Note: the venn diagrams below are all the same, since original figures showing the solutions are missing; hint: use color pencils to complete these diagrams 🤭

10 (100 PTS.) This is all wrong.

10.A. (30 PTS.) Let $\Sigma = \{a, b\}$. For a word $w = w_1 w_2 \dots w_n \in \Sigma^*$, let $w_o = w_1 w_3 w_5 \dots w_{2\lceil n/2 \rceil - 1}$ be the string formed by the odd characters of w. Prove that the following language is not regular by providing a fooling set. Your fooling set needs to infinite, and you need also to prove that it is a valid fooling set. The language is $L = \left\{ w w_o \;\middle|\; w \in \Sigma^+ \right\}$.

Solution:

Let

$$F = \left\{ a^{2i} b a \;\middle|\; i \geq 1 \right\}$$

The set F is clearly infinite. As for being an infinite fooling set, consider any $j > i > 0$, and observe that

$$w_i = a^{2i} b a \quad \text{and} \quad w_i a^i b \in L$$
$$w_j = a^{2j} b a \quad \text{and} \quad w_j a^i b \notin L.$$

Name w_i and w_j are distinguishable for L, and F is a valid fooling set proving that L is not regular.

10.B. (30 PTS.) Provide a counter-example for the following claim (if you need to prove that a specific language is regular [or not], please do so):

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

Solution:

The claim is false, and we provide a counter example.

Let $L = \left\{ 0^i 1^j \;\middle|\; i \geq j \right\}$ and $L' = \left\{ 0^i 1^j \;\middle|\; i \leq j \right\}$. Consider their intersection language $K = L \cap L' = \left\{ 0^i 1^j \;\middle|\; i = j \right\}$. The language K is not regular, since $F = \left\{ 0^i \;\middle|\; i \geq 0 \right\}$ is a fooling set for it. Indeed,

$$0^i \cdot 1^i \in K \text{ but } 0^j \cdot 1^i \notin K,$$

for $i \neq j$. Fortunately, F is a fooling set also for L and L', as

$$0^i \cdot 1^i \in L \text{ but } 0^j \cdot 1^i \notin L,$$

for $j < i$. Similarly,

$$0^i \cdot 1^i \in L' \text{ but } 0^j \cdot 1^i \notin L',$$

for $j > i$. We conclude that L, L' and $L \cap L'$ are all not regular languages. But $L \cup L' = 0^* 1^*$, which is definitely regular.

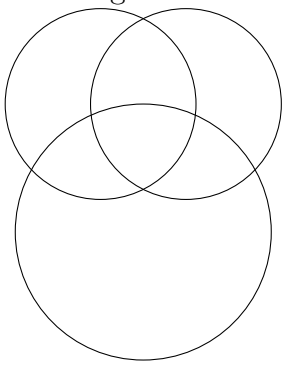
10.C. (40 PTS.) Suppose you are given three languages L_1, L_2, L_3, such that:

- $L_1 \cup L_2 \cup L_3$ is not regular.
- For all $i \neq j$: $L_i \setminus L_j$ is regular.

