

Common Errors in CS154 Midterm Solutions

Problem 1(a) (3 Points)

Which of the following statements is true about a regular language L that is infinite?

- (a) \bar{L} is necessarily finite.
- (b) \bar{L} is sometimes finite and sometimes infinite.
- (c) \bar{L} is necessarily infinite.

If you choose (a) or (c), sketch a short (2/3 lines) proof. If you choose (b), provide two examples, one where \bar{L} is finite and another where \bar{L} is infinite.

Possible Answer:

(b) \bar{L} is sometimes finite and sometimes infinite

Let $\Sigma = \{0\}$.

Let $L = \Sigma^* - \{000\}$. Then \bar{L} is finite.

Let $L = \{w \mid w \text{ has even length}\}$. Then \bar{L} is infinite.

Common Pitfalls:

1. Giving a *non-regular* or *Context Free* language as an example for L .
2. The following has a small bug: $L = \{0x \mid x \in \Sigma^*\}$ and $\bar{L} = \{1x \mid x \in \Sigma^*\}$ where $\Sigma = \{0, 1\}$.

Problem 1(b) (2 Points)

Design a regular expression for the language $L = \{0^a \mid a \text{ is an even number and } a \geq 1\}$.

Possible Answers:

$00(00)^*$ or $(00)^+$

Common Pitfalls:

1. $(00)^*$ and $00 \cup (00)^*$ are wrong.

Problem 1(c) (3 Points)

Design a regular expression for the language $L = \{0^a 1^b \mid (a + b) \text{ is an odd number and } a, b \geq 0\}$.

Possible Answers:

$0(00)^*(11)^* + (00)^*1(11)^*$ or $(00)^*(0 + 1)(11)^*$

Common Pitfalls:

1. $(00)1(1^*)^2 \cup 0(0^*)^2(11)^*$
2. $1^+(00)^+ \cup (11)^+0^+$
3. Making ϵ part of L .
4. Writing something like $(011|100)^*$. Note that all zeros *must* be before all the ones.
5. Writing a CFG like $S \rightarrow 0S1 \mid 00S \mid S11 \mid 0 \mid 1$

Problem 1(d) (5 Points)

Let $L \in \Sigma^*$ be a regular language with $|\Sigma| \geq 1$. For $a \in \Sigma$, let L_{aaa} denote the language $\{w \mid w \in L \text{ and } w \text{ does not contain the string } aaa \text{ as a substring}\}$. Prove that L_{aaa} is regular.

Hint: There is a very short proof for this problem.

Possible Answer:

The language $L_1 = \{w \mid w \text{ contains } aaa \text{ as a substring}\}$ is equivalent to the regular expression $\Sigma^*aaa\Sigma^*$. It is straightforward to construct an NFA for this language. Now $L_{aaa} = L \cap \overline{L_1}$. Since complementation and intersection preserve regularity, L_{aaa} is regular.

Common Pitfall:

An astonishingly large number of students got this problem wrong. Note that there is a difference between the following two languages:

$L_{aaa} : \{w \mid w \in L \text{ and } w \text{ does not contain the string } aaa \text{ as a substring}\}$.

$L_{wrong} : \{w \mid w \in \Sigma^* \text{ and } w \text{ does not contain the string } aaa \text{ as a substring}\}$.

The subscript aaa in L_{aaa} is like an operator that takes a language as an operand and returns a new language. Most students drew an NFA for L_{wrong} , claiming it is L_{aaa} .

Problem 1(e) (7 Points)

Prove that the language generated by the following Context-Free Grammar is not regular:

$$S \rightarrow aSbb \mid aaSb \mid \epsilon$$

The set of terminals is $\{a, b\}$ and the start symbol is S .

Possible Answer:

We prove that L is non-regular by Pumping Lemma:

$\forall p \geq 1$

$\exists s = a^{2p}b^p \in L$ such that

$\forall x, y, z$ with $s = xyz$,

$\exists k \geq 0$ such that $xy^kz \notin L$. The choice $k = 2$ makes the string xy^kz of the form $a^{2p+\ell}b^p$ where $\ell = |y| \geq 1$. We cannot generate $a^{2p+\ell}b^p$ with $\ell \geq 1$ using the grammar above because each production generates at most two a 's for each b . Thus L is not regular.

