DCFS 2009 (Draft)

Descriptional Complexity of Formal Systems
Deadline for submissions: April 14, 2009
Final versions: June 15, 2009

Quotient Complexity of Regular Languages

Janusz Brzozowski

David R. Cheriton School of Computer Science – University of Waterloo
Waterloo, ON – Canada N2L 3G1
brzozo@uwaterloo.ca

Abstract. The past research on the state complexity of operations on regular languages is examined, and a new approach based on an old method (derivatives of regular expressions) is presented. Since state complexity is a property of a language, it is appropriate to define it in formal-language terms as the number of distinct quotients of the language, and to call it "quotient complexity". The problem of finding the quotient complexity of a language $f(K, L)$ is considered, where K and L are regular languages and f is a regular operation, for example, union or concatenation. Since quotients can be represented by derivatives, one can find a formula for the typical quotient of $f(K, L)$ in terms of the quotients of K and L. To obtain an upper bound on the number of quotients of $f(K, L)$ all one has to do is count how many such quotients are possible, and this makes automaton constructions unnecessary. The advantages of this point of view are illustrated by many examples. Moreover, new general observations are presented to help in the estimation of the upper bounds on quotient complexity of regular operations.

Keywords: automaton, derivative, expression, language, operation, quotient, rational, regular, state complexity, upper bound

1 Introduction

It is assumed that the reader is familiar with the basic concepts of regular languages and finite automata, as described in many textbooks. General background material can be found in Dominique Perrin's [25] (1990) and Sheng Yu's [30] (1997) handbook articles; the latter has an introduction to state complexity. A more detailed treatment of state complexity can be found in Sheng Yu's survey [31]. The present paper concentrates on the complexity of basic operations on regular languages. Other aspects of complexity of regular languages and finite automata are discussed in [2, 6, 9, 14, 16, 18, 28, 29]; this list is not exhaustive, but it should give the reader a good idea of the scope of the work on this topic.

This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant no. OGP0000871.

2 State complexity or quotient complexity?

The English term *state complexity* of a regular language seems to have been introduced by Birget[1] [1] in 1991, and is now in common use. It is defined as the number of states in the minimal deterministic finite automaton (DFA) accepting the language [31]. There had been much earlier studies of this topic, but the term "state complexity" was not used. For example, in 1963 Lupanov [20] showed that the bound 2^n is tight for the conversion of nondeterministic finite automata (NFA's) to DFA's, and he used the term *slozhnost' avtomatov,* meaning *complexity of automata* representing the same set of words. The case of languages over a one-letter alphabet was studied in 1964 by Lyubich [21]. Lupanov's result is almost unknown in the English-language literature, and is often attributed to the 1971 paper by Moore [23]. In 1970, Maslov [22] studied the complexity of basic operations on regular languages, and stated without proof some tight bounds for these operations. In the introduction to his paper he states:

> An important characteristic of the complexity of these sets [of words] is the *number of states of the minimal representing automaton.*[2]

In 1981 Leiss [19] referred to *(deterministic) complexity* of languages. Some additional references to early works related to this topic can be found in [11, 31], for example.

A *language* is a subset of the free monoid Σ^* generated by a finite alphabet Σ. If state complexity is a property of a language, then why is it defined in terms of a completely different object, namely an automaton? Admittedly, regular languages and finite automata are closely related, but there is a more natural way to define this complexity of languages, as is shown below.

The *left quotient*, or simply *quotient* of a language L by a word w is defined as the language $w^{-1}L = \{x \in \Sigma^* \mid wx \in L\}$. The *quotient complexity* of L is the number of distinct languages that are quotients of L, and will be denoted by $\kappa(L)$ (*kappa* for both *kwotient* and *komplexity*). Quotient complexity is defined for *any* language, and so may be finite or infinite.

