

Requirements:

You are required to build a command line tool that converts a regular expression into its corresponding NFA using Thompson's construction algorithm. You must implement the algorithm yourself without using any libraries.

Your tool should run like this for example: convert "[A-Za-z]+[0-9]*"

Input format:

The input to the tool is simply a regular expression in text form. The expression maybe invalid so you need to validate it before transforming it first. Example of an invalid regular expression: A| , AB** , etc.

Regex Rules:

- Concatenation: AB
- ORing: A|B or A+B
- Repetition (0 or more): A*

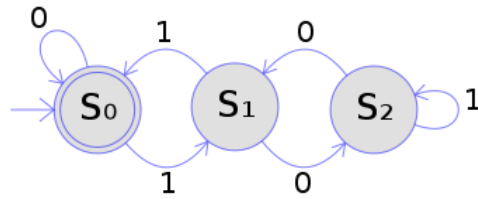
Output format:

You are required to output a JSON file representing the FSM states and transition using the following format:

```
{
  "startingState": "stateA"
  "stateA": {
    isTerminatingState: false,
    "inputCharacterA": "stateB",
    "inputCharacterB": "stateA"
  },
  "stateB": {
    isTerminatingState: true,
    "inputCharacterA": "stateB",
    "inputCharacterB": "stateB"
  }
}
```

For example

The following NFA



Would be corresponding to the following JSON:

```
{
  "startingState": "S0"
  "S0": {
    isTerminatingState: true,
    "0": "S0",
    "1": "S1"
  },
  "S1": {
    isTerminatingState: false,
    "0": "S2",
    "1": "S0"
  },
  "S2": {
    isTerminatingState: true,
    "0": "S1",
    "1": "S2"
  }
}
```

[Optional] You are also required to output an image containing the graph of the NFA, using any graphics library of your choice. You should distinguish between accepting and non-accepting states.