Homework XI Sample Solutions

Problem 1.
The (potentially) reachable configurations appear in the right-hand column of the table below with a brief explanation in the left-hand column. A thorough analysis involves numerous cases.

if EXECUTE only occurs at start	1. <WAITING_FOR_COMMAND, DISCONNECTED>
if GO only occurs at start	2. <SETTING_UP, IDLE>
if GO and EXECUTE occur at start	nondeterministic - same as either 1 or 2
if RESET but not tm(2) occurs in 2	3. <SETTING_UP, DISCONNECTED>
if tm(2) but not RESET occurs in 2	4. <WAITING_FOR_COMMAND, IDLE>
if both RESET and tm(2) occur in 2	same as 1
if tm(2) but not GO occurs in 3	same as 1
if GO but not tm(2) occurs in 3	same as 2
if both tm(2) and GO occur in 3	same as 4
if EXECUTE only occurs in 4	5. <COMPARING, IDLE>
if RESET only occurs in 4	same as 1
if GO only occurs in 4	same as 2
if EXECUTE and RESET but not GO occur in 4	6. <COMPARING, DISCONNECTED>
ir RESET and GO but not EXECUTE occur in 4	same as 3
if EXECUTE and GO but not RESET occur in 4	nondeterministic - same as either 2 or 5
if EXECUTE, GO and RESET occur in 4	nondeterministic - same as either 3 or 6
if en(COMPARING) but not RESET or OUT_OF_RANGE occur in 5	7. <COMPARING, OPERATING>
if en(COMPARING) and OUT_OF_RANGE but not RESET occur in 5	8. <GENERATING_ALARM, OPERATING>
if RESET but not OUT_OF_RANGE occurs in 5	same as 1
if RESET and OUT_OF_RANGE occur in 5	nondeterministic - either 9. <GENERATING_ALARM, DISCONNECTED> or same as 1
if RESET only occurs in 6	same as 1
if OUT_OF_RANGE only occurs in 6	same as 9
if GO only occurs in 6	same as 5
if both RESET and GO but not OUT_OF_RANGE occur in 6	same as 4
if both GO and OUT_OF_RANGE but not RESET occur in 6	10. <GENERATING_ALARM, IDLE>
if all of GO, RESET and OUT_OF_RANGE occur in 6	nondeterministic - same as either 4 or 10

if RESET but not OUT_OF_RANGE occurs in 7	same as 1
if OUT_OF_RANGE but not RESET occurs in 7	same as 8
if RESET and OUT_OF_RANGE both occur in 7	nondeterministic - same as either 1 or 8
if ex(COMPARING) only occurs in 8	same as 10
if ex(COMPARING) and STOP but not RESET occur in 8	same as 4
if RESET but not STOP occurs in 8	same as 9
if RESET and STOP occur in 8	same as 1
if GO only occurs in 9	same as 10
if STOP only occurs in 9	same as 1
if both STOP and GO occur in 9	same as 4
if RESET only occurs in 10	same as 9
if STOP only occurs in 10	same as 4
if STOP and RESET both occur in 10	same as 1

Problem 2.

There are a variety of ways to resolve this situation. The circumstance arises from the potential conjunction of the generation of an internal occurrence of the 'RESET' event by the 'STOP' event and a simultaneous occurrence of an external 'GO' event that could move the 'MONITORING' subchart out of its initial state at the moment the 'PROCESSING' subchart reaches its initial state. Therefore, one direct way to avoid this is to add a condition to the 'GO' transition in the 'MONITORING' subchart, namely not(ex(GENERATING_ALARM)), to assure that the 'MONITORING' subchart is blocked from leaving its initial state, when the immediately previous transition was not the 'STOP' transition (generating a one step delayed 'RESET').

This might be considered a moot change for this statechart since we found in problem 1 that the configuration <WAITING_FOR_COMMAND, OPERATING> is not reachable. But this problem is about transitions from this state, not transitions to it, and transitions *from* this state are possible. It is important for a statechart to forbid transitions that are specifically forbidden as well as to describe those that are allowed. For instance, without such a alteration other changes made elsewhere may add <WAITING_FOR_COMMAND, OPERATING> to the reachable states, and then the forbidden behavior would become sanctioned by the description.