10 (100 PTS.) Aberrant.

10.A. (25 PTS.) Prove that the following language is not regular by providing a fooling set. You need to prove an infinite fooling set and also prove that it is a valid fooling set. For $\Sigma = \{a, b\}$, the language is

$$L = \left\{ ww \;\middle|\; w \in \Sigma^+ \right\}.$$

Solution:

Consider the set $F = \left\{ b^\ell a \;\middle|\; \ell \geq 4 \right\}$.

Lemma 4.1. *The set F is an infinite fooling set for L.*

Proof: The set F is clearly infinite. As for the other property, consider two distinct words in L: $w_i = b^i a$ and $w_j = b^j a$, for any $i \neq j$. Observe that $w_i w_i = b^i a b^i a \in L$, but $w_j w_i = b^j a b^i a$ is not in L. Indeed, for $w_j w_i$ to be in L, we must have for some $w \in \Sigma^*$ that

$$b^j a b^i a = ww$$

for some w. This is impossible.

For completeness, here is the easy (low level) case analysis (which you did not have to write, but you should know how to do) – indeed, w must contain a single a. If a is the last character in w, then $ww = b^j a b^j a$, which is not the desired word. As such, it must be that the last character in w is a b, but that implies that the last character in ww is a b, which is again impossible. ■

10.B. (25 PTS.) Same as (A) for the following language. Recall that a ***run*** in a string is a maximal non-empty substring of identical symbols. Let L be the set of all strings in Σ^* that contains two distinct runs of equal length. A few examples about L:

- L contains any string of the form $b^i a^* b^* a^i$.
- L contains any string of the form $b^i a^* b^i$.
- L does not contain the strings $abbaaa$, $abbaaabbbb$.

Solution:

Lemma 4.2. *The set $F = \left\{ b^\ell \;\middle|\; \ell \geq 2 \right\}$. is an infinite fooling set for L.*

Proof: The set F is clearly infinite. As for being a fooling set, consider two distinct words in L: $w_i = b^i$ and $w_j = b^j$, for any $i \neq j$. Observe that $w_i a w_i = b^i a b^i \in L$, but $w_j a w_i = b^j a b^i$ is not in L.

This implies that w_i and w_j, for $i \neq j$, behave differently for the suffix $a w_i$, which implies that F is indeed the desired fooling set. ■

10.C. (25 PTS.) Suppose you are given two languages L, L' that are not regular but such that $L' \setminus L$ is regular. Prove that $L \cup L'$ is not regular. (Hint: Use closure properties of regular languages.)

Solution:

Claim 4.3. *Let L, L' be two languages that are not regular, but such that $L' \setminus L$ is regular. Then $L \cup L'$ is not regular.*

Proof: The proof is by magic, in this case the magic of negation. Assume that $L \cup L'$ is regular. We then have that

$$L = (L \cup L') \setminus (L' \setminus L) = \underbrace{(L \cup L')}_{B} \cap \underbrace{\overline{L' \setminus L}}_{C}$$

It is given that $L' \setminus L$ is regular. As such, its negation is also regular, since regular languages are closed under negation. That is $C = \overline{L' \setminus L}$ is regular. The intersection of two regular languages is a regular language, which implies that $L = B \cap C$ is regular, as B is regular by assumption. But that is a contradiction. ∎

10.D. (15 PTS.) Provide a counter-example for the following claim:

Claim: Consider two languages L and L'. If $\overline{L}$ is not regular, L' is regular, and $L \cup L'$ is regular, then $L \cap L'$ is regular.

Solution:

Let $L' = \Sigma^*$. Pick L to be any non regular language. Say $L = \left\{ a^i b^i \;\middle|\; i \geq 0 \right\}$. The complement language $\overline{L}$ is not regular, since regular languages are closed under complement (whatever this language is). Here $L \cup L' = \Sigma^*$, which most definitely regular. However, we have that $L \cap L' = L \cap \Sigma^* = L$, which is not regular, which the desired contradiction.

10.E. (10 PTS.) (Slightly harder[1]) Same as (A) for $L = \left\{ 0^{n^4} \;\middle|\; n \geq 3 \right\}$.

Solution:

The key observation is that the gaps between words in L keep increasing. In particular, we have the following.

