CMPT 125 Assignment 9 Solutions

Question 1 [10 marks]
Let \( \Sigma = \{a, b, c\} \) be the alphabet for this part. Draw an FSM bubble diagram which accepts the language of all strings with an even total number of a's and an even total number of b's. Both must be even. The number of c's can be anything. And before you ask, yes, 0 is an even number. Examples: Your machine should accept: abcabc, acca, abbcbbba, ccccc, empty string \( \{\lambda\} \), but reject: abc, cccacc, bbabb.

Solution:

Notes:
- Every state has a self-loop with input c
- Parity of a and b alternate. However, each a and b have two possible parities; therefore there are four states in total besides the start state

Question 2 [10 marks]
Let \( \Sigma = \{0, 1\} \) be the alphabet for this part. Draw an FSM bubble diagram which accepts the language of all binary strings which represent integers evenly divisible by 3. Thus, your machine should accept 0, 11, 110, 1001, 1100, 1111, 10010, etc. We won't be fussy about leading 0's on your integers, so you have the option to accept or reject 00, 011, and also empty string \( \{\lambda\} \).
Solution:
Notes:
- An input of 0 is multiplication by 2
- An input of 1 is multiplication by 2 and then adding 1
- The following table summarizes the behaviour of remainders when processing the next input of 0 or 1

<table>
<thead>
<tr>
<th>State</th>
<th>Equivalent representation</th>
<th>Result of processing a next input of 0</th>
<th>Result of processing a next input of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remainder 0</td>
<td>3N</td>
<td>2*3N → remainder 0</td>
<td>2*3N+1 → remainder 1</td>
</tr>
<tr>
<td>Remainder 1</td>
<td>3N+1</td>
<td>2*(3N+1) = 6N+2 → remainder 2</td>
<td>2*(3N+1)+1 = 6N+3 → remainder 0</td>
</tr>
<tr>
<td>Remainder 2</td>
<td>3N+2</td>
<td>2*(3N+2) = 6N+4 → remainder 1</td>
<td>2*(3N+2)+1 = 6N+5 → remainder 2</td>
</tr>
</tbody>
</table>

**Question 3 [5 marks]**
Write a POSIX-style regular expression that represents all lines that begin with b or B and are followed by 0 or more characters.

Solution: The following are all correct (but the list is not exhaustive)
- `^[bB].*$`
- `^[bB].*$`
- `^(B|b).*$`

End of line character is optional

**Question 4 [5 marks]**
Write a POSIX-style regular expression that represents all lines that contain a word that ends in ion, but not the word "ion" by itself.

Solution: The following are all correct (but the list is not exhaustive)
- `.*<[A-Za-z]+ion\>.*$`
- `^[.]*<[A-Za-z]+ion\>.*$`

End and beginning of line characters are optional