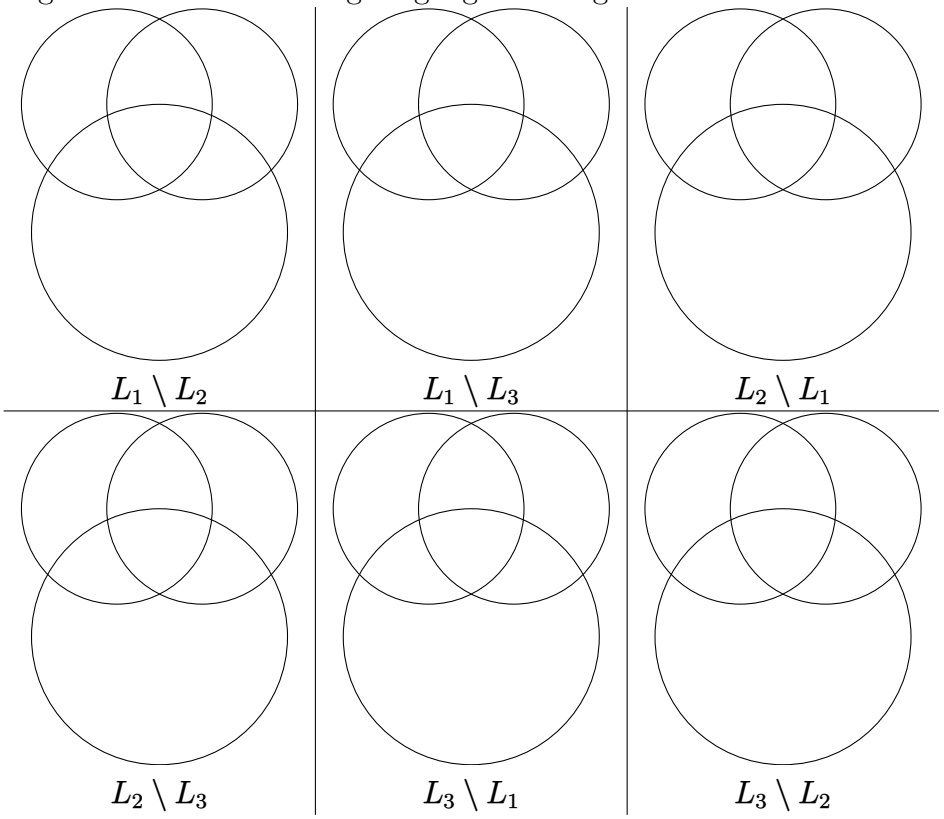
Prove that $L_1 \cap L_2 \cap L_3$ is not regular. (Hint: Use closure properties of regular languages.)
(Not for submission: Can you come up with an example of such languages?)

Solution:

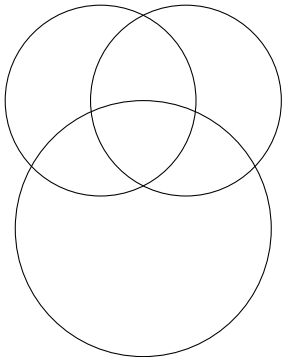
Proof: Let us consider the Venn diagram of three languages:



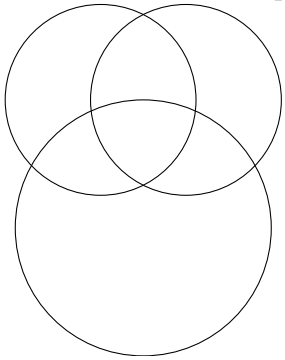
We are given that the following languages are regular:



In particular, the union of all these languages $K = \cup_{i<j} L_i \setminus L_j$ is regular, since regular languages are closed under finite union.



Now, consider the set $L_1 \cap L_2 \cap L_3$, which is the purple region in the following figure.



If $L_1 \cap L_2 \cap L_3$ is regular, then $L_1 \cup L_2 \cup L_3 = K \cup (L_1 \cap L_2 \cap L_3)$, would be regular, since it is the union of two regular languages. A contradiction. We conclude that $L_1 \cap L_2 \cap L_3$ is not regular. ■

Solution:

Here is a version of the proof without figures.

Proof: Observe that $K = \bigcup_{i \neq j}(L_i \setminus L_j)$ is regular, being the union of regular languages. Furthermore, observe that $(L_1 \cup L_2 \cup L_3) \setminus K = L_1 \cap L_2 \cap L_3$, as can easily be verified. Or stated differently $L_1 \cup L_2 \cup L_3 = K \cup (L_1 \cap L_2 \cap L_3)$.

Now, assume for the sake of contradiction that $L_1 \cap L_2 \cap L_3$ is regular. But then $L_1 \cup L_2 \cup L_3 = K \cup (L_1 \cap L_2 \cap L_3)$ is the union of two regular languages, which implies that it is itself regular (as regular languages are closed under union). This is a contradiction, and we conclude that $L_1 \cap L_2 \cap L_3$ is not regular. ■

11 (100 PTS.) Grammarticus.

For (A) and (C) below, 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.) $L = \{a^i b^j c^k d^\ell \mid i,j,k,\ell \geq 0 \text{ and } j+\ell = i+k\}$.