Since languages are sets, it is natural to define set operations on them. The following are typical set operations: *complement* ($\overline{L} = \Sigma^* \setminus L$), *union* ($K \cup L$), *intersection* ($K \cap L$), *difference* ($K \setminus L$), and *symmetric difference* ($K \oplus L$). A general *boolean operation* with two arguments is denoted by $K \circ L$. Since languages are also subsets of a monoid, it is also natural to define *product*, usually called *(con)catenation*, ($K \cdot L = \{w \in \Sigma^* \mid w = uv, u \in K, v \in L\}$), *star* ($K^* = \bigcup_{i\geq 0} K^i$), and *positive closure* ($K^+ = \bigcup_{i\geq 1} K^i$).

The operations union, product and star are called *rational* or *regular*. *Rational (or regular) languages* over Σ are those languages that can be obtained from the set $\{\emptyset, \{\varepsilon\}\} \cup \{\{a\} \mid a \in \Sigma\}$ of *basic languages*, where ε is the empty word, (or, equivalently, from another basis, such as the finite languages over Σ) using

[1] An error in [1] was corrected in [32].

[2] The emphasis is mine.

a finite number of rational operations. Since it is cumbersome to describe regular languages as sets—for example, one has to write $L = (\{\varepsilon\} \cup \{a\})^* \cdot \{b\}$—one normally switches to regular (or rational) expressions. These are the terms of the free algebra over the set $\Sigma \cup \{\emptyset, \varepsilon\}$ with function symbols[3] $\cup, \cdot$, and * [25]. For the example above, one writes $E = (\varepsilon \cup a)^* \cdot b$. The mapping $\mathcal{L}$ from this free algebra onto the algebra of regular languages is defined inductively as follows:

$$\mathcal{L}(\emptyset) = \emptyset, \quad \mathcal{L}(\varepsilon) = \{\varepsilon\}, \quad \mathcal{L}(a) = \{a\},$$

$$\mathcal{L}(E \cup F) = \mathcal{L}(E) \cup \mathcal{L}(F), \quad \mathcal{L}(E \cdot F) = \mathcal{L}(E) \cdot \mathcal{L}(F), \quad \mathcal{L}(E^*) = (\mathcal{L}(E))^*,$$

where E and F are regular expressions. The product symbol $\cdot$ is usually dropped, and languages are denoted by expressions without further mention of the mapping $\mathcal{L}$. Since regular languages are closed under complementation, complementation is treated here as a regular operator.

Because regular languages are defined by regular expressions, it is natural to use regular expressions also to represent their quotients; these expressions are their derivatives [4]. First, the *ε-function* of a regular expression L, denoted by L^ε, is defined as follows:

$$a^\varepsilon = \begin{cases} \emptyset, \text{ if } a = \emptyset, \text{ or } a \in \Sigma; \\ \varepsilon, \text{ if } a = \varepsilon. \end{cases} \tag{1}$$

$$(\overline{L})^\varepsilon = \begin{cases} \emptyset, \text{ if } L^\varepsilon = \varepsilon; \\ \varepsilon, \text{ if } L^\varepsilon = \emptyset. \end{cases} \tag{2}$$

$$(K \cup L)^\varepsilon = K^\varepsilon \cup L^\varepsilon, \ (KL)^\varepsilon = K^\varepsilon \cap L^\varepsilon, \ (L^*)^\varepsilon = \varepsilon. \tag{3}$$

One verifies that $\mathcal{L}(L^\varepsilon) = \{\varepsilon\}$ if $\varepsilon \in L$, and $\mathcal{L}(L^\varepsilon) = \emptyset$, otherwise.

