Final Exam

Instructions: This is a closed book exam with 5 pages of problems. Do all problems. You are allowed one letter-sized page of handwritten notes. No electronic devices are allowed. Communicate your ideas ***clearly*** and ***succinctly***. Show all work. Good luck!

Your name and class ID:

Your email (in case I have a question):

On problem	you got	out of
1		5
2		5
3		5
4		5
5		5
6		5
7		10
8		5
9		5
Σ		50

1) Draw a three-state DFA over alphabet $\{a, b\}$ that accepts a string if and only if the number of a's it contains is not a multiple of 3. For example λ, *baaba*, and *aaaaaa* would each be rejected.

2) Draw an NPDA that accepts the same language that the grammar $S \to aSa \mid bSb \mid \lambda$ generates. (I think it can be done in 3 states).

3) Write a function `count_true` in Scheme or Racket that takes a one-input boolean function `f` and a list `xs` as inputs, and returns how many of the inputs cause `f` to evaluate to true. For example `(count_true odd?  '(1 2 3))` would evaluate to 2.

---

4) Write a function `remove` in Scheme or Racket that takes a list of numbers and a number to remove as inputs, and returns a copy of the list but without any occurrences of the number to be removed. For example `(remove '(1 2 3 3 2 1) 2)` would evaluate to `'(1 3 3 1)`.

5) Let $L = \{w \mid w$ has the same number of a's and b's$\}$ be a language over alphabet $\{a, b\}$. For example, λ and *abba* are both in L. Finish the following proof that L is not regular.

Theorem: L is not regular.

Proof:

– For purposes of contradiction, assume L is regular.

– Because L is regular there is a DFA M where $L(M) = L$. Let's say M has $k - 1$ states.

– Consider the string $w =$ ______________.

– Because $w \in L$, we know $w \in L(M)$, and because $|w| \geq k$ the pumping lemma says there are strings x, y, z such that $w = xyz$, $|xy| \leq k$, and $xy^i z \in L(M)$ for all $i \geq 0$.

– This causes a contradiction because . . .

---

6) Write a regular expression that generates the set of all non-empty strings over alphabet $\{a, b\}$ that never have the same character twice in succession and begin and end with the same character. For example *a*, *aba* and *babab* are in the set, but *abab* and *abba* are not.

7) Let's say you want to write a parser for the following grammar.

$$
\begin{array}{rcl}
S & \rightarrow & A\$ \\
A & \rightarrow & BAa \mid \lambda \\
B & \rightarrow & bBc \mid AA
\end{array}
$$

List all of the First and Follow set relationships that you can establish from the $A \rightarrow BAc$ production above. Set relationships are things that look like $x \in First(X)$, $Follow(X) \subseteq Follow(Y)$, etc. *Hint: There are five of them, including one that's a tautology.*

To determine whether a grammar is suitable for predictive parsing and to determine which production is selected for which input tokens, we've been filling tables like the following. Complete the table.

Production	First RHS	If nullable RHS, Follow LHS
$A \rightarrow BAa$		
$A \rightarrow \lambda$		
$B \rightarrow bBc$		
$B \rightarrow AA$		

Based on the table, is the grammar suitable for predictive parsing? Explain.

8) Demonstrate that the following grammar is ambiguous.

$$\begin{aligned} A &\rightarrow BAa \mid \lambda \\ B &\rightarrow bBc \mid AA \end{aligned}$$

9) Write a pseudocode reduction to show that `min(L)`, the problem of finding the minimum element of a list, reduces to `max(L)`, the problem of finding the maximum element of a list. What two implications does your reduction establish? *Hint: Multiply all the elements by -1.*