## Overview

The purpose of this assignment is to broaden your understanding of the *acceptance relation* for *deterministic finite automata*. Recall that a string ω is accepted by a DFA M, if and only if there exists a path from the start state of M to some final state in M labeled by the constituent characters of ω. In this assignment you will implement a program in Java which will accepts as its input the name of a file containing a list of strings and will determine whether or not they are accepted by a specific automaton.

Specification

Define a program called `DFACheckBasic` which will accept as its input the name of a file containing a collection of strings and determines which of them are accepted by the *deterministic finite automaton*, M, shown in the figure below.
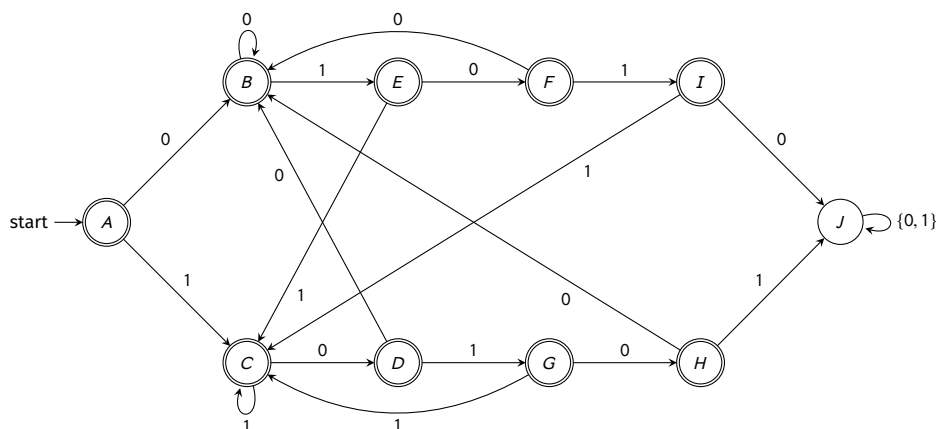
Input File Specification

For this assignment you may assume that the input file is formatted with exactly one string per line and that the strings are guaranteed to be formed from characters of the input alphabet $\{0, 1\}$. As an example, the following is one possible input file[1]:

```
1   100100
2   111
3   00100100
4   10101
5   01010101
```

---

[1]Line numbers are included for reference only.

Usage

The program `DFACheckBasic` is intended to be run via the following command-line invocation:

```
1   $ java DFACheckBasic input.txt
```

If no input file is specified or if the input file is not found, an appropriate error message must be display. For example:

```
1   $ java DFACheckBasic
2   DFACheckBasic: no input file specified
3
4   $ java DFACheckBasic no-such-file.txt
5   DFACheckBasic: the file 'no-such-file.txt' could not be opened
```

If a valid file is specified as input, then for every string in the file you should output both the string and the word `accepted` if M accepts it and `rejected` otherwise, with only one string's results given per line. Using the file `input.txt` mentioned above, your program should behave as follows[2]:

```
1   $ java DFACheckBasic input.txt
2   100100 accepted
3   111 accepted
4   00100100 accepted
5   10101 rejected
6   01010101 rejected
```

Compilation Notes

Your program must compile using the following command-line invocation:

```
1   $ javac *.java
```

Do not submit any IDE specific files and in the interest of simplicty do not use packages for this assignment. If your program does not compile using the command mentioned above you will receive no credit.

---

[2]Note that in the example output there is exactly one space between the string and its results. As before, line numbers are included for reference only.