Lemma 4.4. *For any $n \geq 1$, we have that $(n+1)^4 > n^4 + n + 1$.*

Proof: $(n+1)^4 - n^4 \geq (n+1-n)(n^3 + n^2 + n + 1) > n + 1$. ∎

Let $F = \left\{ 0^i \;\middle|\; i \geq 3 \right\}$.

Lemma 4.5. *The set F is an infinite fooling set for L.*

[1]Feel free to use IDK.

Proof: The set F is clearly infinite.

Consider any two positive distinct integers i, j such that $j > i > 3$, and observe that $s_i = 0^i$ and $s_j = 0^j$ are both in F. Let $\Delta = j^4 - i$. Observe that

$$s_i 0^\Delta = 0^i 0^\Delta = 0^{j^4 - i + i} = 0^{j^4} \in L.$$

$$s_j 0^\Delta = 0^j 0^\Delta = 0^{j^4 - i + j} = 0^{j^4 + (j-i)} \notin L,$$

since the next word in L (in length) after 0^{j^4} is $0^{(j+1)^4}$, and $j^4 + (j-i) < j^4 + j + 1 < (j+1)^4$, by the above lemma. ∎

11 (100 PTS.) Grammar it.

Describe a context free grammar for the following languages. Clearly explain how they work and the role of each non-terminal. Unclear grammars will receive little to no credit.

11.A. (40 PTS.) $\left\{ a^i b^j c^k d^\ell e^t \;\middle|\; i, j, k, \ell, t \geq 0 \text{ and } i + j + k + t = \ell \right\}$.

Solution:

$$\begin{aligned}
&E \to \varepsilon \mid dEe && // \; L(E) = \left\{ d^t e^t \;\middle|\; t \geq 0 \right\}\\
&C \to \varepsilon \mid cCd && // \; L(D) = \left\{ c^k d^k \;\middle|\; k \geq 0 \right\}\\
&B \to C \mid bBd && // \; L(C) = \left\{ b^j c^k d^\ell \;\middle|\; \ell = j + k \right\}\\
&A \to B \mid aAd && // \; L(B) = \left\{ a^i b^j c^k d^\ell \;\middle|\; \ell = i + j + k \right\}\\
&S \to AE.
\end{aligned}$$

Here S is the start symbol.

To see why this is correct, observe that for $w = a^i b^j c^k d^\ell e^t$ in the language, we have $i + j + k + t = \ell$, which implies

$$a^i b^j c^k d^\ell e^t = a^i b^j c^k d^{i+j+k+t} e^t = a^i b^j c^k d^i d^j d^k d^t e^t = a^i b^j c^k d^k d^j d^i d^t e^t = \underbrace{a^i \overbrace{b^j \underbrace{c^k d^k}_{C} d^j}^{C} d^i}_{A} \underbrace{d^t e^t}_{E}.$$

11.B. (60 PTS.) (Harder.) $L = \left\{ z \in \{a, b, c\}^* \;\middle|\; \text{there is a suffix } y \text{ of } z \text{ s.t. } \#_a(y) > \#_b(y) \right\}$. (Hint: First solve for the case that z has no cs.)

Solution:

We first consider the case when there are no cs. Let $z = z_1 z_2 \ldots z_m$ be a word in L, and let $y = z_k z_{k+1} \ldots z_m$ be the shortest suffix of z that has the desired property.

Observe that any proper suffix y' of y has $\#_b(y') \geq \#_a(y')$. For $x = z_{k+1} \ldots z_m$, we have $\#_b(x) = \#_a(x)$. That implies that x can be interpreted as a balanced parenthesis expression, with $b =$), and $a =$ (. The grammar for such expressions is easy:

$$R_1 \to \varepsilon \mid aR_1b \mid R_1R_1.$$

The variable R_1 can generate x. To generate y, we just need to add a. That is, x is generated by the rule:

$$R_2 \to aR_1.$$

Before the suffix y, z can have any string it whatsoever. As such, we add the rule that can generate any string:

$$R_3 \to bR_3 \mid aR_3 \mid \varepsilon.$$

And the string z is generated by the rule:

$$S' \to R_3R_2.$$

Now, we need to inject as many as we want to a generated string. To this end, we create a symbol that generates the language c^*. We also create "alternatives" to a and b that can include such runs before and after the letter appears. That is

$$\begin{aligned} C &\to \varepsilon \mid cC \\ A &\to CAC \mid a \\ B &\to CBC \mid b \end{aligned}$$

