

CSC 212—Algorithms and Complexity

Create a Java program to emulate any DFA.

Practical 1 Term 4

7 September 2015

Due by 17h40 on 7 September 2015.

Submit your own code today using `cd; make submit`. Consultation is permitted.

A deterministic finite automaton (DFA) is a very simple type of machine. It has a *control mechanism*, a *finite number of states*, and an *input stream* with a *read-only head* called the *input* that reads a single character at a time from a *finite* stream. DFAs have a unique *start* state where computation starts and a set of *final* or *accept* states that are also known as *halt* states. Computation ceases when a *halt* state is reached—with no more input to process. The machine is then said to *accept* its input. If the string is depleted and the machine is NOT in a halt state the DFA is said to *reject* its input.

In this practical you must program a ‘machine’ that reads in one or more DFAs and several inputs and one-by-one processes these inputs and prints out the input on a single line followed by the word **accept** if the machine accepts the string or **reject** when the string is rejected. Each DFA is represented by giving its alphabet, followed by the DFA represented as a state table. The inputs are then given, each on one line.

Since we will mark your code automatically your input data and output must conform strictly to the specifications given below.

The specifications of the automaton are as follows:

1. Lines starting with a ‘#’ are comments that are ignored.
2. Lines starting with a ‘D’ list the allowable symbols. This list is terminated by the word ‘**finals**’ that heads the column that marks states that are final. Any and all the states may be final.
3. Each line ends with a ‘.’. Extra but redundant white space may appear anywhere.
4. Each state is named in the first column. The target state corresponding to each input is given in subsequent columns. States are given by a preceding ‘q’ followed by the number of the state. The *start* state is always state ‘q1’. There may be 99999 states.
5. There is one column for each input. The last column is either empty if the corresponding state is not a final or has an ‘*’ if it is a final.
6. The data can be interspersed with comments.
7. The final line of each automaton is either a blank line or an end-of-file.
8. There may be no data for a DFA, or there might be no DFAs at all.
9. There is a sample data file in the `../notes/ds/` directory in the file called `dfadata.text`.

Please work in your `14practical` directory and submit as usual before leaving the lab.
