Integrative practice
Part 2
Computational mathematics

Professor: Gilberto Huesca Juárez

Name: _____

Student ID: _____

Read the assignment before start
In Canvas, write the names and student IDs of all the team members, and the group
This activity is in pairs
Make the proposed activities and at the end, if there are programs to deliver, upload the files packaged within a ZIP file.
You may use java, python, or C++.
The name of the main file must be Main.xxx where xxx is the extension for your program file.
The name of the ZIP file must be Exxxxproject2.zip where xxxxx is your student ID. For example, if your student ID is 123456, the file name must be E123456project2.zip
The programs must be properly commented (for example, if you do it in Java, your program should use JavaDoc).
LATE DELIVERY NOT APPLICABLE.
This activity counts as 10% of the final grade.

Make a program that reads, from a file, the elements that define an NDFA and builds the equivalent DFA.

The transition table will be defined in a txt file. The file shall be defined as follows:

- The first line indicates the set of states of the automata separated by commas.
- The second line indicates the alphabet symbols separated by commas
- The third line indicates the initial state
- The fourth line indicates the set of final states separated by commas.
- The following lines indicate the evaluation of the transition function with the elements of the alphabet in the following format:

state, symbol = > state1, state2, ... stateN

Example, the following line

q0, a = > q0 , q1, q5

indicates that the NDFA processes the following: $\delta(q0, a) = \{q0, q1, q5\}$

It is not necessary that all transitions are specified in this file. An evaluation may not appear indicating that the result of that evaluation is the empty set

Finally, the program should print the set of states in the DFA and the equivalence of each state to the states of the original automaton. In addition, you must print the transition table of the DFA.

The test files are only examples. Your program should work for any file that follows the conventions for the **structure** denoted above. Use functions as “split” to process the lines in the file instead of read char by char. This will make your program more general. Your program will be tested with files different to those used as examples.

Your program should have at least three methods:

1. Simple transition function. It should receive a state and a character. It should return a set of states
2. Union. It will receive two sets of states and return their union as a single set.
3. A method for the transformation process.

Your program should show a correct use of data structures. At least, the transition table of the NDFA should be represented as

Dictionary {key: state, value: Dictionary {key: char or lambda, value: list_of_states}}

Your program should correctly apply the conceptual elements of this course.

Do not worry about validating the values in the input file. Suppose they were built correctly.