Recall: $A_{DFA} = \{ <M, w> \mid M \text{ is a DFA that accepts } w \}$

(a) Is $<M, 0100> \in A_{DFA}$?
Yes. On input 0100, $M$ ends in state $q$, which is an accepting state.

(b) Is $<M, 011> \in A_{DFA}$?
No. On input 011, $M$ ends in state $q_2$ which is not accepting.

(c) Is $<M> \in A_{DFA}$?
No. Does not type check.

(d) Is $<M, 0100> \in A_{REX}$? $A_{REX} = \{ <R, w> \mid R \text{ is a regular expression that generates } w \}$
No. Does not type check. $M$ is a DFA, not a regular expression.

(e) Is $<M> \in E_{DFA}$?
No. $L(M) \neq \phi$, $\emptyset \in L(M)$.

(f) Is $<M, M> \in EQ_{DFA}$?
Yes. $L(M) = L(M)$. 
4.3 \( \text{ALLDFA} = \{ \langle A \rangle \mid A \text{ is a DFA and } L(A) = \Sigma^* \} \).

Show that \( \text{ALLDFA} \) is decidable.

\[ L(A) = \Sigma^* \Rightarrow \overline{L(A)} = \emptyset \]

Define Turing Machine \( M = \)

"On input \( \langle A \rangle \), where \( A \) is a DFA,"

1. Construct DFA \( \overline{A} \)
2. Run the TM \( F \) which decides \( \text{EDFA} \), on input \( \langle A \rangle \).
3. If \( F \) accepts, accept. If \( F \) rejects, reject."

\( M \) is a decider because \( F \) is a decider, \& \( M \) accepts \( \langle A \rangle \) iff \( L(A) = \emptyset \) i.e. \( L(A) = \Sigma^* \). \( M \) decides \( \text{ALLDFA} \), so \( \text{ALLDFA} \) is decidable.

\((4.4) \text{ AceCFA} = \{ \langle A \rangle \mid A \text{ is a CFG that generates } \varepsilon \} \)

Show that \( \text{AceCFA} \) is decidable.

Define Turing Machine \( M = \)

"On input \( \langle A \rangle \), where \( A \) is a CFG"

1. Run the TM \( F \) which decides \( \text{ACFG} \), on input \( \langle A, \varepsilon \rangle \).
2. If \( F \) accepts, accept. If \( F \) rejects, reject."

\( M \) is a decider because \( F \) is a decider, \& \( M \) accepts \( \langle A \rangle \) iff \( \text{ACFG} \) accepts \( \langle A, \varepsilon \rangle \) (i.e. \( A \) generates \( \varepsilon \)). \( M \) decides \( \text{ACFG} \).

\( \therefore \) \( \text{AceCFA} \) is decidable.
(4.5) \( \overline{E_{TM}} = \{ \langle M \rangle \mid M \text{ is a T.M., } L(M) = \emptyset \} \).

Show that \( \overline{E_{TM}} \) is recognizable.

\[ \overline{E_{TM}} = \{ \langle M \rangle \mid M \text{ is a T.M., } L(M) \neq \emptyset \} \]

Define T.M. \( N = \langle \rangle \) on input \( \langle M \rangle \), where \( M \) is a T.M.

1. For \( i = 1, 2, 3, \ldots \) up to \( 2^m \).
2. Run \( M \) for \( i \) steps on strings \( S_1, S_2, \ldots, S_i \) (the first \( i \) strings over the alphabet in shortlex order).
3. If \( M \) accepts, accept.

On input \( \langle M \rangle \), if \( L(M) \neq \emptyset \), there exists \( i, j \) such that \( S_i \) is accepted by \( M \) in \( j \) steps. \( N \) will eventually run \( M \) on input \( S_i \) for \( j \) steps & accept.

If \( L(M) = \emptyset \), then \( M \) does not accept any strings, so \( N \) simply loops, since no value of \( i \) will result in \( M \) accepting some string.

\( N \) recognizes \( \overline{E_{TM}} \), so \( \overline{E_{TM}} \) is recognizable.


(4.30) Let \( A = \{ \langle M_1 \rangle, \langle M_2 \rangle, \ldots \} \) be a Turing-recognizable language, where every \( M_i \) is a decider. Show that there exists a decidable language \( D \) not decided by any of the deciders \( M_i \).

We prove this by constructing a decidable language using diagonalization.
let \( S = \{ S_0, S_1, \ldots \} \) be the shortlex order of strings over the alphabet \( E \).

Observation: Since \( A \) is recognizable, there is some enumerator \( E \) that enumerates \( A \).

Construct T.M \( T = \) "On input \( w \)

1. Let \( i \) be the index of \( w \) in \( S \) (i.e. \( w = S_i \))
2. Use \( E \) to obtain \( \langle M_i \rangle \).
3. Run \( M_i \) on input \( w \).
4. If \( M_i \) accepts, reject. If \( M_i \) rejects, accept."

\( T \) is a decider because each \( M_i \) is a decider.

However \( \langle T \rangle \) doesn't appear in \( A \) because \( T \) differs from every \( M_i \) on at least one input - \( S_i \). \( \therefore \) \( L(T) \) is a decidable language not decided by any \( M_i \).