Simplifying Regular Expressions. A Quantitative Perspective

Hermann Gruber
Institut für Informatik, Universität Gießen
Arndtstraße 2, D-35392 Gießen, Germany
E-Mail: hermann.gruber@informatik.uni-giessen.de

Stefan Gulan
Fachbereich IV—Informatik, Universität Trier
Campus II, D-54296 Trier, Germany
E-Mail: gulan@uni-trier.de

We propose a new normal form for regular expressions which tightly bounds the ratio of two common size measures for regular expressions. We also give a conversion from regular expressions to ε-NFAs, which implicitly computes this normal form while maintaining an optimal ratio of expression-to-automaton-sizes. This allows us to resolve a problem posed by Ilie and Yu [4].

1 Definitions and Constructions

Regular expressions, *expressions* for brevity, may not contain ε or $\emptyset$ and are otherwise defined as usual with the additional operator ?, where $L(r^?) = \{\varepsilon\} \cup L(r)$. If every subexpression $s^?$ of r satisfies $\varepsilon \notin L(s)$, we call r *mildly simplified*. The number of leaves in the parse of r is denoted $\mathrm{alph}(r)$, the number of nodes $\mathrm{arpn}(r)$; further, let $\mathrm{rpn}(r)$ equal $\mathrm{arpn}(r)$ plus the number ?s occuring in r. Let $\mathrm{alph}(L) = \min\{\mathrm{alph}(r) \mid L(r) = L\}$; $\mathrm{rpn}(L)$ and $\mathrm{arpn}(L)$ are defined accordingly.

The operators $\circ$ and $\bullet$ are defined as: $a^\circ = a$, $(r+s)^\circ = r^\circ + s^\circ$, $r^{?\circ} = r^\circ$, $r^{*\circ} = r^{\circ *}$, if $\varepsilon \notin L(rs)$ then $(rs)^\circ = rs$, else $(rs)^\circ = r^\circ + s^\circ$; $a^\bullet = a$, $(r+s)^\bullet = r^\bullet + s^\bullet$, $(rs)^\bullet = r^\bullet s^\bullet$, $r^{*\bullet} = r^{\bullet\circ *}$, if $\varepsilon \in L(r)$ then $r^{?\bullet} = r^\bullet$, else $r^{?\bullet} = r^{\bullet ?}$. We call $r^\bullet$ the *strong star normal form* of r (cf. [1]).

We construct εNFAs from expressions by graph rewritings (Figs. 1,2), taken from [3], with additional precedences. Let $A(r)$ denote any automaton constructed this way, its size $|A(r)|$ is the combined number of states and transitions.

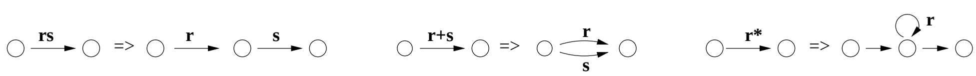

Figure 1: Introducing states/transitions while deconstructing the input in labels.

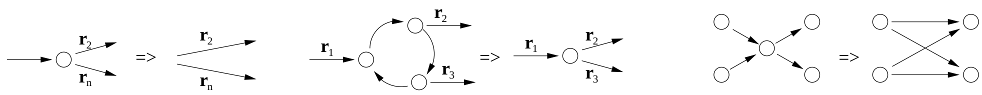

Figure 2: Removing redundant ε-transitions (unlabeled arcs) and incident states.

2 Results

Theorem 2.1. *Any regular language L satisfies* $\operatorname{rpn}(L) \leq 4\operatorname{alph}(L) - 1$.

This improves on previous bounds ([2, 4]) of $\operatorname{rpn}(L)$ wrt. $\operatorname{alph}(L)$. The concept of strong star normal form is crucial in the proof. This normal form is implicitly computed upon converting a mildly simplified expression into an εNFA.

Theorem 2.2. *Let r be mildly simplified, then* $A(r) = A(r^{\bullet})$.

The precondition poses no severe restriction, since any r can be transformed in linear time into a mildly simplified r', s.t. $L(r) = L(r')$ and $|A(r')| \leq |A(r)|$. The size of an εNFA constructed from such an expression is bounded from above as follows

Theorem 2.3. *Let r be mildly simplified, then $|A(r)| \leq 4\frac{2}{5}\operatorname{alph}(r) + 1$. This bound is tight for an infinite family of regular languages.*

Finally, we show that for some regular languages, the number of operators makes up for two thirds of even the shortest equivalent expression's size.

Theorem 2.4. *There are regular languages L_i such that* $\operatorname{alph}(L_i) \leq n$ *and* $\operatorname{arpn}(L_i) \geq 3n - 1$.

References

[1] A. Brüggemann-Klein. Regular Expressions into Finite Automata. *Theoretical Computer Science*, 120(2):197–213, 1993.

[2] K. Ellul, B. Krawetz, J. Shallit, and M. Wang. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics*, 10(4):407–437, 2005.

[3] S. Gulan and H. Fernau. An Optimal Construction of Finite Automata from Regular Expressions. In: *FSTTCS 08*, pp. 211–222, Dagstuhl Seminar Proceedings 08004, 2008.

[4] L. Ilie and S. Yu. Follow automata. *Information and Computation* 186(1):140–162, 2003.