We now modify the overall grammar to use A and B instead of a and b. That is, we have the following grammar (with S being the start symbol):

$$\begin{aligned}
&C \to \varepsilon \mid cC && // \ L(C) = c^* \\
&A \to CAC \mid a && // \ L(A) = c^*ac^* \\
&B \to CBC \mid b && // \ L(B) = c^*bc^* \\
&T_1 \to \varepsilon \mid AT_1B \mid T_1T_1 && // \ L(T_1) = \text{balanced } a, b \text{ strings} \\
&T_2 \to AT_1 && \\
&T_3 \to AT_3 \mid BT_3 \mid CT_3 \mid \varepsilon && // \ L(T_2) = \{a, b, c\}^* \\
&S \to T_3T_2. &&
\end{aligned}$$

This is by no mean the shortest grammar for this task.

12 (100 PTS.) As easy as 1,2,3,6.

Let $L = \left\{ a^i b^j c^k \;\middle|\; k = i + j \right\}$.

12.A. (20 PTS.) Prove that L is context free by describing a grammar for L.

Solution:

$$B \rightarrow \varepsilon | bBc$$
$$S \rightarrow B | aSc.$$

12.B. (80 PTS.) Prove that your grammar is correct. (See extra problems for an example of how this is done.)

Solution:

Lemma 4.6. *$L(B) = R$, where $R = \left\{ b^i c^i \;\middle|\; i \geq 0 \right\}$.*

Proof: The proof is by induction, on the length of strings. For $n = 0$, both languages contains the empty string, since $B \rightarrow \varepsilon$, and by choosing $i = 0$.

It is easy to verify that both sets do not contain words of length 1.

Inductive assumption. So assume that the two sets contains exactly the same strings up to (and including) length $n = k$, for $k \geq 0$.

Inductive step. Consider a string $s \in L(B)$, which is of length $n = k + 1$. It must be that s was derive by first applying the rule $B \rightarrow bBc$. That is s can be written as $s = btc$, where $t \in L(B)$ and is of length $k + 1 - 1 = k - 1$. By induction, $t \in R$. namely, $t = a^i b^i$. But this implies that $s = atb = a^{i+1} T b^{i+1}$, which implies that $s \in R$.

Next for the other direction. Consider a string $u \in R$ of length $n = k + 1$. It can be written as $u = a^j b^j$, for $j \geq 1$. In particular, $u = aa^{j-1} b^{j-1} b$, and $v = a^{j-1} b^{j-1} \in R$. By induction, we have that $v \in L(B)$, which implies that using $B \rightarrow aBb$, one can conclude that $u = a^j b^j = avb \in L(B)$. ■

Lemma 4.7. *$L(S) = R'$, where $R' = \left\{ a^i b^j c^k \;\middle|\; k = i + j \right\}$.*

Proof: We repeat the same proof structure as above.

Base of induction. The empty string is clearly in both languages, and they do not contain any string of length 1.

Induction hypothesis. Assume that the two sets have exactly the same strings for all strings of length n where $n \leq \ell$, for $\leq \geq 2$.

Inductive step. Consider a string s of length $n = \ell + 1$.

$\Longrightarrow$ If $s \in L(S)$, then there are two possibilities for the first rule of derivation used in generating it:

- The first rule of derivation was $S \to B$. But this implies that $s \in L(B)$, and by the above, $s \in R$. Since $R \subseteq R'$, it follows that $s \in R$.
- The first rule of derivation was $S \to aSc$. This implies that $s = as'c$, where $s' \in L(S)$. Since $|s'| = k - 1$, by induction, it follows that $s' \in R$. As such, $s' = a^i b^j c^k$, such that $k = i + j$. This implies that

  $$s = as'c = aa^i b^j c^k c = a^{i+1} b^j c^{k+1},$$

  Since $(i+1) + j = k + 1$, it follows that $s \in R'$, as desired.

$\Longleftarrow$ If $s \in R'$, then it can be written as $a^i b^j c^k$ with $i + j = k$. We can write $s = a^i b^j c^j c^i$, since $k = i + j$. Let $s' = b^j c^j$. The string $s' \in R$, and by Lemma 4.6, we have $s' \in L(B)$. Doing i applications of the rule $S \to aSc$ followed by $S \to B$, implies that we can use the derivation $S \to a^i B c^i$. As such, we have $S \Rightarrow a^i B c^i \to a^i s' c^i = s$. This implies that $s \in L(S)$, as desired. ■