Abstract

This paper describes a method for constructing a minimal deterministic finite automaton (DFA) from a regular expression. It is based on a set of graph grammar rules for combining many graphs (DFA) to obtain another desired graph (DFA). The graph grammar rules are presented in the form of a parsing algorithm that converts a regular expression $R$ into a minimal
Construction of a Minimal Deterministic Finite Automaton from a Regular Expression

deterministic finite automaton M such that the language accepted by DFA M is same as the language described by regular expression R. The proposed algorithm removes the dependency over the necessity of lengthy chain of conversion, that is, regular expression --> NFA with ε-transitions --> NFA without ε-transitions --> DFA --> minimal DFA. Therefore the main advantage of our minimal DFA construction algorithm is its minimal intermediate memory requirements and hence, the reduced time complexity. The proposed algorithm converts a regular expression of size n in to its minimal equivalent DFA in \( O(n \log 2n) \) time. In addition to the above, the time complexity is further shortened to \( O(n \log n) \) for \( n \geq 75 \).

Reference

- Hagenah, C. and A. Muscholl [1998]. “Computing epsilon-free NFA from regular
- Thompson, K. [1968]. "Regular expression search algorithms". Communications of the ACM. vol. 11, no. 6, pp. 419-422.
Index Terms

Computer Science  Algorithms

Key words

Alphabet  Automaton Construction  Combined State Union

Concatenation

Kleene Closure

Minimization

Transition