Name ______________________________

Student Number ______________________________

HAND IN
answers recorded
on question paper

BISHOP'S UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

CS 310

MID-TERM EXAMINATION

9 March 2023

Instructor: Stefan D. Bruda

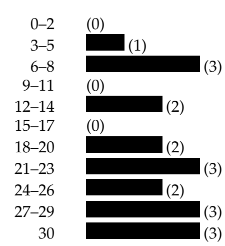
Instructions

- This examination is *80 minutes in length* and is *open book*. You are allowed to use any kind of printed documentation. Electronic devices are permitted only if they demonstrably have no communication capabilities. You are *not* allowed to share material with your colleagues. *Any violation of these rules will result in the complete forfeiture of the examination.*
- There is no accident that the total number of marks add up to the length of the test in minutes. The number of marks awarded for each question should give you an estimate on how much time you are supposed to spend answering the question.
- *To obtain full marks provide all the pertinent details.* This being said, do not give unnecessarily long answers. In principle, all your answers should fit in the space provided for this purpose. If you need more space, use the back of the pages or attach extra sheets of paper. However, if your answer is not (completely) contained in the respective space, clearly mention within this space where I can find it.
- The number of marks for each question appears in square brackets right after the question number. If a question has sub-questions, then the number of marks for each sub-question is also provided.

When you are instructed to do so, turn the page to begin the test.

1		10	/	10			
2		5	/	5			
3	a	10	/	10			
	b	5	/	5			
4		5	/	5			
5		5	/	5			
6	a	10	/	10			
	b	10	/	10			
	c	5	/	5			
	d	5	/	5			
7		10	/	10			
Total:		80	/	80	=	30 /	30

The highest grade was 30 (100%), the lowest grade was 5 (17%), and the average grade was 20.11 (67%). Here is the grade distribution:



1. [10] For each question below check *one* answer. Your response is a priori incorrect if more than one answer is checked.

   - Which of the following statements is true:
     - ☐ Some deterministic finite automata do not have equivalent regular expressions.
     - ☐ Some regular expressions do not have equivalent nondeterministic finite automata.
     - ☐ Some nondeterministic finite automata do not have equivalent regular expressions.
     - ☑ The above three statements are all false.
   - Which of the following statements is true:
     - ☐ Some finite languages are not regular.
     - ☐ Some regular languages are not context-free.
     - ☑ Some context-free languages are not regular.
     - ☐ The above three statements are all false.
   - Let $L = a^*b(b + ab^*a)^*$. Which of the following statements is true:
     - ☑ $bbbaba \in L$
     - ☐ $abbba \in L$
     - ☐ $aaba \in L$
     - ☐ None of the above strings is in L.
   - The language $\{0^k 1^{2n} 0^{3k} : k, n \geq 1\}$ is generated by the following context-free grammar:
     - ☐ $S \to 0S000 \quad S \to A \quad A \to 11A \quad A \to \varepsilon$
     - ☑ $S \to 0S000 \quad S \to 0A000 \quad A \to 11A \quad A \to 11$ (corrected here, sorry)
     - ☐ $S \to AB \quad A \to 0A000 \quad A \to \varepsilon \quad B \to 1B \quad B \to \varepsilon$
     - ☐ $S \to AB \quad A \to 0A000 \quad A \to 0000 \quad B \to 1B \quad B \to 1$
   - The language $\{0^{2k} 1^n : k \geq n \geq 0\}$ is denoted by the regular expression:
     - ☐ $(00)^*1^*$
     - ☐ $0^*(00)^*1^*$
     - ☐ $0^*(001 + 000011 + 000000111)^*$
     - ☑ None of the above.

2. [5] The definition of regular languages in the textbook is:

   > A regular language is a formal language definable using only union, concatenation, and closure from alphabet elements, ε, and $\emptyset$.

   Even if we eliminate ε from the definition we still define regular languages. Explain why.

ANSWER:

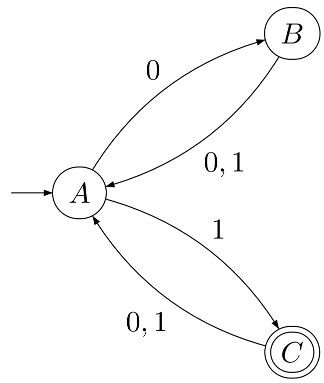
ε can be produced using $\emptyset$ and closure as follows: $\emptyset^* = \varepsilon$. So any occurrence of ε can be replaced in any regular expression by $\emptyset^*$ without changing the language being generated.

---

3. [15] Let L be the language of exactly all the strings over $\Sigma = \{0, 1\}$ of odd length that end with the symbol 1.

(a) [10] Draw the *deterministic* state transition diagram that recognizes L.

ANSWER:



(b) [5] Give a grammar that generates the language L.

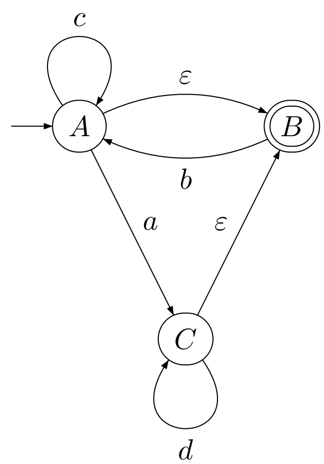
ANSWER:

We follow the conversion from finite automaton: $G = (\{A, B, C\}, \{0, 1\}, R, A)$, where

$$
\begin{aligned}
R = \{ \quad & A \to 0B, \\
& A \to 1C, \\
& B \to 0A, \\
& B \to 1A, \\
& C \to 0A, \\
& C \to 1A, \\
& C \to \varepsilon \quad \}
\end{aligned}
$$

---

4. [5] Use the systematic method described in the course to convert the following state transition diagram into an equivalent state transition diagram without ε-transitions.

