4.4.1 Consider the following Turing machine
M = “on input <A>, where A is a DFA
1. Create a DFA B such that L(B) = \( \Sigma^* \)
2. Submit <A,B> to the decider for EQ\textsubscript{DFA}
3. If it accepts, \textbf{accept}
4. If it reject, \textbf{reject}.”

M is clearly a decider since steps 1, 3, and 4 will not create an infinite loop and step 2 calls a decider. Furthermore, M will accept those DFA’s whose language is the same as B’s language – i.e. DFA’s whose language is \( \Sigma^* \) -- and will reject all other languages. Therefore, M is a decider for ALL\textsubscript{DFA} so ALL\textsubscript{DFA} is decidable.

4.12 Consider the following Turing machine
M = “On input <R,S> where R and S are regular expressions
1. Convert R to DFA A and S to DFA B
2. Construct DFA C such that \( L(C) = L(B) \cap L(A) \)
3. Submit <A,C> to the decider for EQ\textsubscript{DFA}
4. If it accepts, \textbf{accept}
5. If it rejects, \textbf{reject}.”

M is a decider since steps 1, 2, 4, and 5 will not create and infinite loops and step 2 calls a decider. Also, M accepts <R,S> iff \( L(R) = L(R) \cap L(S) \) -- i.e., iff \( L(R) \subseteq L(S) \). Therefore, M is a decider for A so A is decidable.

4.19.1 Note that if the DFA accepts \( w^R \) whenever it accepts \( w \), then \( L(M) = L(M^R) \), where \( M^R \) is the DFA that accepts the reverse of strings accepted by M. In a previous homework, we proved that if M is a regular language, then so is \( M^R \).

Consider the following Turing machine
T = “On input <M>, where M is a DFA
1. Construct DFA N that accept the reverse of strings accepted by M
2. Submit <M,N> to the decider for EQ\textsubscript{DFA}
3. If it accepts, \textbf{accept}
4. If it rejects, \textbf{reject}.”

T is a decider since 1, 3, and 4 will not create an infinite loop and step 2 calls a decider. Also, T accept M iff \( L(M) = L(M^R) \). Therefore T decides S so S is decidable.