

Regular Expressions and DFAs

William Schultz

November 12, 2021

Formally, a *language* is a set of strings $w \in \Sigma^*$, where Σ is some finite alphabet i.e. some finite set of atomic symbols. A language L is *regular* if it can be accepted by some deterministic finite automaton (DFA). That is, there is some DFA that can determine, for any string $w \in \Sigma^*$, whether $w \in L$ or $w \notin L$. A *regular expression* is one way to represent a regular language. It provides a concise form for expressing regular languages i.e. it defines a language as the set of strings accepted by that regular expression.

Any regular expression can be converted into an equivalent nondeterministic finite automaton (NFA), or an equivalent DFA. Recall that the translation from an NFA to DFA may, in the worst case, lead to an exponential blow up in the number of states. See Thompson's Construction algorithm for transforming a regular expression into an equivalent NFA, which can then be converted to a DFA e.g. via the *powerset construction*.