Hint for this question would be posted on piazza question thread.

> Hint: As a warm-up, consider the special cases that $i = 0$ or $j = i$. Then, consider the case that $j \geq i$, introduce the variable $\Delta = j - i$, and restate the language in this case. Then handle the other case that $j \leq i$. And then put everything together. You solution should include only the final grammar, not all the middle steps.

Solution:

First consider the case that $j \geq i$ and $\ell \leq k$. Since $k = (j-i)+\ell$, we have that

$$a^i b^j c^k d^\ell = a^i b^i \cdot b^{j-i} c^{j-i+\ell} \cdot d^\ell = \left(a^i b^i\right) \cdot \left(b^{j-i} c^{j-i}\right) \cdot \left(c^\ell d^\ell\right).$$

We readily get the following grammar for this language:

$$\begin{array}{ll} S_1 \to CDE & \\ C \to \varepsilon \mid aCb & // \; C = \{a^\tau b^\tau \mid \tau \geq 0\} \\ D \to \varepsilon \mid bDc & // \; D = \{b^\tau c^\tau \mid \tau \geq 0\} \\ E \to \varepsilon \mid cEd & // \; E = \{c^\tau d^\tau \mid \tau \geq 0\} \end{array}$$

Now, consider the case that $j \leq i$ and $\ell \geq k$. We have that $\ell - k = i - j$ and

$$a^i b^j c^k d^\ell = a^{i-j}\left(a^j b^j\right) \cdot \left(c^k d^k\right) d^{\ell-k} = a^{i-j} \underbrace{\left(a^j b^j\right) \cdot \left(c^k d^k\right)}_{C \cdot E} d^{i-j}.$$

We readily get the following grammar for this language:

$$S_2 \to aS_2d \mid CE$$

The desired language is

$$G \to S_1 \mid S_2.$$

Or stated fully, it is the language $L(G)$, for

$$\begin{array}{ll} G \to S_1 \mid S_2 & \\ S_1 \to CDE & \\ S_2 \to aS_2d \mid CE & \\ C \to \varepsilon \mid aCb & // \; C = \{a^\tau b^\tau \mid \tau \geq 0\} \\ D \to \varepsilon \mid bDc & // \; D = \{b^\tau c^\tau \mid \tau \geq 0\} \\ E \to \varepsilon \mid cEd & // \; E = \{c^\tau d^\tau \mid \tau \geq 0\} \end{array}$$

11.B. (30 PTS.) Let $\Sigma = \{a, b\}$. Consider the language

$$L_B = \{z \in \Sigma^* \mid \text{ for any prefix } y \text{ of } z \text{ we have } \#_a(y) \geq \#_b(y)\}.$$

Prove that any $w \in L_B$, can be written as $w = w_1 \cdots w_m$, such that $w_i = a$ or w_i is a balanced string, for all i. A string $s \in \{a.b\}^*$ is ***balanced*** if $\#_a(s) = \#_b(s)$.

(One can also prove a stronger version, where in addition each w_i is strongly balanced [i.e., $w_i \in L_B$].)

Solution:

For the string w, let $w[1 \ldots i]$ denote the substring formed by the first i characters of w. For a string s let $F(s) = \#_a(s) - \#_b(s)$. Similarly, let $f(i) = F(w[1 \ldots i])$. Observe that a string s is in $L_B \iff$ for all prefixes s' of s, we have $F(s') \geq 0$.

Lemma 4.1. *Any string $w \in L_B$, can be written as $w = w_1 \cdots w_m$, where the strings $w_1, \ldots, w_m \in L_B$, and, for all i, we hve*

(i) $w_i = a$, *and*

(ii) w_i *is balanced,*

Proof: Let w be a string made out of n characters. The proof is by induction on the length of w.

If $|w| = 0$ then $w = \epsilon$, and the claim holds.

If $|w| = 1$, then $w = a$, and the claim holds.

Assume the claim is true if $|w| < n$, for $n \geq 2$. And consider the case that $|w| = n$,

If $f(n) = 0$ then w itself is balanced, and the claim holds.

