Regular Languages are in $\mathcal{NC}$

Theorem 1 *All regular languages are in $\mathcal{NC}$.*

Proof: Let L be a regular language over an alphabet Σ. We will prove that $L \in \mathcal{NC}$. We show how to determine whether $w \in \Sigma^*$ is a member of L in $O(\log n)$ steps using n processors, where $n = |w|$. We use a divide and conquer method adapted to Nick's Class. Let M be a DFA which accepts L. Let $Q = \{q_0, q_1, \ldots q_k\}$ be the set of states of M, q_0 the start state, $\delta : Q \times \Sigma \to Q$ the transition function, and $F \subseteq Q$ the set of final states. If $a \in \Sigma$, we write $\delta(a, \ ) : Q \to Q$. In the usual manner, we also write $\delta(x, \ ) : Q \to Q$ for any $x \in \Sigma^*$.

We identify a collection of $O(n)$ subproblems, each of which can be solved in $O(1)$ time by one processor, given the solutions to two smaller subproblems. Each subproblem is defined by a state $q_i \in Q$ and a string $u \in \Sigma^*$: the subproblem is to determine the value of $\delta(q_i, u) \in Q$. We define the *size* of that subproblem to be the length of u.

The subproblems of size 0 and 1 are trivial. Our method is to solve a subproblem of size $m \geq 2$ by combining the solutions to two subproblems of approximately half the size.

Let $x \in \Sigma^*$ of length $m \geq 2$. Write $x = yz$, where $|y| = \lfloor m/2 \rfloor$. For any $q_i \in Q$, let $q_j = \delta(q_i, y)$, and let $q_k = \delta(q_j, z)$. Then $\delta(q_i, x) = q_k$. Finally, we note that $w \in L$ if and only if $\delta(q_0, w) \in F$.

Our recursion is $O(\log n)$ deep, and at each recursive step, we perform $O(n)$ computations, each of which takes $O(1)$ time by one processor. Thus, the time complexity of our algorithm is $O(\log n)$ and the work complexity is $O(n \log n)$. This demonstrates that $L \in \mathcal{NC}$. $\square$