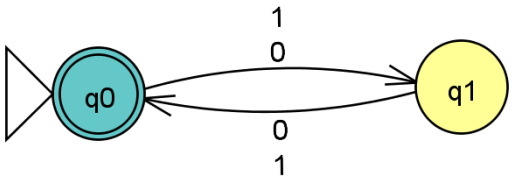
Amit Nijjar, I worked alone

A11489111

4/17/17

CSE 105 HW 2

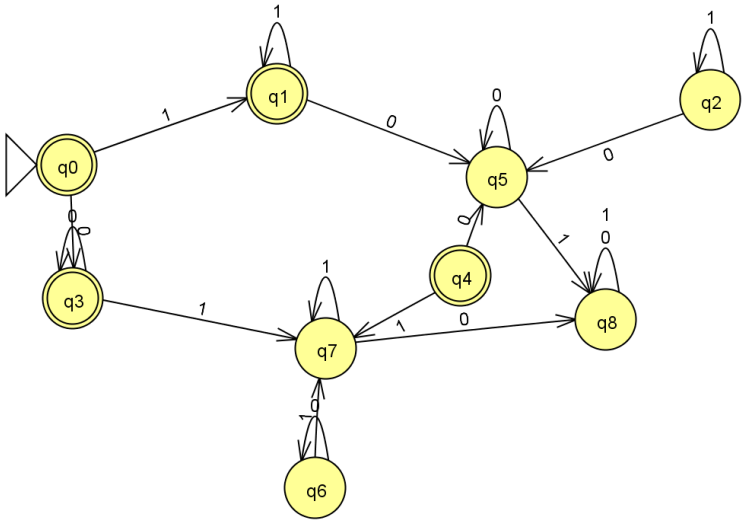
1)



Brief Description: q0 is the initial state, and is an accepting state because it will be reached whenever there is an even number of inputs. Q1 is included to keep track of when there is an odd number of inputs. These two states are needed to see if m+n is an non-negative integer.

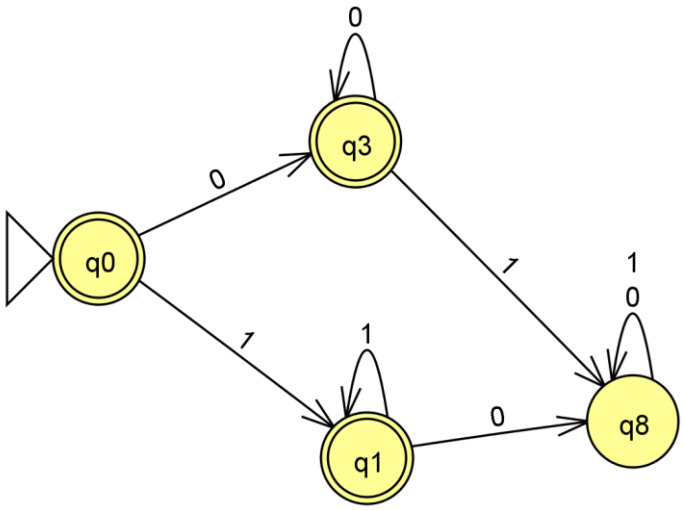
2)

a) There would be 9 states in this dfa because we are combining two dfas. When we do this, we multiply the number of existing states because the intersection of the two dfas would be the same thing as going through both dfa's in parallel. Lastly, 3x3 = 9 so there would be 9 states. Running HW2.hs has confirmed that there would be 9 states.



b) -States 6, 4, and 2 can be eliminated because they can never be reached

- States 5 and 7 can be absorbed into 8 because they are redundant. The combined dfa is accepting only if there are only 1's or 0's in the string. If there is ever a switch from 1 to 0 or 0 to 1, then the dfa will reject the string. The combined dfa can be reduced to this:



JFLAP code for the reduced dfa:

<?xml version="1.0" encoding="UTF-8" standalone="no"?><!--Created with JFLAP 6.4.--><structure>

<type>fa</type>

<automaton>

<!--The list of states.-->

<state id="0" name="q0">

<x>47.0</x>

<y>109.0</y>

<initial/>

<final/>

</state>

<state id="1" name="q1">

<x>164.0</x>

<y>196.0</y>

<final/>

</state>

<state id="3" name="q3">

<x>159.0</x>

<y>58.0</y>

<final/>

</state>

<state id="8" name="q8">

<x>280.0</x>

<y>180.0</y>

</state>

<!--The list of transitions.-->

<transition>

<from>3</from>

<to>8</to>

<read>1</read>

</transition>

<transition>

<from>1</from>

<to>8</to>

<read>0</read>

</transition>

<transition>

<from>3</from>

<to>3</to>

<read>0</read>

</transition>

<transition>

<from>8</from>

<to>8</to>

<read>0</read>

</transition>

<transition>

<from>1</from>

<to>1</to>

<read>1</read>

</transition>

<transition>

<from>8</from>

<to>8</to>

<read>1</read>

</transition>

<transition>

<from>0</from>

<to>3</to>

<read>0</read>

</transition>

<transition>

<from>0</from>

<to>1</to>

<read>1</read>

</transition>

</automaton>

</structure>

3)

Setup: A is a regular language, so there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes A.

Construction: M' =

Q'= Q - the same states as A

δ'(q, x)= {

δ(q,x) if q is in Q

} – the same transitions as A

q'= q' – the same initial state as A

F' = ev{f} (only the even F) – only accept the even states from the accepting states

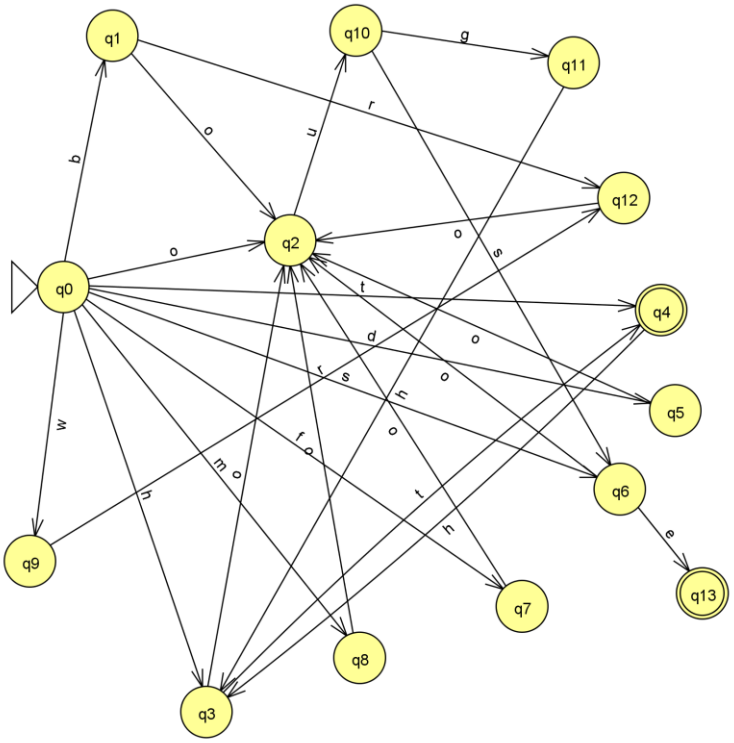
Justification

1) w is accepted by M' is given. This is accepted in even only because the accepting state of M' has to be even. All even states of Q' are in EvenOnly(L). Because the accepting state in M' is also in EvenOnly(L) and w is accepted by M', w is in EvenOnly(L).

2) y is accepted by EvenOnly(L) is given. This is accepted in M' because the accepting state of M' has to be even and therefore in EvenOnly(L). All even states of Q' are in EvenOnly(L). Because the accepting state in M' is also in EvenOnly(L) and y is in EvenOnly(L), y is in M'.

