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
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 \( \rightarrow \) NFA with \( \varepsilon \)-transitions \( \rightarrow \) NFA without \( \varepsilon \)-transitions \( \rightarrow \) DFA \( \rightarrow \) 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 \( O(n \log_2 n) \) time. In addition to the above, the time complexity is further shortened to \( O(n \log_e n) \) for \( n \geq 75 \).

**Reference**

- Hagenah, C. and A. Muscholl [1998]. “Computing epsilon-free NFA from regular
expressions in \(o(n \cdot \log^2(n))\) time”. In Proceedings of the 23rd International Symposium on

- Hopcroft, J. E. and J. Ullman [1979]. Introduction to Automata Theory, Languages and
- Hromkovic J., S. Seibert, and T. Wilke [2001]. "Translating regular expressions into small
4, pp. 565-588.
- Ilie L. and S. Yu [2003]. "Follow automata". Information and Computation. vol. 186, no. 1,
pp. 140-162.
efficient lexical processors using finite state techniques". Communications of the ACM. vol. 11,
no. 12, pp. 805-813.
- Leiss, E. [1980]. “Constructing a finite automaton for a given regular expression”. ACM
Special Interest Group on Algorithms and Computation Theory (ACM SIGACT News). vol. 12,
no. 3, pp. 81-87.
- Mayer, Ernst W., G. Schmidt, and G. Tinhofer (eds.) [1995]. Graph-Theoretic Concepts in
Computer Science. Lecture notes in Computer Science no. 903. Springer-Verlag,
Berlin/Heidelberg, New York.
- Möhring, R. H. (ed.) [1991]. Graph-Theoretic Concepts in Computer Science, 16th
International Workshop, WG '90, Berlin, Germany, June 20-22, 1990, Proceedings. Lecture
Notes in Computer Science no. 484. Springer. London, UK.
- Rytter, W. [1989]. “A note on optimal parallel transformations of regular expressions to
- Thompson, K. [1968]. “Regular expression search algorithms”. Communications of the
ACM. vol. 11, no. 6, pp. 419-422.
- Watson, B. [1995]. “Taxonomies and toolkits of regular language algorithms”. Ph.D.
Company, Inc. Boston, MA, USA.
- Yamamoto, H. [2005]. “New finite automata corresponding to semiextended regular
expressions”. Systems and Computers in Japan. vol. 36, no. 10, pp. 54-61.
regular expression into its Glushkov automaton”. Laboratoire d'Informatique de Rouen. vol. 215,
no. 1-2, pp. 69-87.
<table>
<thead>
<tr>
<th>Key words</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td>Automaton Construction</td>
<td>Combined State Union</td>
</tr>
<tr>
<td>Concatenation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kleene Closure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Index Terms

Computer Science

Algorithms