CS131 Compilers: Writing Assignment 1
Due Monday, September 30, 2019 at 22:00

Name - ID

This assignment asks you to prepare written answers to questions on regular languages, finite automata, and lexical analysis. Each of the questions has a short answer. You may discuss this assignment with other students and work on the problems together. However, your write-up should be your own individual work and you should indicate in your submission who you worked with, if applicable. Written assignments are turned in at the start of lecture. You should use the Latex template provided at the course web site to write your solution and use the *tikz* package to draw automata.

<center>I worked with: (zhanglw,2018533113)</center>

1. ($2 \times 3 = 6$ **pts**) For each of the follow prompts, write any non-empty sentence:

   (a) Name one reason why you would like to learn in this class.

   $$For\ a\ wider\ knowledge\ in\ compiler\ and\ meeting\ great\ people\ in\ our\ school.$$

   (b) Write a question you would like the professor to answer on any topic, from personal opinions to the class material.

   $$Is\ that\ possible\ to\ invent\ EEG\ compiler?$$

   (c) What do you expect from this class.

   $$A+$$

2. ($2 \times 4 = 8$ **pts**) Write regular expressions for the following languages over the alphabet $\Sigma = \{0, 1\}$:

   (a) L_1: The set of all finite strings containing at least three $1's$.

   $$\Sigma^*1\Sigma^*1\Sigma^*1\Sigma^*$$

   (b) L_2: The set of all finite strings containing at most two $0's$.

   $$1^* \cup 1^*01^* \cup 1^*01^*01^*$$

   (c) L_3: The set of all finite strings containing at most two $0's$ and at least three $1's$.

   $$111^+ \cup 111^+0 \cup 11^+01^+ \cup 1^+011^+ \cup 0111^+ \cup 11^+001^+ \cup 1^+0011^+ \cup 00111^+ \cup$$

   $$01^+011^+ \cup 011^+01^+ \cup 0111^+0 \cup 1^+01^+01^+ \cup 1^+011^+0 \cup 11^+01^+0 \cup 111^+00$$
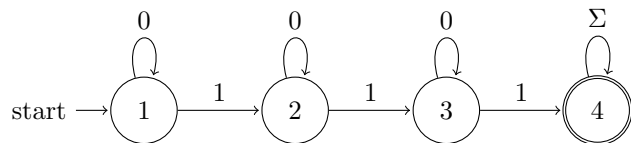
   (d) L_4: The set of all finite strings containing at least three $1's$, but no two $1's$ appear consecutively.
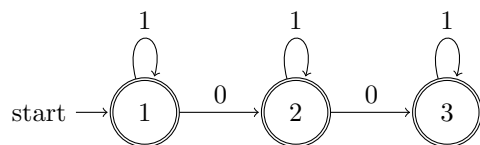
   $$0^*10^+1(0^+1)^+0^*$$

   This example illustrates that regular languages are closed under intersection. Note that $L_3 = L_1 \cap L_2$.

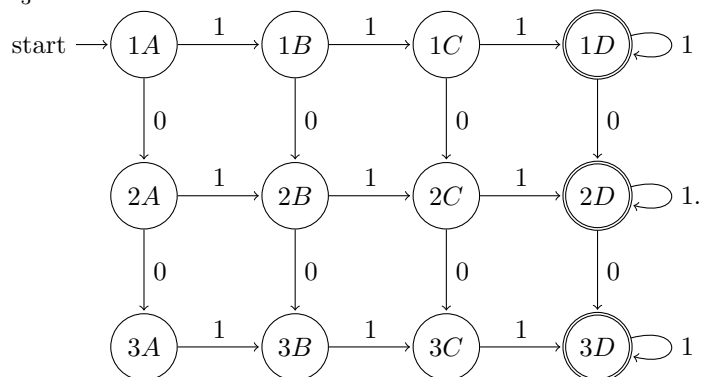3. $(2 \times 4 = 8 \text{ pts})$ Draw DFA's for each of the languages L_1, L_2, L_3 and L_4 from Question 2.
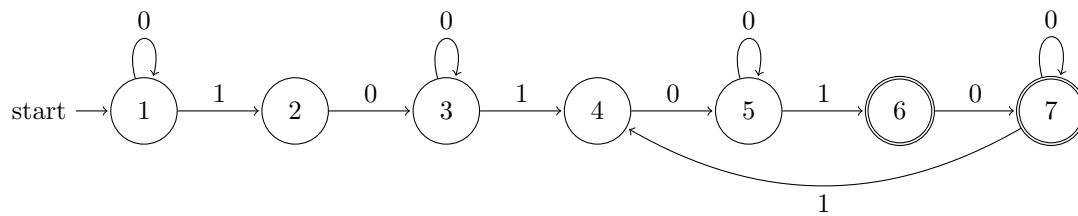
(a) L_1.

start $\rightarrow$ (1) $\xrightarrow{1}$ (2) $\xrightarrow{1}$ (3) $\xrightarrow{1}$ ((4))

with self-loops: 0 on 1, 0 on 2, 0 on 3, Σ on 4.

