

## Daily Problem: 22–Jan–2026

### Problem Statement

For every  $n \in \mathbb{N}$ , construct a regular language  $L_n$  such that:

- $L_n$  can be recognized by an NFA with at most  $n$  states, but
- every DFA recognizing  $L_n$  requires at least  $2^n$  states.

Give a specific family of languages and prove both the upper bound (small NFA) and the lower bound (large DFA).

### Construction of the Language

Fix  $n \in \mathbb{N}$  and consider the alphabet  $\Sigma = \{0, 1\}$ . Define the language

$$L_n = \{w \in \Sigma^* \mid |w| \geq n \text{ and the } n\text{-th symbol from the right in } w \text{ is } 1\}.$$

Equivalently, a string  $w$  lies in  $L_n$  if we can write  $w = x1y$  where  $|y| = n - 1$ .

Intuitively,  $L_n$  consists of all binary strings whose  $n$ -th-from-last position contains a 1.

### Upper Bound: NFA with at most $n + 1$ States

We show that  $L_n$  can be recognized by an NFA with  $n + 1$  states.

## Construction of the NFA

Define an NFA  $N_n = (Q, \Sigma, \delta, q_0, F)$  as follows:

- $Q = \{q_0, q_1, \dots, q_n\}$ ,
- the start state is  $q_0$ ,
- the only accepting state is  $q_n$ , i.e.  $F = \{q_n\}$ .

The transition function  $\delta$  is defined as:

- From  $q_0$ :

$$\delta(q_0, 0) = \{q_0\}, \quad \delta(q_0, 1) = \{q_0, q_1\},$$

meaning the NFA stays in  $q_0$  on 0, and on 1 it may either stay in  $q_0$  or guess that this 1 is the  $n$ -th-from-last bit and jump to  $q_1$ .

- From  $q_i$  for  $1 \leq i < n$ :

$$\delta(q_i, 0) = \{q_{i+1}\}, \quad \delta(q_i, 1) = \{q_{i+1}\},$$

meaning that once the guess is made, the NFA moves deterministically one step forward for exactly  $n - 1$  symbols.

- From  $q_n$ :

$$\delta(q_n, 0) = \{q_n\}, \quad \delta(q_n, 1) = \{q_n\},$$

so once in the accepting state, it stays there.

## Correctness of the NFA

If  $w \in L_n$ , we can write  $w = x1y$  with  $|y| = n - 1$ . While reading the highlighted 1, the NFA can choose to jump from  $q_0$  to  $q_1$ , and then after reading the remaining  $n - 1$  symbols of  $y$ , it will reach  $q_n$ , which is accepting.

Conversely, if the NFA accepts  $w$ , then there must be some nondeterministic jump at a 1 that places the NFA into  $q_1$ , and from there exactly  $n - 1$  symbols are consumed before acceptance. Thus the jump must correspond to the  $n$ -th symbol from the end, which must be 1. Therefore  $w \in L_n$ .

Thus  $L_n$  is recognized by an NFA with  $n + 1$  states, so in particular at most  $n + 1 \leq n + \text{constant}$ , satisfying the required upper bound.

## Lower Bound: Every DFA Requires $2^n$ States

We now prove that any DFA recognizing  $L_n$  must have at least  $2^n$  states.

### High Level Idea

Any DFA must, after reading the first  $n$  symbols of an input, remember exactly which length- $n$  string it has seen so far, because this determines whether or not the input is in  $L_n$ .

### Formal Proof

Let  $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$  be any DFA recognizing  $L_n$ . Consider all strings in  $\Sigma^n$ , i.e. all binary strings of length  $n$ . There are  $2^n$  such strings.

We claim that for any two distinct strings  $u, v \in \Sigma^n$ , the DFA must enter different states after reading them from the start state  $q_D$ . Formally, if

$$u \neq v \in \Sigma^n,$$

then

$$\hat{\delta}_D(q_D, u) \neq \hat{\delta}_D(q_D, v).$$

### Proof of the Claim

Suppose for contradiction that  $u \neq v$  but

$$\hat{\delta}_D(q_D, u) = \hat{\delta}_D(q_D, v).$$

Since  $u \neq v$ , they differ at some position, hence their  $n$ -th-from-last symbols differ. Exactly one of  $u$  or  $v$  has a 1 in that critical position and hence exactly one of them belongs to  $L_n$ .

However, if the DFA ends in the same state after reading both, appending the empty string  $\varepsilon$  would lead to the same accept/reject outcome for both. This contradicts the fact that exactly one of them is in  $L_n$ .

Thus the reached states must be distinct for all  $2^n$  length- $n$  strings.

### Conclusion of the Lower Bound

Since there are  $2^n$  distinct strings of length  $n$ , and each must lead to a distinct state of the DFA, the DFA must have at least  $2^n$  states. Hence every DFA recognizing  $L_n$  requires at least  $2^n$  states.

## Final Conclusion

We have exhibited a family of regular languages  $\{L_n\}_{n \in \mathbb{N}}$  such that:

$$\text{NFA-size}(L_n) \leq n + 1 \quad \text{and} \quad \text{DFA-size}(L_n) \geq 2^n.$$

Therefore, this construction demonstrates an exponential gap between NFA and DFA state complexity.