Morgan Baccus
Cpt S 317
Homework #6

1) After converting to table:

| | State | 0 | 1 |
|---|---|---|---|
| → | a | b | f |
| | b | g | c |
| * | c | a | c |
| | ~~d~~ | ~~c~~ | ~~g~~ |
| | e | h | f |
| | f | c | g |
| | g | g | e |
| | h | g | c |

'd' is unreachable so we can remove this state.

$\{a, b, e, f, g, h\}$ $\{c\}$

$(a,b) \xrightarrow{0} (b,g)$ ✓
$(a,b) \xrightarrow{1} (f,c)$ ✗

$(b,c) \xrightarrow{0} (g,a)$ ✓
$(b,c) \xrightarrow{1} (c,c)$ ✓

$(a,e) \xrightarrow{0} (b,h)$ ✓
$(a,e) \xrightarrow{1} (f,f)$ ✓

$(e,f) \xrightarrow{0} (h,c)$ ✗
$(e,f) \xrightarrow{1} (f,g)$ ✓

$(b,f) \xrightarrow{0} (g,c)$ ✗
$(b,f) \xrightarrow{1} (c,g)$ ✗

$(e,g) \xrightarrow{0} (h,g)$ ✓
$(e,g) \xrightarrow{1} (f,e)$ ✓

$(c,h) \xrightarrow{0} (a,g)$ ✓
$(c,h) \xrightarrow{1} (c,c)$ ✓

continued)

[b, c, h] [a, e, g] [f] [c] 1 equivalence states

| | |
|---|---|
| $(b,c) \xrightarrow{0} (g,a)$ ✓<br>$(b,c) \xrightarrow{1} (c,c)$ ✓ | $(c,h) \xrightarrow{0} (a,g)$ ✓<br>$(c,h) \xrightarrow{1} (c,c)$ ✓ |
| $(a,e) \xrightarrow{0} (b,h)$ ✓<br>$(a,e) \xrightarrow{1} (f,f)$ ✓ | $(e,g) \xrightarrow{0} (h,g)$ x<br>$(e,g) \xrightarrow{1} (f,e)$ x |

[b, c, h] [a, e] [g] [f] [c] 2 equivalence states

| | |
|---|---|
| $(b,c) \xrightarrow{0} (g,a)$ x<br>$(b,c) \xrightarrow{1} (c,c)$ ✓ | $(c,h) \xrightarrow{0} (a,g)$ x<br>$(c,h) \xrightarrow{1} (c,c)$ ✓ |
| $(b,h) \xrightarrow{0} (g,g)$ ✓<br>$(b,h) \xrightarrow{1} (c,c)$ ✓ | $(a,e) \xrightarrow{0} (b,h)$ ✓<br>$(a,e) \xrightarrow{1} (f,f)$ ✓ |

[b, h] [c] [a, e] [g] [f] [c] 3 equivalence states

1 continued)

