EXERCISE 1

Convert the following regular expression to \( \varepsilon \text{-NFA}'s.

a) \((\varepsilon+1)(01)^*(\varepsilon+0)\)

b) \(a^*b^*c^*\)

EXERCISE 2

Convert the following DFA to a regular expression, using the state elimination technique.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s)</td>
<td>(s)</td>
<td>(p)</td>
</tr>
<tr>
<td>(q)</td>
<td>(p)</td>
<td>(s)</td>
</tr>
<tr>
<td>(r)</td>
<td>(r)</td>
<td>(q)</td>
</tr>
<tr>
<td>(s)</td>
<td>(q)</td>
<td>(r)</td>
</tr>
</tbody>
</table>

EXERCISE 3

Repeat exercise 2, eliminating the states in a different order. Then, verify that the following binary strings (accepted by the above DFA) are in the language of the regular expressions obtained here and in exercise 2.

a) 00101100101

b) 010001101110
4a) ε-NFA's for ε+1, ε+0, and (01)* are as follows:

(0+1)*

Composition of the above ε-NFA's yields:

4b) We get the following ε-NFA:

2) The DFA looks as follows:

First, we eliminate the state r

(0+1)*1 for s → q)
and obtain

Second, we eliminate the state \( q \)

and obtain

Third, we eliminate the state \( S \)

and obtain

The regular expression is therefore

\[
(1 + 0(01 + 10^*11)(00 + 10^*10))^* = E
\]
3) Note that, if we eliminate first the state r, but then s before q, we get the following regular expression:

\[ E' = (1 + (00 + 010^*1)(10 + 110^*1)^0)^* \]

3)a) \(00101100101 \in L(E)\):

\[ \underbrace{(0(01)(01)(10010))}_1 \]
\[ \text{from } (00 + 10^*10) \]
\[ \text{from } (01 + 10^*11)^* \]

\[00101100101 \in L(E')\]

\[ \underbrace{(00)(10)(1100101)}_1 \]
\[ \text{from } (10 + 110^*1)^* \]
\[ \text{from } (00 + 010^*1) \]

3)b) \(0(100011)(01)(110) \in L(E)\)

\[ \text{from } (00 + 10^*10) \]
\[ \text{from } (01 + 10^*11)^* \]

\[010001(10)(111)0 \in L(E')\]

\[ \text{from } (10 + 110^*1)^* \]
\[ \text{from } (00 + 010^*1) \]