If $f(i)$ is zero, for any i between 1 and $n-1$, then $x = w[1 \ldots i]$ is balanced, and $w = xy$, and $y = w[i+1 \ldots n]$. As x is balanced, we have that $F(x) = 0$. This implies that for any $j \geq i+1$, we have

$$F(w[i+1 \ldots j]) = F(w[1 \ldots j]) - F(x) = f(j) - F(x) = f(j) \geq 0.$$

Namely $w[i+1 \ldots n] \in L_B$, and inductively it can be broken into strings $y_1, \ldots, y_t \in L_B$, such that for any j, $y_j = a$ or y_j is balanced. As such, $x \cdot y_1 \cdots y_t$ is the desired decomposition of w, as $x \in L_B$.

The remaining cases are when $f(1), \ldots, f(n)$ are all non-zero.

So consider the case that $f(n) > 0$. This implies that for $i = 1, \ldots, n-1$, we have $f(i) > 0$ (as f changes its value by at most one between consectuvie values). This implies that for the suffix string $w'_j = w[2 \ldots j]$, for any $j \geq 2$, we have $F(w'_j) = f(j) - 1 \geq 0$. Namely, $w'_n = w[2 \ldots n] \in L_B$. By induction, it has the desired decomposition $\widehat{w_1} \cdots \widehat{w_m}$, and $w = a \cdot \widehat{w_1} \cdots \widehat{w_m}$, as desired.

The case that $f(n) < 0$, is impossible, as it is given that $f(n) \geq 0$. ■

11.C. (30 PTS.) Describe a grammar for the language L_B defined above, using the property you proved in (11.B.) (you can use the stronger version without proving it). **Prove** the correctness of your grammar.

Solution:

We need to first prove something similar to what we had seen in homework problem (1.B.):

Lemma 4.2. *if a string $s \in \Sigma^*$ is balanced, then either:*

(i) $s = \epsilon$,
(ii) $s = xy$ *where* x *and* y *are non-empty balanced strings, or*
(iii) $s = axb$, *where* x *is a balanced string.*
(iv) $s = bxa$, *where* x *is a balanced string.*

Proof: If $|s| = 0$, then s is an empty string, $F(s) = 0$, and $s = \epsilon$, as claimed.

If there is any proper prefix x of s, such that $F(x) = 0$ (*proper* means $x \neq \epsilon$ and $x \neq s$), then $s = xy$, and $0 = F(s) = F(x) + F(y) = F(y)$, which implies that $F(y) = 0$, and the claim holds as both x and y are balanced.

If $F(x) > 0$ for every proper prefix of s, then it must be that the first character of s is a, and the last character is b. We can thus write s as $as'b$ Since $F(s) = 0$, it follows that $0 = F(s) = F(s') + 1 - 1 = F(s')$. Namely, s' is balanced, and the claim holds.

If $F(x) < 0$ for every proper prefix of s, implies in a similar fashion that $s = bs''a$, and s'' must be balanced, thus implying the claim. ■

This lemma readily implies, that the language $L(Z_B)$, for

$$Z_b \rightarrow bZ_ba \mid aZ_bb \mid Z_BZ_b \mid \varepsilon,$$

is the language of all balanced strings. The problem is that we are interested only in string that are both balanced, and are in L_B. As such, we have

$$Z \rightarrow aZb \mid ZZ \mid \varepsilon,$$

We are now ready to the kill. **(11.B.)** implies that a string $w \in L_B$ is a sequence of strings, each one of them either a, or alternatively balanced. As such, the grammar to generate such a string is

$$W \rightarrow Z \mid a.$$

And the grammar to generate an arbitrary sequence of such strings is

$$S \rightarrow SW \mid \varepsilon.$$

Putting everything together, we get the grammar

$$\begin{aligned} S &\rightarrow SW \mid \varepsilon \\ W &\rightarrow Z \mid a \\ Z &\rightarrow aZb \mid ZZ \mid \varepsilon, \end{aligned}$$

12 (100 PTS.) The pain never ends.

