Implementation:

I implemented the program in javascript because I'm more familiar with this language. I created a class containing the **states**, **alphabet**, **transitions**, **initial state** and **final state**, all of them being arrays of strings, only the **transitions** are an array of classes which contain the **initialState**, **element** and **finalState**.

After reading from the file and initializing the "***Transition***" class, a function checks if the given sequence is viable using the defined automaton's rules.

Reading from the file:

- the elements in the file are separated by a '**,**' (comma)
- the transitions are separated by '**;**' (semicolon)

Printing the Transition is rather easy, because it prints the whole class in a readable format.

The function **checkSequence** verifies if the given sequence is **viable**. It goes through all the transitions, step by step and verifies at the end if the final step of the ***Transition*** class is the same as the one found by the function.

The sequence which needs to be verified is given in the **first line** of the file you are reading from.

File:

language = letter | digit
letter = "A" | "B" | … | "Z" | "a" | "b" | … | "z"
digit = "0" | nonZero
nonZero = "1" | "2" | … | "9"
sign = + | -
state = letter
transition = state "," language "," state
line1 = {alphanumeric ","}
line2 = {state ","}
line3 = {alphanumeric ","}
line4 = {transition ","}
line5 = {state}
line6 = {state ","}
file= line1\nline2\nline3\nline4\nline5\nline6