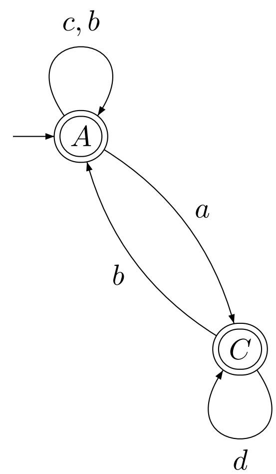


ANSWER:

Only following paths need to be "contracted":

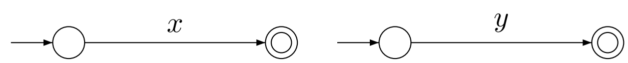
$$A \xrightarrow{\varepsilon} B \xrightarrow{b} A \quad \text{becomes} \quad A \xrightarrow{b} A$$
$$C \xrightarrow{\varepsilon} B \xrightarrow{b} A \quad \text{becomes} \quad C \xrightarrow{b} A$$

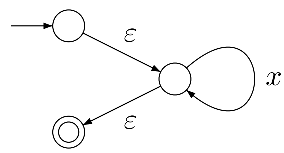
This also eliminates the state B from the picture, since it becomes unreachable (and so useless). However, the states A and C become final, since in the initial automaton there are ε-transitions from a final state (B) to each of them. The resulting automaton looks like this:

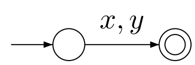


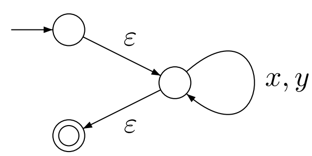
5. [5] Draw a state transition diagram that recognizes the language $x^*(x+y)^*$. Justify your answer.

ANSWER:

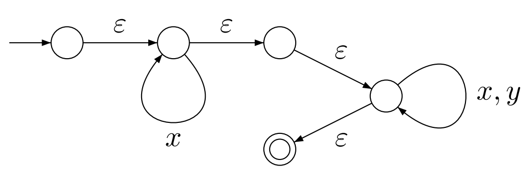
(a) Automata for x and y: 

(b) Automaton for x^* (from 5a): 

(c) Automaton for $x+y$ (from 5a): 

(d) Automaton for $(x+y)^*$ (from 5c): 

(e) Automaton for $x^*(x+y)^*$ (from 5b and 5d) = the answer to the question:



---

6. [30] Consider the language $L = \{a^i b^{2i} c^{2k} : i \geq 1, k \geq 1\}$.

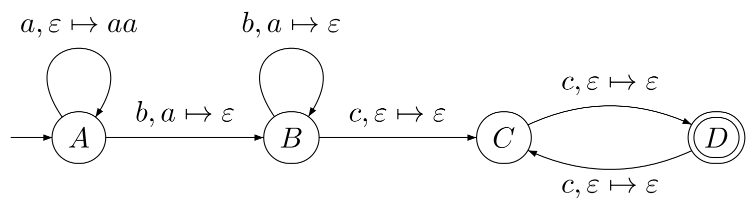
(a) [10] Give a context-free grammar that generates L.

ANSWER:

$S \to AC$
$A \to aAbb \quad A \to abb$
$C \to ccC \quad C \to cc$

(b) [10] Draw a *deterministic* push-down automaton that recognizes the language L.

ANSWER:



(c) [5] Give a derivation for the string $aabbbbcc$ using your grammar from Question 6a.

ANSWER:

$S \Rightarrow AC \Rightarrow aAbbC \Rightarrow aabbbbC \Rightarrow aabbbbcc$

(d) [5] Draw a table that traces the run of your push-down automaton from Question 6b on input $aabbcc$. The table should list the current state, currently remaining input, and current stack contents at each step of the computation. Explain why the input is accepted or rejected (as the case might be).

ANSWER:

State	Input	Stack
A	$aabbcc$	ε
A	$abbcc$	aa
A	$bbcc$	$aaaa$
B	bcc	aaa
B	cc	aa
C	c	aa
C	ε	aa

At the end of the run we are in an accepting state and but the stack is not empty, so the input is rejected.

---

7. [10] Is the language $A = \{a^{2i}b^{3i} : i \geq 1\} + \{c^{j+5}d^{2k+1} : j \geq 1, k \geq 1\}$ regular? If so, then give a regular expression that defines it; if not, then give a proof.

ANSWER:

The language is not regular. We assume that it is, so the pumping lemma applies; let n be the threshold from the lemma. We can choose any "long enough" string, so we choose one from the first component. Therefore let $w = a^{2n}b^{3n}$. This is clearly long enough so that it can be written as $w = xyz$ with $|xy| < n$ and $y \neq \varepsilon$. $|xy| < n$ means that xy (and so y) contains only a's and therefore $y = a^m$ for some $m > 0$. Therefore $xy^2z = a^{2n+m}b^{3n}$. According to the pumping lemma $xy^2z \in A$ and therefore $(2n+m)/3n = 2/3$, which implies that $m = 0$, in contradiction with $m > 0$ as established above. Thus A is not regular.
