40-414 – Compiler Design, Fall 1400, Sharif University of Technology

Writing Assignment 1

Released: Tuesday, 06/07/1400

Due: Tuesday, 20/07/1400 at 11:59pm

This assignment covers lectures 1 to 3 about regular languages, finite automata and lexical analysis. You can discuss and work out this assignment with other students. However, your write-up must be your own individual work. Solutions to all questions (except for question 3) must be written by hand. A PDF format of the solution (**with your full name and student number**) must be uploaded to the course page in Quera **before 11:59 PM, Tuesday 20/07/1400** (https://quera.ir/course/9134/). Submissions with more than 50 hours delay will not be graded. Submissions with less than 50 hours delay will be penalized by the following rule:

Penalized mark = M * (100 - 2 * D) / 100

Where M = the mark achieved from your solution and D[1] is the number of hours passed the deadline.

1. Consider the following set of token types:

Token Type	**Description**
NUM	Any string matching: [**0-9**]$^+$
ID	Any string matching: [**A-Za-z**][**A-Za-z0-9**]*
KEYWORD	**if, else, void, int, repeat, break, until, return**
SYMBOL	**; : , [ ] ( ) { } + - * = < ==**
COMMENT	Any string between a **/*** and a ***/** OR any string after a **//** and before a \n or **EOF**
WHITESPACE	blank (ASCII 32), \n (ASCII 10), \r (ASCII 13), \t (ASCII 9), \v (ASCII 11), \f (ASCII 12)

- Draw appropriate DFAs (i.e., similar to the DFAs in pages 56-58 in Lecture note 3) for recognizing these tokens. Note that ID and KEYWORD are recognized by the same DFA, which is almost identical to the one on Page 57 of Lecture note 3!
- In each of these DFAs, specify exactly what characters should be considered as compatible with the 'other' label. Note label 'other' in different DFAs are not necessarily referring to the same set of symbols.
- Then combine these DFAs into a single DFA. The resultant DFA (provided that the DFA is correct and complete) can then be used as a flowchart for implementing a scanner in Programming Assignment 1, to be later released.

[1] You may have a total of 12 hours delay free of penalty in your writing assignments. If you have less than 12 hours delay in submitting WA1, remaining free of penalty hours will be deducted from your delay hours in WA2. You can use (all or part of) this allowance for WA3, only if you have not used all the 12 hour allowance for WA1 and/or WA2.

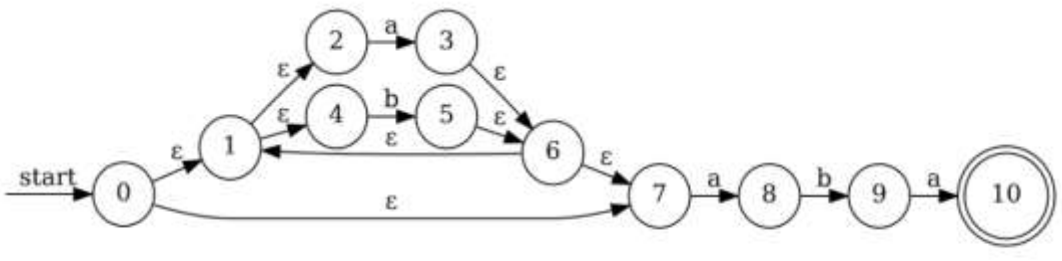
2. In a programming language, comments start with a '/*' and end with '*/'. Comments can contain any character strings except for '*/' unless it is surrounded by two double quote characters ("). The followings are example of valid and invalid comments:

Valid Comments	Invalid Comments
/*abc"*/"de*/	/*abc*/de*/
/**/	/*ab"*/ " */
/*21"*/""*/"43*/	/*21"*/"*/"43*/
/*a2"*/	/*a2"*

   a. Draw a dfa with minimum possible number of states to recognize these comments. Note that your dfa should only accept the input only if the whole input is a comment. In other words, when the input is ended the dfa should be in an accepting (final) state.

   b. Write a regular expression with minimum number of characters[2] that defines the set of comments in the above language[3].

3. Convert the following NFA to an equivalent DFA with minimum states.



How to deliver: This question actually has a judge in Quera. All you have to do is to clone this repository from github. You can download it from here if you don't want to clone from githhub. Then open path /html/index.html from the repository in your browser and draw your dfa there. Get json dump of you dfa and rename it to **dfa.json** and upload it in Quera.

[2] Left and right parentheses are not counted when we consider optimal solution.

[3] Note that in all questions regarding "regular expressions", you can only use the operators mentioned on page 2 of lecture note 3.

4. Design an optimal (with minimum number of characters[4]) regular expression that accepts the following good samples and rejects bad samples:

Good samples:	00, 1010, 000010, 1100, 1000
Bad samples:	11, 1001, 0110, 10110, 0100

5. For each of the following languages, give two **arbitrary** strings that are members and two **smallest non-empty** strings that are not members (a total of four strings for each part) of these languages. Assume the alphabet is $\Sigma = \{a, b\}$ in all parts. String **a** is considered smaller than string **b** if **a** is shorter in length; or if they have equal length and **a** appears in the dictionary before **b**.

   a. $a(ba)^*b$
   b. $a^* \cup b^*$
   c. $(\varepsilon \cup a)b$
   d. $(a \cup ba \cup bb)\ \Sigma^*$

Ghassem-Sani, Gholamreza

06/07/1400

[4] Again, left and right parentheses are not counted when we consider optimal solution.