# Quiz 1 Solutions

Automata Theory Monsoon 2025, IIIT Hyderabad

September 8, 2025

1. [3 points] Prove that regular languages are closed under dropout.

$$Dropout(A) = \{xz \mid xyz \in A, y \in \Sigma \text{ and } x, z \in \Sigma^*\}$$

**Solution:** Since $A$ is a regular language, it must be recognized by a DFA. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that recognises $A$. Now, we will construct an NFA $N = (Q', \Sigma \cup \{\epsilon\}, \delta', q_o', F')$ that recognizes Dropout($A$).

The **idea** behind the construction is that $N$ simulates $M$ on its input, non-deterministically guessing the point at which the dropped out symbol occurs. At that point $N$ guesses the symbol to insert in that place, without reading any actual input symbol at that step. Afterwards, it continues to simulate $M$.

We implement this idea in $N$ by keeping two copies of $M$ , called the top and bottom copies. The start state is the start state of the top copy. The accept states of $N$ are the accept states of the bottom copy. Each copy contains the edges that would occur in $M$. Additionally, include $\epsilon$ edges from each state $q$ in the top copy to every state in the bottom copy that $q$ can reach.

We describe $N$ formally. The states in the top copy are written with a $T$ and the bottom with a $B$, thus: $(T, q)$ and $(B, q)$.

- $Q' = \{T, B\} \times Q$
- $q_0' = (T, q_0)$
- $F' = \{B\} \times F$
- $\delta'((T, q), a) = \begin{cases} \{(T, \delta(q, a))\} & a \in \Sigma \\ \{(B, \delta(q, b)) \mid b \in \Sigma\} & a = \epsilon \end{cases}$
- $\delta'((B, q), a) = \begin{cases} \{(B, \delta(q, a))\} & a \in \Sigma \\ \emptyset & a = \epsilon \end{cases}$

2. [3 points] Find a regular expression for the language recognized by this machine, using the procedure we have studied in class: Show all your work, in particular, the state diagrams after the removal of each successive state. You may omit $\epsilon$-transitions from your diagrams while constructing the GNFA.

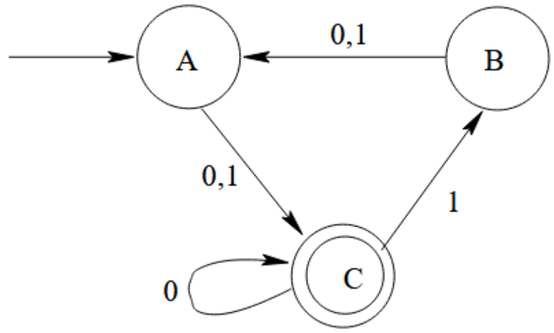


Figure 1: DFA for Question 2

**Solution:** Use the algorithm for GNFAs taught in class:

- Add a new start and end state to the DFA.
- Select a state $q_{rip}$ and rip it out of the machine.
- "Repair" the machine by altering the regular expressions that label each of the remaining arrows.
- Repeat this process until only two states (start and end) are left.
- The label from of the arrow from the start state to the final state of the GNFA is the required regex.

**Required regex:** $(0+1)0^*((1(0+1)(0+1))+0)^*$ (or equivalent)

You are expected to show the GNFA state diagram every time you rip out a state.

---

**Marking scheme**

- Adding a new start state $\rightarrow$ 0.25
- Adding a new end state $\rightarrow$ 0.25
- Ripping out state A $\rightarrow$ 0.5
- Ripping out state B $\rightarrow$ 0.5
- Ripping out state C $\rightarrow$ 0.5
- Correct final regex $\rightarrow$ 1

3. [2 + 2 points] For the language L = $\{w \in \{0,1\}^* | \ w$ has twice as many 0's as 1's$\}$,

   (i) Construct a Context Free Grammar for language L.

(ii) Draw the PDA for this language.

**Solution:**

1. • $S$

---

**Marking Scheme**

1. • The correct grammar is given full marks.
   • If the grammar accepts any string outside the language, then **NO MARKS WILL BE GIVEN**
   • If the grammar accepts a subset of the language, then marks are given proportionally.
2. • A correct PDA is given full marks.
   • A meaningful PDA or a PDA that corresponds to your incorrect grammar in part (i) will fetch you 0.5 marks.
   • A PDA that does not correspond to the grammar described in part (i) that accepts a subset of the language will be given partial marks.

4. [5 points] Let $\Sigma = \{0, 1, 2\}$ and

$$L = \{\, 0^i 1^j 2^k \mid i, j, k \geq 0, \quad i > j > k \,\}.$$

Prove that this language is not context-free using the Pumping Lemma for CFLs. First explain the pumping lemma for CFLs (the conditions that need to be specified). Choose the string to be $s = 0^{p+2}1^{p+1}2^p$. Show all cases where split is possible and show the contradictions clearly. At the end include the marking scheme specified.

