CSCE A351: Automata
Spring 2019. Homework Assignment 1
Due: 02/17/2019 11:59PM AKST
Individual assignment

For this assignment, you are supposed to hand in:

- code implementing the functionality specified below, written in any programming language,
- a short write-up explaining your code, your testing strategy and the design choices you made.

In a programming language that allows for representation of high-order data types like sets, lists and (anonymous) functions, it is easy to model a Deterministic Finite Automaton (DFA) just like we did in class, with a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$:

```
def dfa_accept(M, w):
    """ Runs the DFA M = (Q, Sigma, delta, q0, F) on the string w.

        Returns True if all characters in w are in Sigma
        and the DFA accepts the word. Also checks if
        all states returned by delta are in Q.

        Returns False otherwise.
    """
    (Q, Sigma, delta, q0, F) = M
    q = q0
    if q not in Q:
        return False
    s = w
    while s != "":
        a = s[0]
        s = s[1:]
        if a not in Sigma:
            return False
        q = delta(q, a)
        if q not in Q:
            return False
    if q in F:
        return True
    return False

def delta_odd(q, a):
    if q == 0:
        if a == '0':
            return 0
```

```
            elif a == '1':
                return 1
            else:
                return 2
        elif q == 1:
            if a == '0':
                return 1
            elif a == '1':
                return 0
            else:
                return 2
        else:
            return 2
print("Odd?␣",
      dfa_accept(({ 0, 1 }, { '0', '1' }, delta_odd, 0, { 0 }),
                 "0001010101011"))
print("Odd?␣",
      dfa_accept(({ 0, 1 }, { '0', '1' }, delta_odd, 0, { 0 }),
                 "0001010101010"))
```

For this assignment, you must hand in valid code for the following problems:

1. A functional model for DFAs, as shown in the example above, in the form of a function taking a DFA and a string in argument and returning a boolean indicating whether the string is in the language of the DFA[1].

2. A function (or method) that takes a model of a Nondeterministic Finite Automaton (NFA) in argument and returns a valid DFA, which the function above is able to execute. NEEDS TO RETURN A FUNCTION - DOABLE IN PYTHON

3. A function (or method) that takes two models of NFAs in argument and returns an NFA for the union of their languages.

4. A function (or method) that takes two models of NFAs in argument and returns an NFA for the concatenation of their languages.

5. A function (or method) that takes a models of an NFA in argument and returns an NFA for the star of its languages.

6. A function (or method) that takes a string describing a regular expression in argument (`'|'` denoting the union, `'*'` denoting the star, `'('` and `')'` grouping the operations, any other character denoting itself) and returns an NFA for that regular expression.

7. A function taking a string denoting a regular expression and another string in argument, returning a boolean indicating whether the second string is in the language described by the regular expression in the first string.

As a matter of course, you must write your code from scratch and not use any libraries supporting the manipulation of Finite Automata or Regular Expressions.

[1] If you use Python as a programming language, you may return the example code as-is for this question.