4)

a)



JFLAP:

<?xml version="1.0" encoding="UTF-8" standalone="no"?><!--Created with JFLAP 6.4.--><structure>

<type>fa</type>

<automaton>

<!--The list of states.-->

<state id="0" name="q0">

<x>64.0</x>

<y>221.0</y>

<initial/>

</state>

<state id="1" name="q1">

<x>102.0</x>

<y>26.0</y>

</state>

<state id="2" name="q2">

<x>240.0</x>

<y>185.0</y>

</state>

<state id="3" name="q3">

<x>175.0</x>

<y>551.0</y>

</state>

<state id="4" name="q4">

<x>528.0</x>

<y>239.0</y>

<final/>

</state>

<state id="5" name="q5">

<x>539.0</x>

<y>317.0</y>

</state>

<state id="6" name="q6">

<x>496.0</x>

<y>378.0</y>

</state>

<state id="7" name="q7">

<x>420.0</x>

<y>469.0</y>

</state>

<state id="8" name="q8">

<x>294.0</x>

<y>509.0</y>

</state>

<state id="9" name="q9">

<x>38.0</x>

<y>434.0</y>

</state>

<state id="10" name="q10">

<x>291.0</x>

<y>22.0</y>

</state>

<state id="11" name="q11">

<x>460.0</x>

<y>47.0</y>

</state>

<state id="12" name="q12">

<x>499.0</x>

<y>152.0</y>

</state>

<state id="13" name="q13">

<x>560.0</x>

<y>459.0</y>

<final/>

</state>

<!--The list of transitions.-->

<transition>

<from>6</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>0</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>8</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>7</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>5</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>2</from>

<to>10</to>

<read>u</read>

</transition>

<transition>

<from>1</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>0</from>

<to>9</to>

<read>w</read>

</transition>

<transition>

<from>12</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>9</from>

<to>12</to>

<read>r</read>

</transition>

<transition>

<from>10</from>

<to>11</to>

<read>g</read>

</transition>

<transition>

<from>3</from>

<to>2</to>

<read>o</read>

</transition>

<transition>

<from>11</from>

<to>3</to>

<read>h</read>

</transition>

<transition>

<from>10</from>

<to>6</to>

<read>s</read>

</transition>

<transition>

<from>1</from>

<to>12</to>

<read>r</read>

</transition>

<transition>

<from>0</from>

<to>1</to>

<read>b</read>

</transition>

<transition>

<from>0</from>

<to>5</to>

<read>d</read>

</transition>

<transition>

<from>0</from>

<to>7</to>

<read>f</read>

</transition>

```
<transition>&#13;
        <from>0</from>&#13;
        <to>8</to>&#13;
        <read>m</read>&#13;
</transition>&#13;
<transition>&#13;
        <from>3</from>&#13;
        <to>4</to>&#13;
        <read>t</read>&#13;
</transition>&#13;
<transition>&#13;
        <from>0</from>&#13;
        <to>3</to>&#13;
        <read>h</read>&#13;
</transition>&#13;
<transition>&#13;
        <from>4</from>&#13;
        <to>3</to>&#13;
        <read>h</read>&#13;
</transition>&#13;
<transition>&#13;
        <from>0</from>&#13;
        <to>6</to>&#13;
        <read>s</read>&#13;
</transition>&#13;
<transition>&#13;
        <from>0</from>&#13;
        <to>4</to>&#13;
        <read>t</read>&#13;
```

```
		</transition>&#13;
		<transition>&#13;
			<from>6</from>&#13;
			<to>13</to>&#13;
			<read>e</read>&#13;
		</transition>&#13;
	</automaton>&#13;
</structure>
```


b) This design has 14 states and has paths for each string in the language; it has two accept states that can be reached for any of the strings in the language given. Each state essentially represents a letter from the strings in the language.