$(b,h) \xrightarrow{0} (g,g)$ ✓

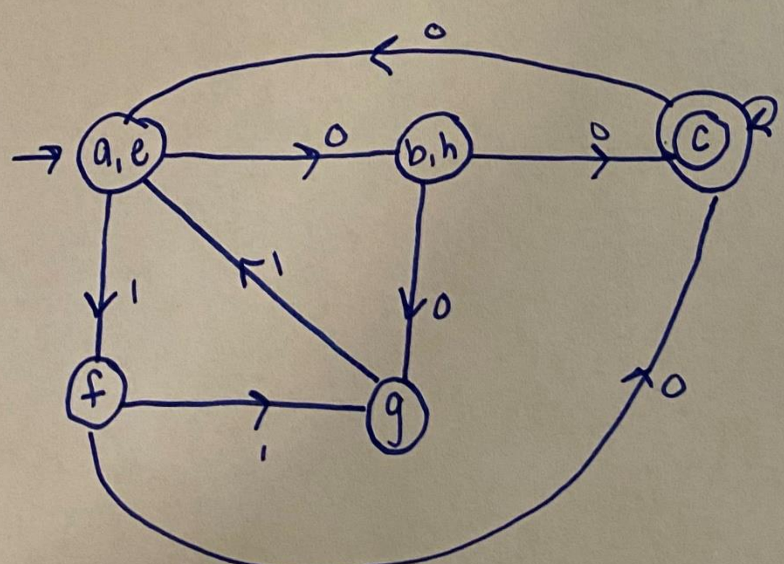
$(b,h) \xrightarrow{1} (c,c)$ ✓

$(a,e) \xrightarrow{0} (b,h)$ ✓

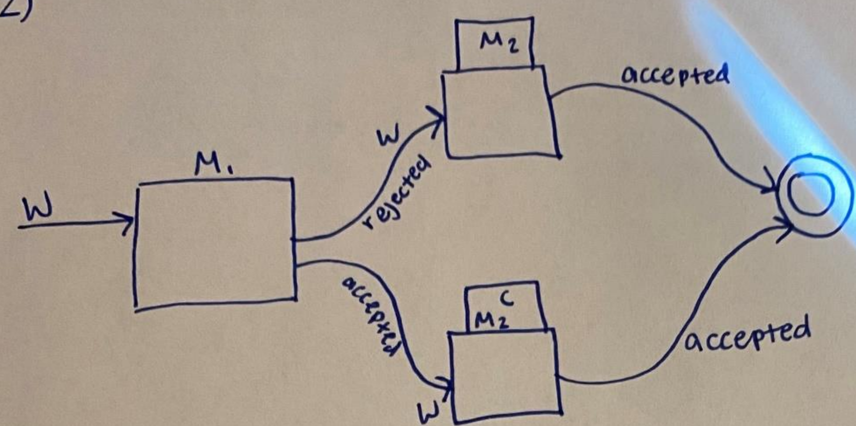
$(a,e) \xrightarrow{1} (f,f)$ ✓

Stop here because 3, 4 equivalence states are equal.

| State | 0 | 1 |
|---|---|---|
| → [a,e] | [b,h] | f |
| [b,h] | g | c |
| * c | [a,e] | c |
| f | c | g |
| g | g | [a,c] |



2)




$M_1$ is NFA

$M_2$ is NFA

$M_2^c$ is complement of $M_2$.

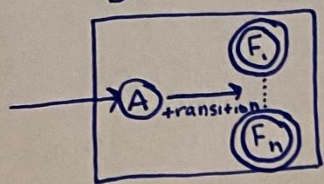
Feed string to $M_1$ $\xrightarrow{\text{accepted}}$ feed to $M_2$

$\xrightarrow{\text{rejected}}$ feed to $M_2^c$

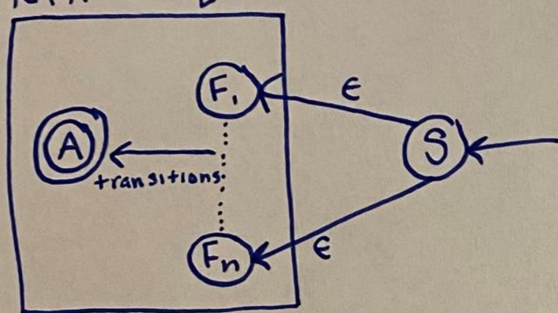
If either $M_2$ accepts or $M_2^c$ accepts, accept string.

3)

DFA of L




NFA of $L^R$




* new initial state S
* Reverse all transitions
* Change final states to non-final
* Change initial state to final

Since we have an NFA accepting $L^R$, $L^R$ is regular.

4)

Since L is regular, there is a DFA which accepts this language.

In that DFA remove ~~the~~ the dead state if there is any and remove transitions involved with it.

Now make all states as final states.

Now we have a DFA which accepts Prefix(L).

Since there is a DFa for this language, it is regular.

5)

Let us assume that this DFA is D. If we understand the problem statement that is given here, we need to show that there is a string x which when concatenated with a string y of same length, then we must be able to deduce that x belongs to half (L).
Assume that we have to begin the process from an initial state say, q0 and we have to reach the final state qn, such that if on taking x as an input the automata reaches a state qi and we need to provide further an input y which is of same length as x, so that when y is given as an input, then the automation can reach to final state qn from the current state of qi.
This can be represented as – $\delta$ (q0, y) $\in$ qn
In order to get qn there exists suppose n number of states. Let Sn be the number of sets to reach qn.
As far as we have come, we can say now that if qi $\in$ Sn, then x $\in$ half (L). [This is what we need to prove].
In order to deduce this, we just need to track the Sn and qi. To store this state of Sn we suppose there is another DFA – D’. So the set of states contained in D can be say T. The set of states that belong to D’ will be T X 2T. Here we can now define a transition function such that, if x is fed as input to D to reach state qi, similarly x can take D’ to state Sn.
Equation 1.1:
$\delta'$ (q, S), a = $\delta$(q, a), prev(S) [Definition for the above described transition function]
Similarly assuming if for the DFA D’, q0’ is the initial state, by induction we can get the below equation from equation 1.1.
Equation 1.2:
$\delta'$ (q0’ , x) = $\delta$(q0, x), Sn , where |x|
Since, x is regarded as an accepted input if and only if $\delta$(q0, x) $\in$ Sn.
Therefore, we conclude that D’ accepts half (L). As, D’ is a DFA that can be constructed, using half (L), we can finally say that half (L) is regular.