(b) L_2.

start $\rightarrow$ ((1)) $\xrightarrow{0}$ ((2)) $\xrightarrow{0}$ (3)

with self-loops: 1 on 1, 1 on 2, 1 on 3.

(c) L_3.

start $\rightarrow$ (1A) $\xrightarrow{1}$ (1B) $\xrightarrow{1}$ (1C) $\xrightarrow{1}$ ((1D)) with self-loop 1 on 1D

(1A) $\xrightarrow{0}$ (2A) $\xrightarrow{0}$ (3A)
(1B) $\xrightarrow{0}$ (2B) $\xrightarrow{0}$ (3B)
(1C) $\xrightarrow{0}$ (2C) $\xrightarrow{0}$ (3C)
(1D) $\xrightarrow{0}$ ((2D)) $\xrightarrow{0}$ ((3D))

(2A) $\xrightarrow{1}$ (2B) $\xrightarrow{1}$ (2C) $\xrightarrow{1}$ ((2D)) with self-loop $1.$ on 2D

(3A) $\xrightarrow{1}$ (3B) $\xrightarrow{1}$ (3C) $\xrightarrow{1}$ ((3D)) with self-loop 1 on 3D

(d) L_4.

start $\rightarrow$ (1) $\xrightarrow{1}$ (2) $\xrightarrow{0}$ (3) $\xrightarrow{1}$ (4) $\xrightarrow{0}$ (5) $\xrightarrow{1}$ ((6)) $\xrightarrow{0}$ ((7))

with self-loops: 0 on 1, 0 on 3, 0 on 5, 0 on 7; and an arrow $\xrightarrow{1}$ from 7 back to 4.

2

4. ($5 \times 3 = 15$ **pts**) Using the techniques covered in class, transform the following NFAs with ϵ-transitions over the given alphabet Σ into DFAs. Note that a DFA must have a transition defined for every state and symbol pair, whereas a NFA need not. You must take this fact into account for your transformations. Hint: Is there a subset of states the NFA transitions to when fed a symbol for which the set of current states has no explicit transition?
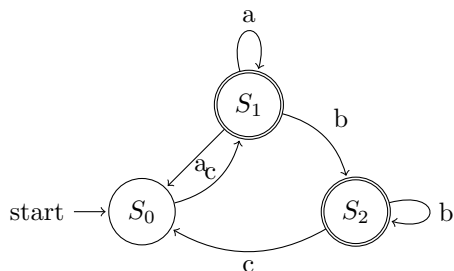
(a) Original NFA, $\Sigma = \{a, b, c\}$:

DFA:
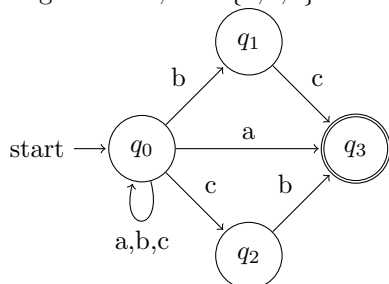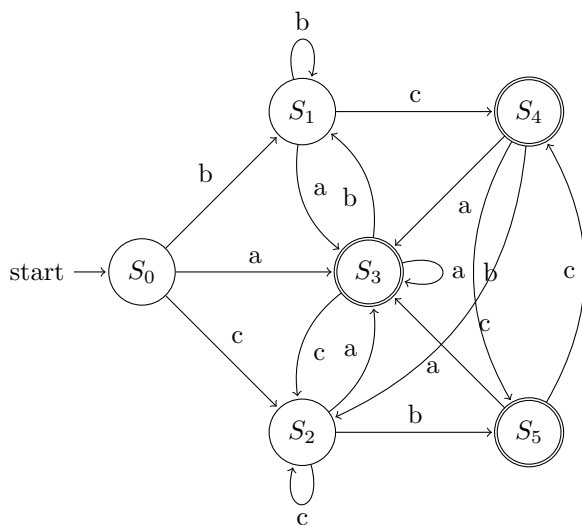
the origin is not elegant, So I replace it with the minimized version.
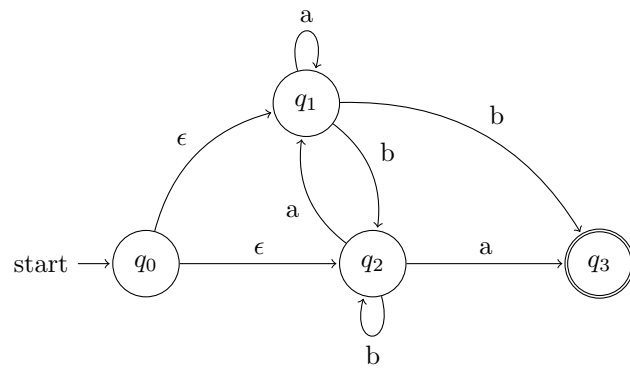
(b) Original NFA, $\Sigma = \{a, b, c\}$:
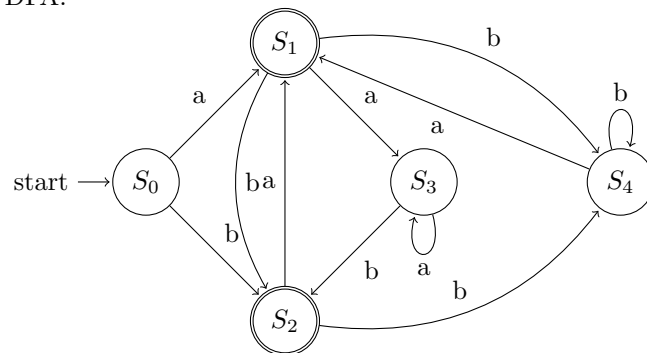
DFA:
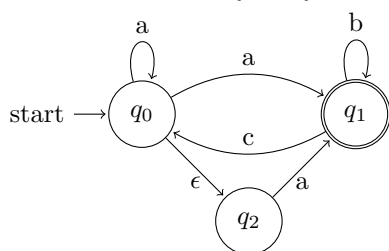
the origin is not elegant, So I replace it with the minimized version.

