1 Exercise 1: Complex Deterministic FSA

The goal of this exercise is to show that the language L described below is a regular language by using the proof by construction in your Deterministic FSA notes. Assume \( \Sigma = \{0, 1\} \).

\[
L = \{ w : w \text{ starts with } 0 \text{ and } |w| \text{ has at least one } 1 \text{ or ends in } 0 \} \]

To do this, follow these steps:

i. Break the language L described by the complex property above into three languages A, B, C described by simpler properties, so that \( L = A \cap (B \cup C) \).

ii. Give the formal definitions and diagrams of the corresponding deterministic FSAs \( M_A, M_B \) and \( M_C \).

iii. Combine \( M_B \) and \( M_C \) into \( M_{B \cup C} \) using the proof in your notes, specifying the formal description and diagram for \( M_{B \cup C} \); then simplify this diagram and the description, if possible.

iv. Combine \( M_A \) and \( M_{B \cup C} \) into \( M_{A \cap (B \cup C)} \) using the procedure in your notes, specifying the formal description and diagram for \( M_{A \cap (B \cup C)} \); then simplify this diagram and the description, if possible.

2 Exercise 2: Complex Deterministic FSA.

Take language J, with alphabet \( \Sigma = \{0, 1\} \), where the English disjunction \( \text{either}...\text{or} \) is taken to express exclusive “or”:

\[
J = \{ w : \text{either } w \text{ has exactly three } 1s \text{ or } w \text{ ends in the sub-string } 10 \}
\]

Break language J into two languages D and E described by two simpler properties, and give the formal definition and diagram for the corresponding deterministic FSA \( M_D \) and \( M_E \). Then, inspired by the proof in your lecture notes, combine \( M_D \) and \( M_E \) into \( M_{D \cap E} \), providing the formal description and diagram for it. Simplify, if possible.
3 Exercise 3: Deterministic vs Non-Deterministic FSA.

For each of the following languages, construct a deterministic FSA that is most economical and a non-deterministic FSA with the indicated characteristics. Assume $\Sigma = \{a, b\}$. (You don’t need to use the proof for the operations $\cup$ and $\cap$ here.)

i. $\{w : w \text{ contains no } a \text{ and ends in } bb\}$. Non-det FSA with three states, one of which is final.

ii. $\{a, ba, aba, baba\}$. Non-det FSA with five states, one of which is final.

iii. $\{a, ba, aba, baba, ababa, \ldots\} =$

$\{w : w \text{ ends in } a \text{ and does not contain the sub-strings } aa \text{ or } bb \}$. Non-det FSA with three states, one of which is final, and as few transition arrows as possible.

4 Exercise 4: Deterministic and Non-Deterministic FSA for NatLg.

Take the subset of English described by the following regular expression:

$\text{Eng}^{-}: \text{Det} (\text{very}^* \text{ Adj}^* \text{ Noun} \text{ V}_{\text{intr}}$

This language, which we will call Eng$^{-}$, includes the sentences in (1) but not the sentences in (2):

(1)  a. The boy runs.
     b. The nice boy runs.
     c. The tall nice boy runs.
     d. The happy tall nice boy runs.
     e. ...
     f. The very nice boy runs.
     g. The very very nice boy runs.
     h. The very very very nice boy runs.
     i. ...
     j. The tall very nice boy runs.
     k. The very tall nice boy runs.
     l. The very tall very nice boy runs.
     m. The very tall very very nice boy runs.
     n. ...

2
(2) a. The boy nice runs.
    b. The nice very boy runs.
    c. Very nice the boy runs.
    d. Etc.

Design a deterministic FSA diagram and a non-deterministic FSA diagram that accept the language $\text{Eng}^-$. $\Sigma=\{\text{Det, very, Adj, Noun, V}_{\text{intr}}\}$. 