Project 1
Due Date: 2/17, 1:00pm NJ local time

Be sure to read this entire document before starting the assignment.

Academic Integrity

Any student caught cheating on any assignment will be reported to the dean of students. Cheating includes, but is not limited to, getting solutions (including code) from or giving solutions to someone else. You may discuss the assignment with others, but ultimately you must do and turn in your own work.

1 Overview

We define the language L to consist of strings that represent certain web-page addresses (specified below). For this assignment you are to design a DFA to recognize L and write a program that implements your DFA.

2 The Language L

To precisely specify L, first define $\Gamma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$ as the set of lower-case Roman letters, and let $\Sigma = \Gamma \cup \{\mathtt{.}\}$; i.e., Σ is the set consisting of all the lower-case Roman letters and the dot (or period). Define the following sets:

- $S_1 = \{\mathtt{www.}\}$

- $S_2 = \Gamma\Gamma^*$, which consists of strings over Γ of positive length

- $S_3 = \{\mathtt{.com}\} \cup \{\mathtt{.co.jp}\}$

Then we define the following sets of strings over Σ:

- $L_1 = S_1 S_2 S_3$

- $L_2 = S_2 S_3$

- $L = L_1 \cup L_2$

Strings in L_1 are (a subset of) web addresses that have three parts, where the first part is "`www.`", the second part is a positive-length string of lower-case Roman letters, and the third part is "`.com`" or "`.co.jp`". (The suffix "`.co.jp`" is used in some web addresses in Japan.) Strings in L_2 are (a subset of) web addresses that have only the last two parts. For example, the strings `www.abcdef.com`, `www.com.com`, `www.co.com`, `www.abcdef.co.jp`, `www.com.co.jp`, and `www.co.co.jp` belong to L_1, and the strings `abcdef.co.jp`, `www.co.jp`, `abcdef.co.jp`, and `www.co.jp` belong to L_2.

The specification of L does not include all valid web addresses since we included several restrictions to simplify the assignment. For example, only lower-case Roman letters and dots are allowed, strings in L must end with `.com` or `.co.jp`, etc.

3 DFA for L

First construct a DFA $M = (Q, \Sigma, \delta, q_1, F)$ that recognizes L. The DFA must satisfy each of the following properties:

- The DFA must be defined with the above alphabet Σ. Your DFA does not have to handle symbols that are not in Σ.

- The states in the DFA must be labeled $q_1, q_2, q_3, \ldots, q_n$, where q_1 is the start state and n is the number of states in the DFA. (It is also acceptable for the states to be labeled $q_0, q_1, \ldots, q_{n-1}$, with q_0 the start state.)

You will lose points if your DFA is overly complicated. To simplify your DFA drawing, omit any edges going from a state $q \notin F$ to a "trap state" (i.e., a non-accepting state from which the DFA can never leave). All other edges must be included in your drawing. Clearly identify which state is the trap state in the drawing of your DFA. Also, to simplify your drawing, you should define $\Gamma_\ell = \Gamma - \{\ell\}$ for any symbol $\ell \in \Gamma$; i.e., Γ_ℓ is the set of all lower-case Roman letters except for ℓ. For example, $\Gamma_{\tt w} = \{{\tt a}, {\tt b}, {\tt c}, \ldots, {\tt v}, {\tt x}, {\tt y}, {\tt z}\}$, so your DFA might include something like the following:
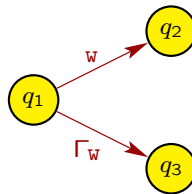
Thus, if the DFA is currently in state q_1, then it moves to q_2 on reading `w`, and it moves to state q_3 on reading any other lower-case Roman letter.

4 Program Specifications

You must write your program in either C, C++, or Java. All input/output must be through standard input/output, and your program is to work as follows:

2

1. Your program asks the user if s/he wants to enter a string. The user then enters "y" for "yes", or "n" for "no".

   - If the user enters "n", then the program terminates.
   - If the user enters "y", then the user is prompted to enter a string over Σ.