The *derivative by a letter* $a \in \Sigma$ of a regular expression L is denoted by L_a and defined by structural induction:

$$b_a = \begin{cases} \emptyset, \text{ if } b \in \{\emptyset, \varepsilon\}, \text{ or } b \in \Sigma \text{ and } b \neq a; \\ \varepsilon, \text{ if } b = a. \end{cases} \tag{4}$$

$$(\overline{L})_a = \overline{L_a}, \ (K \cup L)_a = K_a \cup L_a, \ (KL)_a = K_a L \cup K^\varepsilon L_a, \ (L^*)_a = L_a L^*. \tag{5}$$

The *derivative by a word* $w \in \Sigma^*$ of a regular expression L is denoted by L_w and defined by induction on the length of w:

$$L_\varepsilon = L, \quad L_w = L_a, \text{ if } w = a \in \Sigma, \quad L_{wa} = (L_w)_a. \tag{6}$$

A derivative L_w is *accepting* if $\varepsilon \in L_w$; otherwise it is *rejecting*.

One can verify by structural induction that $\mathcal{L}(L_a) = a^{-1}L$, for all $a \in \Sigma$, and then by induction on the length of w that, for all $w \in \Sigma^*$,

$$\mathcal{L}(L_w) = w^{-1}L. \tag{7}$$

[3]The symbol $+$ is used instead of $\cup$ in [25].

Thus every derivative represents a unique quotient of L, but there may be many derivatives representing the same quotient.

Two regular expressions are *similar* [3, 4] if one can be obtained from the other using the following rules:

$$L \cup L = L, \quad K \cup L = L \cup K, \quad K \cup (L \cup M) = (K \cup L) \cup M, \tag{8}$$

$$L \cup \emptyset = L, \quad \emptyset L = L\emptyset = \emptyset, \quad \varepsilon L = L\varepsilon = L. \tag{9}$$

Upper bounds on the number of dissimilar derivatives, and hence on the quotient complexity, were derived in [3, 4]: If m and n are the quotient complexities of K and L, respectively, then

$$\kappa(\overline{L}) = \kappa(L), \quad \kappa(K \cup L) \le mn, \quad \kappa(KL) \le m2^n, \quad \kappa(L^*) \le 2^n - 1. \tag{10}$$

This immediately implies that the number of derivatives, and hence the number of quotients, of a regular language is finite.

It seems that the upper bounds in Equation (10), derived in 1962 [3, 4], were the first "state complexity" bounds to be found for the regular operations. Since the aim at that time was simply to show that the number of quotients of a regular language is finite, the tightness of the bounds was not considered.

Of course, the concepts above are related to the more commonly used ideas. A *deterministic finite automaton*, or simply *automaton,* is a tuple

$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F),$$

where Q is a finite, non-empty set of *states*, Σ is a finite, non-empty *alphabet*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. The transition function is extended to $\delta : Q \times \Sigma^* \to Q$ as usual. A word w is *recognized* (or *accepted*) by automaton $\mathcal{A}$ if $\delta(q_0, w) \in F$. It was proved by Nerode [24] that a language L is recognizable by a finite automaton if and only if L has a finite number of quotients.

The *quotient automaton* of a regular language L is $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{w^{-1}L \mid w \in \Sigma^*\}$, $\delta(w^{-1}L, a) = (wa)^{-1}L$, $q_0 = \varepsilon^{-1}L = L$, and $F = \{w^{-1}L \mid \varepsilon \in w^{-1}L\}$.

It should now be clear that the state complexity of a regular language L is the number of states in its quotient automaton, *i.e.*, the number $\kappa(L)$ of its quotients. This terminology change may seem trivial, but has some nontrivial consequences.

For convenience, derivative notation will be used to represent quotients, in the same way as regular expressions are used to represent regular languages.

By convention, L_w^ε always means $(L_w)^\varepsilon$.

3 Derivation of bounds using quotients

Since languages over one-letter alphabets have very special properties, we usually assume that the alphabet has at least two letters. The complexity of operations on unary languages has been studied in [26, 31].

In the literature on state complexity, it is assumed that automata $\mathcal{A}$ and $\mathcal{B}$ accepting languages K and L, respectively, are given. An assumption has to be made that the automata are "complete", *i.e.*, that for each $q \in Q$ and $a \in \Sigma$, $\delta(q, a)$ is defined [33]. In particular, if a "dead" or "sink" state, which accepts no words, is present, one has to check that only one such state is included [7]. Also, every state must be "useful" in the sense that it appears on some accepting path [8].