The string $s = a^{2p}b^p \in L$ because we can generate s by using p productions of the form $S \rightarrow aaSb$ followed by $S \rightarrow \epsilon$.

Common Pitfalls:

1. *Choosing a string $\notin L(S)$.* Strings of the form $a^p b^p$ are not necessarily in the language for arbitrary p . For example, $a^2 b^2$ and $a^4 b^4$ do not belong to $L(S)$. For Pumping Lemma, we need to choose that belongs to $L(S)$ and then show that we cannot pump it. Strings like $a^{2p} b^p$, $a^p b^{2p}$ and $a^{3p} b^{3p}$ are okay. A string like $a^{p/2} b^p$ is *not* okay.

2. *Wrong characterization of $L(S)$:*

$\{a^i b^j \mid (i+j) = 0 \pmod{3}\}$. This includes $abbbbb \notin L(S)$.

$\{a^i b^j \mid i+j = 3k \text{ for some } k \geq 0\}$. This includes $abbbbb \notin L(S)$

$\{a^i b^{2i} \mid i \geq 0\} \cup \{a^{2i} b^i \mid i \geq 0\}$

$\{a^{3i} b^{3i} \mid i \geq 0\}$

Here are some interesting descriptions that are correct:

$\{a^i b^j \mid i+j = 3k \text{ where } |i-j| \leq k\}$

$\{a^i b^j \mid i+j = 3n \text{ where } n \leq i, j \leq 2n \text{ and } n \geq 0\}$

$\{a^{3i+j} b^{3i+2j} \mid i, j \geq 0\} \cup \{a^{3i+2j} b^{3i+j} \mid i, j \geq 0\}$.

$\{a^{2i+j} b^{i+2j} \mid i, j \geq 0\}$

Note that for this problem, it is not required that you specify the exact set of strings generated by the grammar. The proof using Pumping Lemma can be written without specifying $L(S)$ explicitly.

3. A few students gave the following argument that is flawed: Let $L_1 = \{a^{3k} b^{3k} \mid k \geq 0\}$. We know that $L_1 \subset L(S)$. It is easy to show that L_1 is not regular. Therefore $L(S)$ must be non regular.

4. A few students gave the following argument that is bizarre: *We can see that $L(S) = L_1 \cup L_2$, where ... (some description of L_1 and L_2) We can see that both L_1 and L_2 are non-regular. Since regular languages are closed under union, $L(S)$ must be non-regular.*

5. Almost a fifth of the students gave the following proof using Pumping Lemma: *Let $s = a^p b^{2p}$. We know that $s \in L(S)$. We will split $s = xyz$ such that $|xy| \leq p$. Now, there are three cases: (a) y consists of a 's, (b) y consists of a 's and b 's, and (c) y consists only of b 's. Contd...*

Notice that cases (b) and (c) above cannot arise because $|xy| \leq p$. If you write (b) and (c), your proof is not flawed. However, it's a waste of time writing them.

6. Using the Pumping Lemma for CFG's. Wake up! The language under consideration *is known to be a CFG.*

Problem 2(a) (6 Points)

Design a Push Down Automata (PDA) to accept the language generated by the following CFG:

$$S \rightarrow aSbb \mid \epsilon$$

The set of terminals is $\{a, b\}$ and the start symbol is S .

Draw a state diagram for the PDA and write a few lines of verbal description.

Possible Answers:

Push \$, push all a 's and then to consume two b 's for every a in the stack. Finally, pop \$ and accept.

or

Push \$, push two a 's for each a . Then start consuming one b for each a in the stack. Finally, pop \$ and accept.

Common Pitfalls:

1. Using S as part of the alphabet Σ .
2. Not accepting the empty string ϵ .
3. Copying the PDA from one of the homework solutions that accepts the following language: $\{w \mid w \in (a+b)^* \text{ and } w \text{ has twice as many } b\text{'s as } a\text{'s}\}$.
4. Creating a PDA that accepts the language $(abb)^*$. Note that all a 's must precede all b 's.