2. If the user chooses to input a string, your program then reads in the string. After reading in the string, your program prints it. Then your program processes the **entire** string on the DFA, one character at a time, in the following manner.

   - Your program must begin in the start state of the DFA and print out the name of that state (q_1 or q_0).
   - After each character from the string is processed on the DFA, your program must print out the character and the name of the current state of the DFA. Even if your DFA is in a trap state, your program must do this for each character in the string until it reaches the end of the string.

   To simplify your program, you should check the ASCII code of each character of the string and process on the DFA accordingly.

3. After processing the entire string on the DFA, your program must indicate if the string is accepted or rejected based on the state in which the DFA ended. Your program then should return to step 1.

5 Test Cases

Test your program on each of the following input strings:

1. `www.commercial.com`

2. `www.co.com`

3. `www.x.com`

4. `njit.com`

5. `wwww.stanford.com`

6. `www.com.com`

7. `ww.com.com`

8. `co.com`

9. `www.www.com.com`

10. `ww.com`

11. `www.com`

12. `wwd.com`

13. `wwwb.com`

14. `www..com`

15. `www.abcd.co`

16. `www.company.co.jp`

17. `company.co.jp`

18. `www.co.co.jp`

19. `www.com.co.jp`

20. `www.co.jp`

You must create an output file containing the output from each test case in the order above.

6   Deliverables

The due date/time for the assignment is 2/17, 1:00pm NJ local time. You must submit all of the following **through moodle** by the due date/time:

1. A Microsoft Word document stating, "I certify that I have not violated the University Code on Academic Integrity", followed by your first and last name and NJIT student ID. If you do not include this pledge, then you will receive a 0 on the assignment. Anyone caught violating the University Code on Academic Integrity will be immediately reported to the dean of students.

2. A drawing of the DFA for L that your program implements. This format of the file must be either Microsoft Word, pdf, or jpeg. The file must be smaller than 5MB in size.

3. A **single file** of your source code, of type `.c`, `.cpp`, or `.java`. Only submit the source code; do not include any class files. You must name your file

<div align="center">

`p1_15s_ucid.ext`

</div>

where `ucid` is replaced by your NJIT UCID and `.ext` is the appropriate file extension for the programming language you used, e.g., `.java`. The first few lines of your source code must be comments that include your full name and UCID.

4. A **single file** containing the output from running your program on all of the test cases, in the order given in Section 5 of this document. The output file must be either .txt or in Microsoft Word.

The files **must not be compressed**. You will not receive any credit if you do not complete all of the above. **Late projects will be penalized at a rate of 25 points per 24-hour period.** A project is considered to be late if all of the above are not completed by the due date/time, and the 25-points-per-day penalty will continue to accumulate until all of the above have been completed. Any submissions completed more than 72 hours after the due date/time will receive no credit.

7 Grading Criteria

The criteria for grading are as follows:

- correctness of your DFA for L (40 points),

- the program works according to the specifications given in Section 4, matches your DFA for L, and follows the directions in Section 6 (10 points),

- your program is properly documented (i.e., comments) (10 points),

- your output is correct for the test cases (40 points).

Your grade will mainly be determined by examining the source code, the drawing of the DFA, and the output that you turn in; the source code will only be tested if there are questions about your code.

To receive any credit for this assignment, you must turn in a drawing of your DFA for L and a *minimally working* program. For a program to be minimally working, it must satisfy all of the following:

- compile without syntax errors;

- properly accept strings in L_1 that end in ".com"; and

- implement the drawing of your DFA for L.

If you do not hand in a minimally working program, then you will receive a 0 for the assignment *and* your grade in the course will be lowered by one step, e.g., from B to C+, or from C to D.

8 Hints

To design a DFA for L, start by drawing a DFA to handle only L_1 and end in ".com", which includes the first three test cases. Initially include only the transitions that will eventually lead to acceptance. Then modify your DFA to handle the rest of L and the other test cases.