariant to be preserved at every iteration of the computation.

2. An implementation level informal description of the Turing machine.

3. A formal implementation of the TM. You should use JFLAP to draw the diagram of the TM. In the assignment, you should include a print out of the diagram of the TM and write down which states correspond to each stage of the informal TM description in part 2. You should also submit the JFLAP file (.jff) containing the code of the TM.

In the solution to this problem, you should use the definition of Turing machines as implemented in JFLAP. These differ from the Turing machines as defined in the textbook as follows:

- Sipser defines Turing machines with tape that is unbounded only when moving to the right, i.e., if the TM tries to move beyond the left edge of the tape, the tape head stays where it is. In contrast, in JFLAP the tape is unbounded in both directions, and if the TM tries to move beyond the left edge, the tape is extended to the left with a blank symbol.

- In JFLAP, the TM tape head movements are $L, R, S$, where $L$ and $R$ move the head to the left or right, and $S$ leaves the tape head in its
current position. Sipser only allows \( L, R \). You are allowed \( S \) if you find it useful.

**Problem 2**

In this problem you are asked to prove that if a set is both recognizable and co-recongnizable, then it is decidable.

More precisely, you should give a WHILE program \( D \) that on input \([“P_1”, “P_2”, x] \) such that

- “\( P_1 \)” is the encoding of a WHILE program recognizing \( A \)
- “\( P_2 \)” is the encoding of a WHILE program recognizing \( \bar{A} \)
- \( x \) is an arbitrary input value.

terminates and accepts the input if and only if \( x \in A \). (If the input to \( D \) does not satisfy the above properties, \( D \) is not required to terminate.)

Your solution should include:

1. A clear informal description of how your program \( D \) operates, including a proof that your program behaves as required.
2. The actual WHILE program, to be submitted electronically.
3. A proof that if \( A \) is both recongnizable and co-recongizable, then \( A \) is decidable. Specifically, you should show that if there exists two WHILE programs \( P_1 \) and \( P_2 \) that recognize \( A \) and \( \bar{A} \) respectively, then there exist a WHILE program that decides \( A \). (This last part easily follow from part 1 and 2. However, notice that \( D \) is not a decider for \( A \). In fact, as already noted, \( D \) is not even guaranteed to be a decider.)

In this problem, you should use the encoding “\( P \)” of WHILE programs presented in class (on October 4 and/or 9), and to be posted on the webpage.

Your program \( D \) is expected to be a valid WHILE program. In particular, we expect to be able to run your program using the interpreter to test its correctness. You are encouraged to use the WHILE interpreter from the webpage to test your program on your own before submitting the assignment. You can report your testing experiments as part of the assignment if you like, but this is not required.

You should use the encoding of WHILE programs into lists presented in class, and available from the webpage as part of the self interpreter for language WHILE. (Use “\#include self.hw” to import the entire file.)  

*Hint: In your solution you are allowed to use the other functions defined in self.wh.*