1. Determine \(L = \{a^nb^m : n/m \text{ is an integer} \} \) whether regular or not?

We claim \(L\) is not regular and we prove this by contradiction.

For contradiction assumption, let's assume \(L = \{a^nb^m : n/m \text{ is an integer} \}\) is a regular language. Since \(L\) is regular it should satisfy the pumping lemma. Assume \(L\) is regular, then by pumping lemma there exist a finite state machine \(D\) with \(p\) states that recognizes the language \(L\).

There exist a number \(p\) (pumping length) by pumping lemma.

Let \(s = a^pb^p \in L\) be a string that \(|s| > p\) and is eligible for pumping lemma. By pumping lemma, \(\exists x, y, z\) such that:

1. \(s = xyz\)
2. \(|xy| \leq p\)
3. \(|y| > 0\)

\(x = a^j; j > 0 \leftarrow x \) can be zero or more of the \(a\)'s.

\(y = a^k; k > 1 \leftarrow y \) has to be at least one of the \(a\)'s.

\(z = a^{p-(j+k)}b^p \leftarrow z\) will be the rest of \(a\)'s and \(b\)'s.

By pumping lemma \(xyz \in L\)

Let \(i = 0\) and pump down \(y\): \(i = 0 \Rightarrow a^j(a^k)a^{p-(j+k)}b^p = a^ja^k\)

\[
\frac{p-k}{p} = 1 - \left( \frac{k}{p} \right) \Rightarrow 0 < 1 - \left( \frac{k}{p} \right) < 1 \Rightarrow \frac{p-k}{p} \text{ is not integer therefore } a^ja^k \not\in L.
\]

(pumping lemma) \(k \geq 1\)

Therefore, the assumption is false and no such machine \(D\) exists that recognizes language \(L\).

\(\therefore L\) is not regular.
2. Determine \( L = \{ a^n b^m : n > 25, m \leq 25 \} \) is regular or not?

We want to prove that \( L \) is regular. We do it by introducing a DFA that recognize it and showing that all strings \( \forall w \) recognized by DFA must be in \( L \).

[Diagram of a DFA with states and transitions for accepting \( a \) and \( b \).]

First, we show that \( \forall w \in L \) the DFA recognize it. Here since it is not clear in the question we assume \( 0 \leq m \leq 25 \) means that \( m \) can be zero.

In the first \( q_{a1} \) to \( q_{a25} \) states we receive \( a \) and if in any of the first states of \( q_{a1} - q_{a25} \) there is \( b \) in input we go to dead state. All the dead states are actually one single dead state that for a clear diagram we duplicate it. In the last, \( q_{a25} \) we wait for more \( a \) or on \( b \) we go to \( q_{b1} \), \( q_{f1} \) to \( q_{f25} \) also accept \( b \) and all of them are accepting states in addition to \( q_{a25} \) (because \( m \) could be \( m = 0 \)). After \( q_{b25} \), if we receive more \( a \) or \( b \) we reject. This shows that \( \forall w \in L \) the DFA recognize it.

Second, we show \( \forall w \) recognized in \( a^* \) it must be in \( L \).

\( \forall w \) which is recognized by DFA has reached \( q_{a25} \) so it has at least 25 \( a \) or more in the beginning of the string and no \( b \) in the first 25 states is accepted. Therefore the string begin with \( a^n \), \( n > 25 \).
iso, \( \forall w \text{ recognized by DFA \ could \ reach \ any \ of \ the \ state \ } q_6 \text{ to } q_{25} \) after \( q_{25} \) and still be accepted by DFA. Therefore, \( w \) could have \( b^m \); \( b \leq m \), \( b \)'s after the first 25 or more a's. If \( w \) has more than 25 b or any a in the middle of b's it is immediately rejected.

Consequently, all \( w \) recognized by DFA and can reach accept states can only be in the form of \( w = a^n b^m \); \( n \geq 25 \); \( m \leq 25 \) and therefore they are in \( L \) and others are rejected. DFA only recognizes all strings in \( L \) and no others.

Since there exist a DFA to recognize \( L \) and DFA only recognizes all strings in \( L \) & no others, \( L \) is regular. \( \checkmark \)