1. Consider the following NFA $N$ over $\Sigma_N = \{a, b, c, d\}$:

![NFA diagram]

(a) Which of the following words are accepted by $N$ and which are not?

i. $\epsilon$

ii. $a$

iii. $abc$

iv. $abd$

v. $abca$

vi. $abda$

vii. $aabccbbd$

viii. $aabccbbdd$

(b) Calculate $\hat{\delta}_N(S_N, ab)$ (where $S_N$ is the set of start states of NFA $N$). Clearly show each step of the calculation.

2. Consider the following NFA $A$ over $\Sigma_A = \{a, b, c\}$:

![NFA diagram]

(a) Construct a DFA $D(A)$ equivalent to $A$ using the “subset construction”. Clearly show each step of your calculations, e.g. in a transition table.

*Hint:* Some of the 32 states (i.e., the $2^{|Q_A|} = 2^5 = 32$ possible subsets of $Q_A$ that would arise by applying the subset construction blindly to $A$ may be
unreachable. You could therefore adopt a strategy where you only consider reachable states. Simply start from the initial states, \(S_A\), and see what other states arise.

(b) Draw the transition diagram for \(D(A)\), ignoring unreachable states.

3. (a) Give regular expressions defining the following languages over the alphabet \(\Sigma = \{a, b, c\}\):
   i. All words that contain at least one \(b\).
   ii. All words such that all \(a\)'s appear before all \(c\)'s.

(b) Using the formal definition of the meaning of regular expressions, compute the sets denoted by the following regular expressions, simplifying as far as possible. Provide a step-by-step account of your derivations.
   i. \((ab + c + \epsilon)\dd\)
   ii. \((a^*\emptyset + cb)c\)

4. Construct an NFA for the regular expression \((a(b + c))^*\) using the graphical construction from the lecture notes.
   For NFAs\(^1\), it is possible to omit “dead ends”; i.e., states from which no final state possibly can be reached, without changing the language of the automaton. Do this.

---

\(^1\)In the case of DFAs one has to be more careful. “Dead ends” are equivalent since only non-accepting states can be reached from them. As they are equivalent, they can be merged. But if a state in a DFA has preceding state(s), it cannot just be removed as that would leave “dangling transition(s)” from the preceding state(s), which in turn means the result is not a DFA.