Computer Science 716 **Project**
2021 **Time limit: To be handed in 4 June 2021, at 14h00**

Plagiarism declaration

By handing in this practical project on git, I declare that I implemented the code myself. I did not download any code from the internet, and I did not get assistance from any other person in the implementation of the code. I realise that plagiarism checks will be run amongst all submissions from the class, as well as against possible solutions on the internet. I realise that if a match is found, I will not receive any marks for this assignment and that legal steps will follow. No libraries may be used – all code must be developed from scratch.

Specification

The project must be implemented in C.

Extend your project from the CS345 project of 2020. Feel free to reuse any code from that project freely. You must implement the conversion from an XNFA to a DFA. That is, given a textual description of an XNFA in an input file, you must output the equivalent DFA, in a similar textual format.

For the textual description, you must specify the number of states in the XNFA (n) – assume that the states are always numbered 0 tot $n-1$. Then specify the alphabet symbols as a set of characters, the set of start states as a set of integers, the set of final states as a set of integers, and then the transitions: state the number of edges, and list each edge.

In order to handle the epsilon transition, please assume that all the alphabet characters are lowercase characters between a and z inclusively, and use the symbol @ as the epsilon symbol.

When you print the resulting DFA, please print the number of states, the single start state, the set of final states, and the transitions (see below). It is important to note that your state names will now be characters, as each state is a combination of states from the XNFA. For example, a state name in the DFA might be 023, if it was created as a combination of the XNFA states 0, 2 and 3.

Instructions

1. Create a C file called <`studentnumber`>`.gcc`, which contains your solution.
2. You must submit your C file on git. You may submit multiple times.
3. Your program must run from the command line. The XNFA description must be read from a text file.
4. Your output for a given input file will be a textual description of a DFA, as described above.
5. All debugging output must be removed before your final submission.
6. Please note that you must do error checking on the input format, and indicate incorrect input in the text file.

Input format

The text file must have the format as given below.

```
"n=" <integer>
"sigma={" (<char>",")* <char> "}"
"start={" (<integer>",")* <integer> "}"
"final={" (<integer>",")* <integer> "}"
<number of edges>
list of edges given as <node> <char> <node>
```

Output format

The output DFA should have the following format:

```
"n=" <integer>
"start=" <charstring>
"final={" (<charstring>",")* <charstring> "}"
list of edges given as <node> <char> <node>
```