Suppose that a bound on the state complexity of $f(K, L)$ is to be computed, where f is some regular operation. In some cases a DFA accepting $f(K, L)$ is constructed directly, (*e.g.*, Theorems 2.3 and 3.1 in [33]), or an NFA with multiple initial states is used, and then converted to a DFA by the subset construction (*e.g.*, Theorem 4.1 in [33]). Sometimes an NFA with empty-word transitions is used and then converted to a DFA [29]. The constructed automata then have to be proved minimal.

Much of this is unnecessary. If quotients are used, the problem of completeness does not arise, since all the quotients of a language are included. A quotient is either empty or "useful". If the empty quotient is present, then it appears only once. Since quotients are distinct languages, the set of quotients of a language is always minimal. To find an upper bound on the state complexity, instead of constructing an automaton for $f(K, L)$, we need only find a regular expression for the typical quotient, and then do some counting. This is illustrated below for the basic regular operations.

3.1 Bounds for basic operations

The following are some useful formulas for the derivatives of regular expressions:

Theorem 1. *If K and L are regular expressions, then*

$$(\overline{L})_w = \overline{L_w}, \tag{11}$$

$$(K \circ L)_w = K_w \circ L_w, \tag{12}$$

$$(KL)_w = K_w L \cup K^\varepsilon L_w \cup \left(\bigcup_{\substack{w=uv \\ u,v\in\Sigma^+}} K_u^\varepsilon L_v \right). \tag{13}$$

For the Kleene star, $(L^*)_\varepsilon = \varepsilon \cup LL^*$, and for $w \in \Sigma^+$,

$$(L^*)_w = \left(\bigcup_{\substack{w=uv \\ u,v\in\Sigma^*}} (L^*)_u^\varepsilon L_v \right) L^*. \tag{14}$$

Proof: Consider first the boolean operations. Since $(K \cup L)_w = K_w \cup L_w$, and $(\overline{L})_w = \overline{L_w}$, it follows that $(K \circ L)_w = K_w \circ L_w$.

Equation 13 is easily verified by induction on the length $|w|$ of a word $w \in \Sigma^*$. Thus $(KL)_w$ consists of $K_w L$ and a union (possibly empty) of derivatives of L. When w is in K, then $\varepsilon \in K_w$ and L is added to the union.

For the star, the claim is obvious when $w = \varepsilon$. For $w \neq \varepsilon$, we first prove that

$$(L^*)_w = \left(L_w \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} (L^*)^\varepsilon_u L_v \right) L^* \tag{15}$$

by induction on $|w|$. Let $M = L^*$. For $w = a \in \Sigma$, we have $M_a = L_a L^* = L_a M$, by definition. This agrees with (15), because there is no decomposition $a = uv$ with $u, v \in \Sigma^+$. Now assume that (15) holds for w, and consider wa:

$$\begin{aligned} M_{wa} &= \left(L_{wa} \cup L^\varepsilon_w L_a \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L_{va} \cup M^\varepsilon_u L^\varepsilon_v L_a \right) M \\ &= \left(L_{wa} \cup \left(L^\varepsilon_w \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L^\varepsilon_v \right) L_a \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L_{va} \right) M. \end{aligned}$$

From (15), for $w \neq \varepsilon$, $M^\varepsilon_w = L^\varepsilon_w \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L^\varepsilon_v$; thus we have

$$M_{wa} = \left(L_{wa} \cup M^\varepsilon_w L_a \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L_{va} \right) M = \left(L_{wa} \cup \bigcup_{\substack{wa=xy \\ x,y \in \Sigma^+}} M^\varepsilon_x L_y \right) M.$$