**Solution:**

**Pumping Lemma for Context-Free Languages.** If $L$ is an infinite context-free language then there exists an integer $p \geq 1$ (called the pumping length) such that for every string $s \in L$ with $|s| \geq p$ we can write

$$s = u\,v\,w\,x\,y$$

with $u, v, w, x, y \in \Sigma^*$ satisfying:

1. $|vwx| \leq p$,
2. $|vx| \geq 1$ (i.e. at least one of $v, x$ is nonempty),
3. for every integer $n \geq 0$, the pumped string $u\,v^n w x^n y \in L$.

**Assume for contradiction.** Assume $L = \{0^i 1^j 2^k \mid i > j > k\}$ is context-free. Let $p$ be the pumping length from the lemma.

**Choice of string.** Choose

$$s = 0^{p+2}\, 1^{p+1}\, 2^{p}.$$

Then $s \in L$ because $p+2 > p+1 > p$, and $|s| > p$. By the pumping lemma write $s = uvwxy$ with $|vwx| \le p$ and $|vx| \ge 1$.

Because $|vwx| \le p$ and the three blocks have lengths $p+2,\ p+1,\ p$ respectively, the substring $vwx$ must lie entirely inside one of the three blocks or cross at most one boundary between adjacent blocks. We consider all possible placements of $v$ and $x$.

**Case 1.** $vwx$ **lies entirely in the** 0**-block.**
Then $vx$ consists only of zeros; let $t \ge 1$ be the number of zeros in $vx$. Pump down with $n = 0$:

$$uv^0wx^0y \quad \text{has} \quad i' = (p+2) - t, \quad j' = p+1, \quad k' = p.$$

Since $t \ge 1$, we get $i' \le p+1 = j'$, so $i' \le j'$, contradicting the requirement $i' > j'$. Thus this case is impossible.

**Case 2.** $vwx$ **lies entirely in the** 1**-block.**
Let $t \ge 1$ be the number of ones in $vx$. Pump down with $n = 0$:

$$i' = p+2, \quad j' = (p+1) - t, \quad k' = p.$$

Because $t \ge 1$, $j' \le p = k'$, so $j' \le k'$, contradicting $j' > k'$. So this case is impossible. (You can also pump up as done in **Case 3** and it will be considere)

**Case 3.** $vwx$ **lies entirely in the** 2**-block.**
Let $t \ge 1$ be the number of twos in $vx$. Pump up with $n = 2$:

$$i' = p+2, \quad j' = p+1, \quad k' = p+t.$$

Since $t \ge 1$, $k' \ge p+1$, hence $j' \le k'$, contradicting $j' > k'$. So this case is impossible.

**Case 4.** $vwx$ **crosses the boundary between the** 0**-block and the** 1**-block.**
Then $vx$ contains some zeros and some ones (in particular at least one 1). Let $b \ge 1$ be the number of ones in $vx$. Pump down with $n = 0$. The number of ones becomes $j' = (p+1) - b \le p$ while $k' = p$, so $j' \le k'$, contradicting $j' > k'$. Thus this case is impossible.

**Case 5.** $vwx$ **crosses the boundary between the** 1**-block and the** 2**-block.**
Then $vx$ contains some ones and some twos (in particular at least one 2). Pump up with $n = 2$ (or pump down with $n = 0$); pumping up increases the number of twos by at least 1, so the new $k' \ge p+1$ while $j' = p+1$, giving $j' \le k'$, contradicting $j' > k'$. Hence this case is impossible.

**Extra case. Only if chosen string is different such that** $vwx$ **contains symbols from all three blocks.**

Then $vx$ contains at least one 1 or at least one 2. If it contains a 1, pump down ($n = 0$) to reduce $j$ by at least 1, yielding $j' \leq p$ and thus $j' \leq k' = p$, contradicting $j' > k'$. If it contains a 2, pump up ($n = 2$) to increase $k$ by at least 1, yielding $k' \geq p+1$ and hence $j' = p+1 \leq k'$, contradicting $j' > k'$. Thus this case is impossible.

All possible positions of $v, x$ lead to a contradiction of the required inequalities $i > j > k$. Thus our assumption that $L$ is context-free is false. Therefore

$$\boxed{L \text{ is not context-free.}}$$

---

**Marking scheme**

- Nothing $\rightarrow 0$
- Writing the correct pumping lemma for CFL $\rightarrow 1$
- Just correct string was chosen $\rightarrow 1.5$
- String + statement (pumping lemma applied to that string) $\rightarrow 2$
- For each case: 0.5 marks (each of the main cases above).
  But for the case where the pumping string completely belongs to $0^{p+2}$: 1 mark.
- Extra case if the chosen string is $0^{p-1}1^{p-2}2^{p-3}$: the special subcase where the pumped substring can produce a mixed block like $01\ldots12$ needs to be considered as well (0.5 marks).