Theorem 2.4.1: *Let L be a regular language. There is an integer $n \geq 1$ such that any string $w \in L$ with $|w| \geq n$ can be rewritten as $w = xyz$ such that $y \neq e$, $|xy| \leq n$, and $xy^iz \in L$ for each $i \geq 0$.*

Proof: Since L is regular, L is accepted by a deterministic finite automaton M. Suppose that n is the number of states of M, and let w be a string of length n or greater. Consider now the first n steps of the computation of M on w:

$$(q_0, w_1 w_2 \ldots w_n) \vdash_M (q_1, w_2 \ldots w_n) \vdash_M \ldots \vdash_M (q_n, e),$$

where q_0 is the initial state of M, and $w_1 \ldots w_n$ are the n first symbols of w. Since M has only n states, and there are $n+1$ configurations $(q_i, w_{i+1} \ldots, w_n)$ appearing in the computation above, *by the pigeonhole principle* there exist i and j, $0 \leq i < j \leq n$, such that $q_i = q_j$. That is, the string $y = w_i w_{i+1} \ldots w_j$ drives M from state q_i back to state q_i, and this string is nonempty since $i < j$. But then this string could be removed from w, or repeated any number of times in w just after the jth symbol of w, and M would still accept this string. That is, M accepts $xy^iz \in L$ for each $i \geq 0$, where $x = w_1 \ldots w_i$, and $z = w_{j+1} \ldots w_m$. Notice finally that the length of xy, the number we called j above, is by definition at most n, as required. ■