1) a. Yes, M is DFA and 0100 is accepted by M. TM M accepts <M, 0100>

b. No, M is DFA and 011 is rejected by M. TM M rejects <M, 011>

c. Yes, M is DFA and ϕ is accepted by M. TM M accepts <M>

d. No, M is DFA and 0100 is not regular expression. TM M rejects <M, 0100>

e. Yes, M is DFA and final state is not marked. TM T accept <M>

f. yes. M and M are DFAs and L(M) = L(M), TM F accept <M, M>

2)

If input is not in the form <G> where G is a CFG

S rejects input

otherwise: input must be in the form <G> where G is a CFG and w is a string run "CNF" on <G> and get <H> which the Chomsky form for <G> repeat for every possible derivation of $2n-1$ step <H> like D

[if D yields $S \to \varepsilon$, S should accept <G>]
lh wise go to top of loop

If none of the derivation $2n-1$ step <H> generate $S \to \varepsilon$ should reject <G>

3.

If input is not in the form <A> where A is a DFA

S rejects input

otherwise: input must be in the form <A> where A is some DFA starting by marking start state repeat until no new state is marked

make any state that has an incoming transition from an already marked state

If any final state is marked, TM accept the <D>. otherwise reject.