Problem 1: Basics

Please circle True or False for the following statements:

DFA's can contain epsilon transitions
- True False

colorblue Taken from Notes and Lecture. Since epsilon transitions are optional
using them causes uncertainty or non-determinism as to how to traverse the graph.

All DFA's are NFA's
- True False

Stated in Lecture. DFAs are a subset of NFAs

All NFA's can be converted to Regex
- True False

Said in lecture and said in notes. Every NFA has a Regex equivalence.
You can translate NFA to DFA to regex systematically, but you can go
directly from NFA to regex, although this process is much harder

Problem 2: NFA to DFA

Convert the following NFA to a DFA. Square/circle your final DFA.

We will use the algorithm in the notes/stated in lecture and will show each iteration of the while loop but first, the setup.
The starting state of the DFA is calculated by doing an ε-closure on the starting of the NFA

\[ \{1,2\} \]

We now need to move on each character of the alphabet and perform ε-closure on the result. We will start with the "a" symbol.

\[ \{1,2\} \rightarrow a \rightarrow \{2,3,4\} \]
Now on the character of "b"

Now we do the same thing on the new states we made. Important to note that the node {} can be treated as the garbage state.
Now we just need to mark the final states:
Without the garbage state (optional):
Problem 3: NFA and Regex

(a) What Regex corresponds exactly to the following NFA? (Only circle the regex that best matches the NFA) [2 pts]

\[(abba)^* \quad (ab)^+ \quad (ba)(ba)^* \quad abba \quad (ab|ba)^*\]

We could solve this a few different ways. First by looking at the structure, noticing there is a cycle and 2 distinct paths. Alternatively we could have come up with a bunch of strings the machine accepts that the regular expression does not. For example:

- \((abba)^*\) - The machine accepts the string "ab" but this regex does not
- \((ab)^+\) - The machine accepts the string "ba" but this regex does not
- \((abba)^*\) - The machine accepts the string "ab" but this regex does not
- \((abba)^*\) - The machine accepts the string "ab" but this regex does not

(b) Which of the following strings are accepted by the NFA? (You may choose more than one answer) [2 pts]

- Empty String
- a
- abba
- baabb
- bba

Here are the graph traces of the accepted strings. Since there are infinitely many paths for the others to try, we will leave it up to you to find a path:

- Empty String - \([0 \rightarrow \epsilon \rightarrow 3]\)
- abba - \([0 \rightarrow a \rightarrow 1 \rightarrow b \rightarrow 3 \rightarrow \epsilon \rightarrow 0 \rightarrow b \rightarrow 2 \rightarrow a \rightarrow 3]\)
(c) Convert the following Regex to NFA: \((a \mid b)^*c\)

Let's build the smaller machines and then combine them to the larger machine. Let's begin by doing the "a", "b", and "c" machines.

Now let's combine to get the "a \mid b" machine

Now let's modify to get the "(a \mid b)^*" machine

Now let's concatenate to get the "(a \mid b)^*c" machine

Optimized version (optional):