## FINITE-AUTOMATA

The time complexity of finite automata (FA) depends on the context:

* **State complexity:** This refers to the number of states in the FA. In itself, it's not a measure of runtime complexity. However, the size of an FA can impact the efficiency of algorithms that operate on them (e.g., simulation, minimization). There's no universal formula, but for some operations (like converting between deterministic and non-deterministic FAs), the worst-case complexity can be exponential in the number of states.
* **Running time on input string:** This is where the interesting complexity analysis comes in. FAs process input strings one symbol at a time. In the best and average case, recognizing a string with a finite automaton takes linear time (O(n)) in the length (n) of the string. This is because each symbol is processed once, and the number of transitions performed scales linearly with the input size.
* **Deterministic vs. Non-deterministic:** Deterministic FAs (DFA) always have a unique next state for a given symbol in a current state. This allows for efficient processing as there's no ambiguity about which transition to follow. Non-deterministic FAs (NFA) can have multiple possible transitions, which might require exploring all paths in the worst case. However, for most regular languages, efficient DFA constructions from NFAs exist, preserving the linear time complexity.

**Here's a breakdown:**

* **Best case & Average case:** O(n) - The FA processes the entire input string in a single pass, recognizing or rejecting it in linear time.
* **Worst case (NFA):** Exponential (O(n^k)) in the worst case, where k is the number of states and the input string can lead to exploring many potential paths in the NFA. However, this is uncommon for well-designed NFAs representing regular languages.

**Key takeaway:** Finite automata offer efficient string processing with linear time complexity in most practical scenarios. The state complexity can affect the efficiency of FA-related algorithms, but it doesn't directly impact the time to recognize a specific string.