12.A. (50 PTS.) Let $\Sigma = \{a, b\}$. A string $s \in \Sigma^*$ is a ***palindrome*** if $s = s^R$. For a prespecified integer $k \geq 0$, a string $s \in \Sigma^*$ is ***k-close*** to being a palindrome, if there is a string $w \in \Sigma^*$ that is a palindrome, and one recover w from s by a sequence of (***at most***) k operations. Each such operation is either inserting one character or deleting a character. Thus $ababaaab$ is 2-close to a palindrome since

$$ababaaab \to babaaab \to baaaab.$$

Similarly, the string $ab^2a^2b^5a^5b^4a^3b^2a$ is 2-close to being a palindrome since

$$ab^2a^2b^5a^5b^4a^3b^2a \to ab^2\underline{a^3}b^5a^5b^4a^3b^2a \to ab^2a^3\underline{b^4}a^5b^4a^3b^2a.$$

Let L_k be the language of all strings that are k-close to being a palindrome. Give a CFG for L_3. Argue why your solution is correct.

Solution:

The grammar for L_0, which is just the language of all palindromes, is

$$S_0 \to aS_0a \mid bS_0b \mid a \mid b \mid \varepsilon.$$

The grammar for L_1 is

$$S_1 \to S_0 \mid aS_1a \mid bS_1b \mid S_0b \mid S_0a \mid aS_0 \mid bS_0 \mid \varepsilon.$$

For generally, the grammar for L_i is

$$S_i \to aS_ia \mid bS_ib \mid S_{i-1}b \mid S_{i-1}a \mid aS_{i-1} \mid bS_{i-1} \mid S_{i-1}.$$

The intuition is simple – every time an insert or delete operation is performed, we are left with one fewer such operations we can perform, and we have to move to the lower level language.

For our question, we have

$$\begin{aligned}
S_3 &\to aS_3a \mid bS_3b \mid S_2b \mid S_2a \mid aS_2 \mid bS_2 \mid S_2 \\
S_2 &\to aS_2a \mid bS_2b \mid S_1b \mid S_1a \mid aS_1 \mid bS_1 \mid S_1 \\
S_1 &\to aS_1a \mid bS_1b \mid S_0b \mid S_0a \mid aS_0 \mid bS_0 \mid S_0 \\
S_0 &\to aS_0a \mid bS_0b \mid a \mid b \mid \varepsilon.
\end{aligned}$$

12.B. (50 PTS.) Let $\Sigma = \{a, b\}$. Prove that if $L \subseteq \Sigma^*$ is context-free language then

$$\text{subsequence}(L) = \{x \in \Sigma^* \mid \exists y \in L, x \text{ is a subsequence of } y\}$$

is a context-free language.

Solution:

Consider a grammar G of L. Replace any appearance in G of $a$by a new symbol A, and similarly replace any appearance of $b$in L by a new symbol B. Add the rules

$$A \to a \qquad \text{and} \qquad B \to b.$$

to G, and let G_1 be this resulting grammar. It is clear that $L(G) = L(G_1)$. Now, replace the two new derivations, by the derivations

$$A \to a \mid \varepsilon \qquad \text{and} \qquad B \to b \mid \varepsilon.$$

Let G_2 be the resulting grammar.

Claim 4.3. $\text{subsequence}(L) = L(G_2)$.

Proof: $\text{subsequence}(L) \subseteq L(G_2)$: For a string $x \in \text{subsequence}(L)$, let y be its superstring in L, and let T be its derivation tree in G_1. For every character in y that is not present in x, replace its derivation from $A \to a$ or $B \to b$ by $A \to \varepsilon$ or $B \to \varepsilon$, respectively. Clearly, the resulting derivations now generate x over G_2, and we conclude that $x \in G_2$.

$L(G_2) \subseteq \text{subsequence}(L)$: For any string $x \in G_2$, consider its derivation tree. Replace any derivation $A \to \varepsilon$ or $B \to \varepsilon$ by $A \to a$ or $B \to b$, respectively. Clearly, the resulting string generated by this derivation string is a superstring y such that $y \in L(G_1) = L(G)$, and as such x is a subsequence of a string of L, which implies that $x \in \text{subsequence}(L)$. ■