(c) Original NFA, $\Sigma = \{a, b, c\}$:

DFA:

5. (13 **pts**) Draw the NFA for the set of all strings over the alphabet $\Sigma = \{a, b\}$, where either a occurs an odd number of times and each of pair of a is separated by exactly $2n + 2$ consecutive b (for some $n \geq 0$), or b occurs an even number of times and each of pair of relative consecutive b is separated by exactly $2m + 1$ consecutive a (for some $m \geq 0$). Examples of strings that should be accepted by this NFA: abbabbbba, babaaabaaaaab. Examples of strings that should **not** be accepted: ababb, abbbabba.
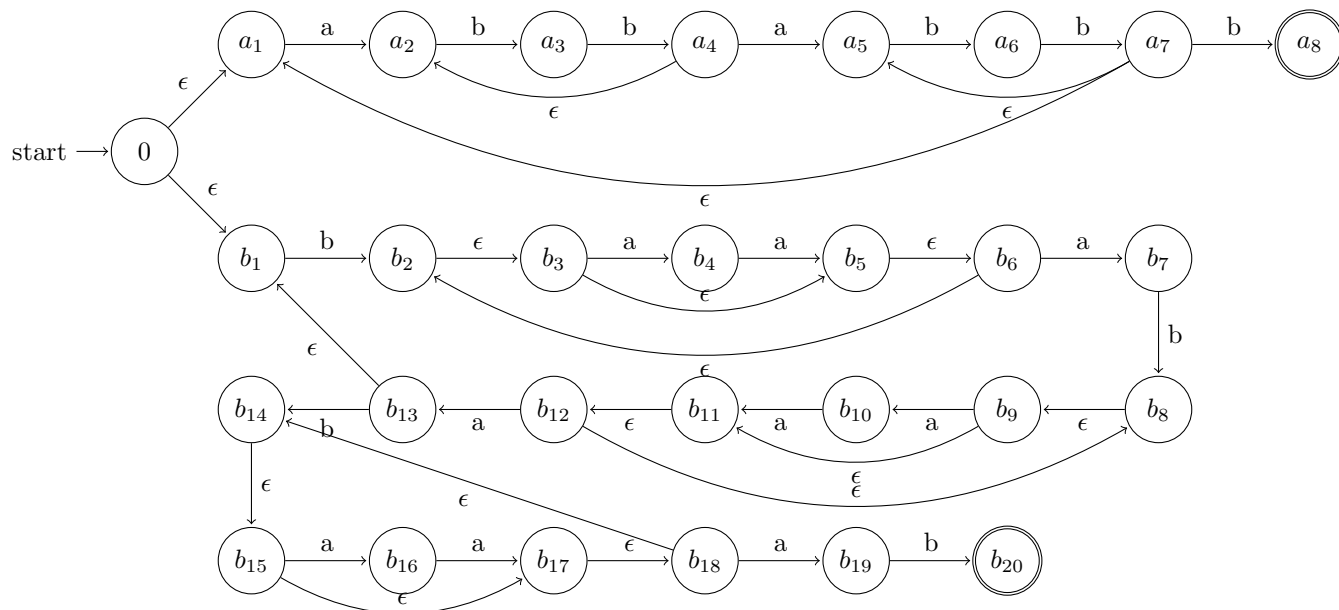
6. Consider the following tokens and their associated regular expressions, given as a **flex** scanner specification:

```
%%
(if)                            {printf("IF");}
[0-9]+                          {printf("NUM");}
[a-zA-Z0-9]+                    {printf("ID");}
[ ]                             {}
```

Give an input to this scanner such that the output string is $(\texttt{IF}^2\texttt{ID}^3\texttt{NUM}^2)^2$, where $\texttt{A}^i$ denotes $\texttt{A}$ repeated i times. (And, of course, the parentheses are not part of the output.) You may use similar shorthand notation in your answer.

$$if\ if\ aA9\ aA9\ aA9\ 9\ 9\ if\ if\ aA9\ aA9\ aA9\ 9\ 9$$

7. Draw the minimal DFA of the DFA constructed in Question 4(c).