CSI3104 (Spring-Summer 2019)
ASSIGNMENT 2

Due Date: **Email your completed assignment file (in Word or PDF format) directly to the Corrector by 14:00 on Tuesday, June 18, 2019.**

Instructions:

- This assignment has 3 pages. It consists of 10 exercises, each of which is worth 5 marks, resulting in 50 marks in total.
- You must do your assignment **individually**.
- Late assignment submissions will **not** be accepted: They will receive the grade of zero.
- Your assignment file must have a cover (front) page containing your full name, student number, course number, and assignment number. The Corrector will provide your assignment mark on this cover page, comments/feedback (if any) on your work, and will send the marked assignment file back to you.
- Name your assignment file using the format:

  CSI3104_A2_YourLastName_YourStudentID.docx or

  CSI3104_A2_YourLastName_YourStudentID.pdf
- Note: It is acceptable if you choose to complete your assignments using handwriting (instead of using a typesetting software). However, you must scan your completed, handwritten assignment into a PDF file and send that scanned PDF file to the Corrector. You must make sure that your handwriting is readable/clear for the Corrector to read in this case.

---

1. Build a TG that accepts the language L_1 of all words that begin and end with the same double letter, either of the form $aa...aa$ or of the form $bb...bb$.

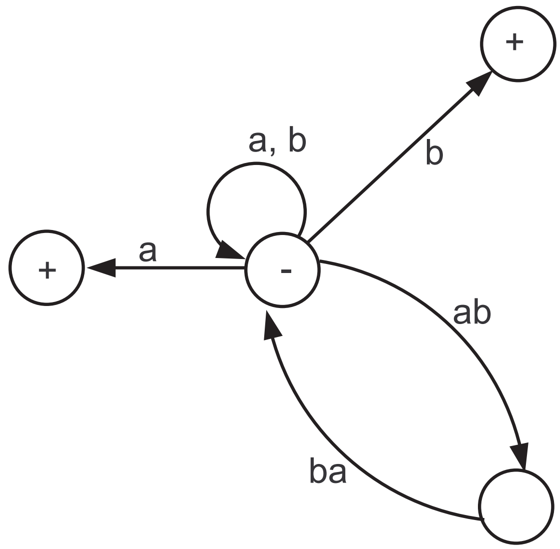
Note: aaa and bbb are not words in this language.

2. An FA with four states was sitting unguarded one night when vandals came and stole an edge labeled a. What resulted was a TG that accepted exactly the language $\mathbf{b}^*$. In the morning the FA was repaired, but the next night vandals stole an edge labeled b and what resulted was a TG that accepted $\mathbf{a}^*$. The FA was again repaired, but this time the vandals stole two edges, one labeled a and one labeled b, and the resultant TG accepted the language $\mathbf{a}^* + \mathbf{b}^*$.

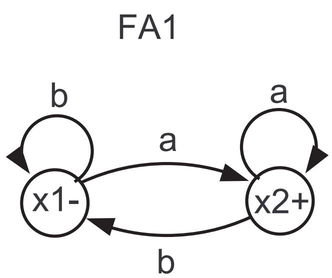
(i) What was the original FA?

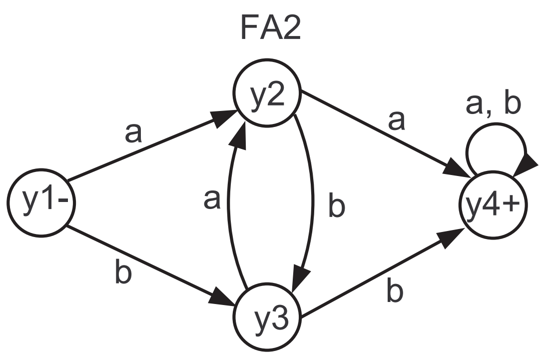
(ii) Clearly identify the stolen edges (e.g., which edge was stolen on night 1? on night 2? and which two edges were stolen on night 3?).

3. Using the bypass algorithm in the first proof of Theorem 6 (Kleene's theorem), Part 2, convert the following TG into regular expression:



4. Given the following FA_1 and FA_2. Use the algorithm in the first proof of Kleene's theorem, Part 3, Rule 2 to construct an FA for the union language $FA_1 + FA_2$.



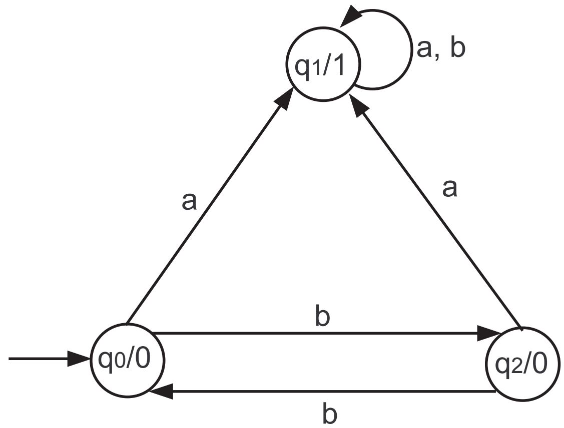


5. Using the algorithm in **Proof 2** of Kleene's theorem, Part 3, Rule 2, construct an NFA for the union language $FA_1 + FA_2$, where FA_1 and FA_2 are given in Problem 4 above.

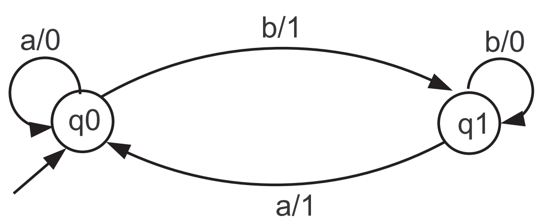
6. We are now interested in proving Part 3, Rule 3, of Kleene's theorem by using NFAs. The basic theory is that when we reach any + state in FA_1, we could continue to FA_1 by following its a-edge and b-edge, or we could pretend that we have jumped to FA_2 by following the a-edge and b-edge coming out of the start state on FA_2. We do not change any states or edges in either machines; we merely add some new (nondeterministic) edges from + states in FA_1 to the destination states of FA_2's start state. Finally, we erase the +'s from FA_1 and the − sign from FA_2, and we have the desired NFA.

Use this algorithm to find the NFA for the product language FA_1FA_2, where FA_1 and FA_2 are given in Problem 4 above.

7. Convert the following Moore machine into Mealy machine:



8. Convert the following Mealy machine into Moore machine:



9. Let $L_1 = language((\mathbf{a}+\mathbf{b})^*\mathbf{a})$ and let $L_2 = language(\mathbf{b}(\mathbf{a}+\mathbf{b})^*)$. Find a regular expression and an FA that each define $L_1 \cap L_2$.

10. Find a regular expression and an FA that each define $L_1 \cap L_2$ where $L_1 = language((\mathbf{a}+\mathbf{b})\mathbf{b}(\mathbf{a}+\mathbf{b})^*)$ and $L_2 = language(\mathbf{b}(\mathbf{a}+\mathbf{b})^*)$.