4.1 (a) Yes. M accepts 0100.

(b) No. M rejects 011.

(c) No. No input string is given.

(d) No. R is not a DFA.

(e) No. $L(M)$ is not empty.

(f) Yes. By definition, $L(M) = L(M)$.

4.2 We define the language EQ_{DR} as

$$EQ_{DR} = \{\langle A, R\rangle \mid A \text{ is a DFA}, R \text{ is a regular expression, and } L(A) = L(R)\}$$

To show EQ_{DR} is decidable, we construct Turing machine M which decides EQ_{DR}.

M = "On input $\langle A, R\rangle$

1. Construct DFA B which recognises R (see theorem 1.39).
2. Run $EQ_{DFA}\langle A, B\rangle$ (see theorem 4.5). If $EQ_{DFA}\langle A, B\rangle$ accepts, *accept*, else *reject*."

4.3 To show ALL_{DFA} is decidable, we construct Turing machine M which decides ALL_{DFA}:

M = "On input $\langle A\rangle$, where A is a DFA

1. Construct DFA A' by running algorithm $MINIMIZE$ on A (see problem 7.25).
2. If A' consists of a single accepting state, *accept*, else *reject*."

4.4 To show $A\epsilon_{CFG}$ is decidable, we construct Turing machine M which decides $A\epsilon_{CFG}$:

M = "On input $\langle A\rangle$, where A is a CFG

1. Construct CFG A' by converting A into Chomsky normal form (use algorithm described by theorem 2.9).
2. If A' consists of a single rule with terminal ϵ, *accept*, else *reject*."