Source task and conventions

Construct NFAs for prefixes, suffixes, scattered subsequences and reversals of a regular language.

Conventional solution and proof

Let $N = (Q, \Sigma, E, s, F)$ be an NFA, possibly with epsilon transitions. *Prefixes:* keep the start and transitions and make every state from which some final state is reachable accepting. A path labelled u reaches such a state exactly when some continuation v gives $uv \in L$. *Suffixes:* add a fresh start with epsilon edges to every state reachable from s, keep the original accepting states. A new accepting path labelled v begins at a state reached by some prefix u, hence $uv \in L$; every such factorization yields a path. *Scattered subsequences:* for every labelled edge $q \xrightarrow{c} r$, add an epsilon edge $q \to r$. An accepting new path can skip any selected original symbols. Reinsert each skipped edge label to obtain a word of L, proving one direction; delete unwanted symbols along any old accepting path to prove the other. *Reversal:* reverse all edges (preserving their labels), make the old start the sole final state, and add a fresh start with epsilon edges to the old final states. Reversing an accepting path reverses its label, and reversing it back proves the converse. All constructions are finite NFAs, establishing regularity for all four operations.

Structural or analytic derivation

These are reachability and path transformations on a finite labelled graph. The prefix operation changes a terminal observation; the suffix operation changes the admissible initial states; subsequence closure adds silent edges; reversal transposes labelled adjacency. Counting accepted words through a resolvent can be added later, but epsilon ambiguity must first be resolved to avoid counting paths as distinct words.