For this assignment, a finite state automaton is defined as consisting of a finite set of states, one of which is designated as the state the machine starts in and at least one other which is designated an end state, a finite set of transitions between two, not necessarily distinct, states, and for each transition, a “Write X” operation that produces a single symbol, X. A finite state automaton is said to “generate a string \( \sigma \)” if \( \sigma \) is a series of symbols that the automaton can produce by going from its start state to one of its end states.

Write a finite state automaton that generates the infinite set of strings in the series in (A).

\[
(A) \begin{pmatrix}
\text{The man danced.} \\
\text{The man danced on the desk.} \\
\text{The man danced on the desk behind the table.} \\
\text{The man danced on the desk behind the table under the rug.} \\
\text{The man danced on the desk behind the table under the rug beside the cabriolet.} \\
\vdots
\end{pmatrix}
\]

Your automaton should generate strings of symbols that correspond to the word types that are illustrated in (A). That is, rather than generating the particular strings in (A), it should generate a string of word types that each of the words in (A) correspond to.

Note:
the belongs to the Determiner class
does not produce any string that isn't in the series. So it shouldn't produce any of the ungrammatical sentences in (B), for example.

\(\text{(B) a. * The stood on the desk.} \)
\(\text{b. * The man stood on.} \)
\(\text{c. * The man stood the desk.} \)
Write a finite state automaton that generates the infinite set of strings in the series in (C).

\[
\begin{align*}
(C) & \\
& \begin{cases}
\text{The man danced.} \\
\text{The man on the desk danced.} \\
\text{The man on the desk behind the table danced.} \\
\text{The man on the desk behind the table under the rug danced.} \\
\text{The man on the desk behind the table under the rug beside the cabriolet danced.} \\
\vdots
\end{cases}
\end{align*}
\]

Make sure that your automaton does not produce any string that is not in (C). So it shouldn't produce the strings in (D), for example.

\[
\begin{align*}
(D) & \\
a. & \ast \text{ The on the desk danced.} \\
b. & \ast \text{ The man on the behind the table danced.} \\
c. & \ast \text{ The man on the desk behind the danced.}
\end{align*}
\]

Consider now the series that would be produced by combining the two in (A) and (C). So, that might be expressed like this:

\[
\begin{align*}
(E) & \\
& \begin{cases}
\text{The man danced.} \\
\text{The man danced on the desk.} \\
\text{The man danced on the desk behind the table.} \\
\vdots \\
\text{The man on the desk danced.} \\
\text{The man on the desk danced on the desk.} \\
\text{The man on the desk danced on the desk behind the table.} \\
\vdots \\
\text{The man on the desk behind the table danced.} \\
\text{The man on the desk behind the table danced on the desk.} \\
\text{The man on the desk behind the table danced on the desk behind the table.} \\
\vdots
\end{cases}
\end{align*}
\]

Write a finite state automaton that generates the set of strings in this series.