Consider the language $C_k = \Sigma^* a \Sigma^{k-1}$ for each $k \geq 1$, over the alphabet $\Sigma = \{a, b\}$. C_k be the language consisting of all strings that contain an '*a*' exactly k places from the right-hand end.

Now it is required to prove that, no DFA (Deterministic finite automation) can recognize C_k with fewer than 2^k states.

If a DFA enters two different states after reading two different strings *lz* and *mz* with same suffix *z*, then the DFA enters two different states after reading the strings *l* and *m*. Take two different strings of length *k* such that $l = l_1 \ldots\ldots l_k$ and $m = m_1 \ldots\ldots m_k$. Let *i* be some position such that $l_i \neq m_i$. One of the strings contains *a* in its i^{th} position and the other string contains *b* in its i^{th} position.

Consider the suffix string $z = b^{i-1}$. In this case, either the string *lz* or *mz* has the k^{th} bit from the end as *a*. The total number of strings of length *k* over the input alphabet $\{a, b\}$ is 2^k. Thus, the DFA needs 2^k states in order to distinguish 2^k strings.

Therefore, the DFA that recognizes the language C_k has at least 2^k states.