Theory of Computation HW7 Decidable Languages Ryzeson Maravich

Let the language $E_{DFA} = \{\langle A \rangle | A \text{ is a DFA and } L(A) = \{\}\}$.
Let the language $EQ_{DFA} = \{\langle B, C \rangle | B \text{ and } C \text{ are DFAs and } L(B) = L(C)\}$.

1

Let $ALL_{CFG} = \{\langle A \rangle | A \text{ is a DFA and } L(A) = \Sigma^*\}$.
Claim: ALL_{CFG} is decidable.
Proof: Define the Turing machine $M =$ "On input $\langle A \rangle$ where A is a DFA:

1. Mark the start state.
2. Repeat step 3 until no new steps are marked.
3. For each unmarked state, mark it if there is a transition to it from a marked state. If any non-final state is marked, *reject*. Otherwise *accept*."

Every DFA has a finite number of states, so this process will eventually complete.

2

Let $A = \{\langle R, S \rangle | R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S)\}$.
Claim: A is decidable.
Proof: Define the Turing machine $M_2 =$ "On input $\langle R, S \rangle$ where R and S are regular expressions:

1. Construct a DFA C whose language $L(C) = \{R \cap \bar{S}\}$.
2. Run the Turing machine for E_{DFA} on C. If it accepts, *accept*; otherwise *reject*."

Regular expressions are closed under intersection and complement, so the language C recognizes is regular.

3

Let $S = \{\langle B \rangle | B \text{ is a DFA that accepts } w^R \text{ whenever it accepts } w\}$.
Claim: S is decidable.
Proof: Define the Turing machine $M_3 =$ "On input $\langle B \rangle$ where B is a DFA:

1. Construct a DFA G whose language $L(G) = \{L(B) \cap L(\overline{B^R})\}$.
2. Run the Turing machine for E_{DFA} on G. If it accepts, *accept*; otherwise *reject*."

Regular expressions are closed under intersection, complement, and reverse, so the language G recognizes is regular.

I affirm that I have upheld the highest principles of honesty and integrity in my academic work and have not witnessed a violation of the honor code.