1. For each of the following problems construct a deterministic finite automaton which describes or recognizes the language given. Write all DFAs in the form of a transition diagram. The underlying alphabet is $\Sigma = \{a, b\}$.

Be sure to give a DFA for each part, and not an NFA unless the problem asks for one. Do not use any notational conveniences or shortcuts given in lecture.

The notation $#a(w)$ is used below to refer to the number of $a$’s occurring in the string $w$. For example, $#a(bbaba) = 2$.

Note: the first seven parts are the same languages which were questions on Homework #1. It would be very instructive to compare the DFA and the regular expression for these languages– it is suggested that you do so.

a. \{ $w \mid w$ begins with $abab$ \}

b. \{ $w \mid w$ ends with $abab$ \}

c. \{ $w \mid w$ begins with $ab$ and ends with $ba$ \}

Note: The string $aba$ is in this language.

d. \{ $w \mid #a(w) \equiv 2 \pmod{5}$ \}

Recall that $i \equiv j \pmod{k}$ if and only if $i - j$ is divisible by $k$.

e. \{ $w \mid #a(w)$ is even or $|w|$ is even \}

f. \{ $w \mid aaa$ is a substring of $w$ \}

g. \{ $w \mid aaa$ is not a substring of $w$ \}

h. \{ $w \mid w$ contains exactly one occurrence of the substring $aaa$ \}

Note: the string $aaaa$ has two occurrences of $aaa$.

i. \{ $w \mid$ either $#a(w)$ is divisible by 3 or $w$ begins with $bbb$ \}

Also write a regular expression which describes or recognizes this language.

j. \{ $w \mid #a(w) \equiv 1 \pmod{3}$ and $#b(w)$ is odd \}

k. \{ $w \mid$ neither $aa$ nor $bb$ is a substring of $w$ \}

l. Write an NFA for the language in part #1i.

2. Consider the following language:

\{ $w \mid w \in \{0, 1\}^*$ and $w$ contains an even number of 0s, and $w$ does not contain three consecutive 1s \}

Determine whether each of the following DFAs correctly describes or recognizes this language or not. Identify why each incorrect DFA is wrong– give a string which the DFA doesn’t give the right results for, and identify what result the DFA should give for that string, and what result it actually gives.