1. Use Thompson’s algorithm to create an NFA for the following regular expressions:

(a) \((0 + 1)^*\)

(b) \(01^* + 10^*\)
(c) $(0 + \epsilon)^*1^*(0 + \epsilon)^*$

![Diagram of NFA and DFA]

2. Use the incremental subset construction to build a DFA that accepts the same language as the following NFAs:

(a)

![Diagram of NFA and DFA]

Solution:

![Diagram of DFA]

(b)

![Diagram of NFA and DFA]

Solution:
1. Introducing Nondeterminism

Nondeterministic Finite Automata (NFA) is a finite set whose elements are called states, \( q \), and (b) they can take transitions without reading any symbol from the input; these are the \( \epsilon \)-transitions are not transitions taken on the symbol "\( \epsilon \)."

Similarly, it is possible for the machine to reach the accepting state without reading any symbol from the input.

- Beware:
  - \( \epsilon \)
  - \( a \)
  - \( q \)
  - \( 0 \)

While there are states with missing transitions, draw the missing transitions creating any new states.

**Figure 1:** Nondeterministic automaton

![NFA Diagram](image)

(c)

**Figure 7:** DFA \( \det(N) \) with only relevant states

<table>
<thead>
<tr>
<th>( N )</th>
<th>( s, u )</th>
<th>( q, a )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>( \epsilon )</td>
<td>( q_0 )</td>
<td>( q_0 )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( q_0 )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td>( q_p )</td>
</tr>
</tbody>
</table>

Solution:

**Work on this later:**

4. Use the incremental subset construction to convert the NFAs from part 1 to DFAs