This problem fetched most students full points.

Problem 2(b) (6 Points)

Design a Context Free Grammar for the language

$$L = \{a^m b^n c^p d^q \mid m + n = p + q \text{ and } m, n, p, q \geq 0\}$$

Possible Answer:

$$S \rightarrow aSd \mid T \mid U$$

$$T \rightarrow aTc \mid V$$

$$U \rightarrow bUd \mid V$$

$$V \rightarrow bVc \mid \epsilon$$

Common Pitfalls:

1. Common wrong CFG: $S \rightarrow \epsilon \mid aSd \mid aSc \mid bSd \mid bSc$. This allows S to produce strings like $abadcd \notin L$.
2. Writing a CFG that produces only a finite number of strings.

Problem 2(c) (6 Points)

Let language $L = \{a^m b^n c^p d^q \mid m+n = p+q \text{ and } m, n, p, q \geq 0\}$ with alphabet $\Sigma = \{a, b, c, d\}$.

Let $\bar{L} = \Sigma^* - L$.

Write an expression for \bar{L} . State whether \bar{L} is Context-Free or not. Prove your claim.

Correct Answer:

$\bar{L} = J \cup K$ where $J = \overline{a^* b^* c^* d^*}$ and $K = \{a^m b^n c^p d^q \mid m + n \neq p + q \text{ and } m, n, p, q \geq 0\}$.

Proof that \bar{L} is Context-Free:

The first language J is in fact regular because it is the complement of a regular language. All regular languages are context-free.

The second language K is context free because there is a PDA to recognize it. The PDA simply pushes all a 's and b 's it encounters. Then, for c 's and d 's, it pops one symbol each from the stack. The PDA accepts only if there are more symbols remaining in the input or the stack.

The union of two CFG's is context free. Therefore $\bar{L} = J \cup K$ is context-free.

An alternative expression for \bar{L} is the following:

$$\{a^m b^n c^p d^q \mid m + n \neq p + q \text{ and } m, n, p, q \geq 0\} \cup \Sigma^*(ba + ca + cb + da + db + dc)\Sigma^*$$

This is equivalent to:

$$\{a^m b^n c^p d^q \mid m + n \neq p + q \text{ and } m, n, p, q \geq 0\} \cup \{w \mid w \in \Sigma^* \text{ and } w \text{ contains any of the following}$$

as a substring: $ba, ca, cb, da, db, dc\}$.

Common Pitfalls:

1. Forgetting the $\overline{a^* b^* c^* d^*}$ portion.
2. Proving that $\{a^m b^n c^p d^q \mid m + n \neq p + q \text{ and } m, n, p, q \geq 0\}$ is a non-CFG using Pumping Lemma.

Problem 3(a) (6 Points)

For the Turing Machine drawn on the next page, the input alphabet is $\Sigma = \{\$, a, b\}$ and the tape alphabet is $\Gamma = \{\$, a, b, \sqcup, x\}$. Which of the following strings does it accept? (Just write Yes/No where Yes denotes acceptance).

Correct Answers:

No (a) \$\$

Yes (b) \$ab\$

Yes (c) \$baab\$abaaa

Problem 3(b) (6 Points)

What exactly is the language accepted by the Turing Machine drawn on the next page?

Correct Answer:

Strings of the form $\$w\$(a + b + \$)^*$ where $|w| \geq 2$, $w \in (a + b)^*$ and w contains an equal number of a 's and b 's.

Common Pitfalls:

1. Failing to mention that $|w| \geq 2$. The number of a 's is equal to the number of b 's. However, this number cannot be zero.
2. Failing to realize the trailing $(a + b - \$)^*$ portion. The TM just verifies that there are an equal (and non-zero) number of a 's and b 's between the first pair of \$ symbols and terminates. It does not even read the remainder of the input string. Part (c) above provides a clue to this behaviour of the TM.