So the induction step goes through, and we have (15). Note that $M = MM$; thus $w \in M$ implies $M_w = (L^*)_w \supseteq L^* \supseteq LL^* = LM$, and we have proved a useful alternate version of (15):

$$M_w = \left(L_w \cup M^\varepsilon_w L \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L_v \right) M = S(w)M, \tag{16}$$

where $S(w) = \left(L_w \cup M^\varepsilon_w L \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} M^\varepsilon_u L_v \right)$. Finally, note that $L_w = M^\varepsilon_\varepsilon L_w$; hence $S(w) = \left(\bigcup_{\substack{w=uv \\ u,v \in \Sigma^*}} M^\varepsilon_u L_v \right)$ and we have (14). □

Theorem 1 can be applied to obtain upper bounds on the complexity of operations. In Theorem 2 below, the second part is a slight generalization of the bound in Theorem 4.3 of [33]. The third and fourth parts are reformulations of the bounds in Theorem 2.3 and 2.4, and of Theorem 3.1 of [33]:

Theorem 2. *For any languages K and L with $\kappa(K) = m$ and $\kappa(L) = n$,*

(I) $\kappa(\overline{L}) = n$.

(II) $\kappa(K \circ L) \leq mn$.

(III) *Suppose K has k accepting quotients and L has l accepting quotients.*
- (A) *If $k = 0$ or $l = 0$, then $\kappa(KL) = 1$.*
- (B) *If $k, l > 0$ and $n = 1$, then $\kappa(KL) \leq m - (k - 1)$.*
- (C) *If $k, l > 0$ and $n > 1$, then $\kappa(KL) \leq m2^n - k2^{n-1}$.*

(IV) (A) *If $n = 1$, then $\kappa(L^*) \leq 2$.*
- (B) *If $n > 1$ and L_ε is the only accepting quotient of L, then $\kappa(L^*) = n$.*
- (C) *If $n > 1$ and L has $l > 0$ accepting quotients not equal to L, then $\kappa(L^*) \leq 2^{n-1} + 2^{n-l-1}$.*

Proof: The first part is well-known, and the second follows from (12).

For the product, if $k = 0$ or $l = 0$, then $KL = \emptyset$ and $\kappa(KL) = 1$. Thus assume that $k, l > 0$. If $n = 1$, then $L = \Sigma^*$ and $w \in K$ implies $(KL)_w = \Sigma^*$. Thus all k accepting quotients of K produce the one quotient Σ^* in KL. For each rejecting quotient of K, we have two choices for the union of quotients of L in (13): the empty union or Σ^*. If we choose the empty union, we can have at most $m - k$ quotients of KL. Choosing Σ^* results in $(KL)_w = \Sigma^*$, which has been counted already. Altogether, there are at most $1 + m - k$ quotients of KL. Suppose now that $k, l > 0$ and $n > 1$. If $w \notin K$, then we can choose K_w in $m - k$ ways, and the union of quotients of L in 2^n ways. If $w \in K$, then we can choose K_w in k ways, and the set of quotients of L in 2^{n-1} ways, since L is then always present. Thus we have $(m - k)2^n + k2^{n-1}$.

For the star, if $n = 1$, then $L = \emptyset$ or $L = \Sigma^*$. In the first case, $L^* = \varepsilon$, and $\kappa(L^*) = 2$; in the second case, $L^* = \Sigma^*$ and $\kappa(L^*) = 1$. Now suppose that $n > 1$; hence L has at least one accepting quotient. If L is the only accepting quotient of L, then $L^* = L$ and $\kappa(L^*) = \kappa(L)$.

Now assume that $n > 1$ and $l > 0$. From (14), every quotient of L^* by a non-empty word is a union of a subset of quotients of L, followed by L^*. Moreover, that union is non-empty, because $(L^*)^\varepsilon_\varepsilon L_w$ is always present. We have two cases:

(I) Suppose L is rejecting. Then L has l accepting quotients.
- (A) If no accepting quotient of L is included in the subset, then there are $2^{n-l} - 1$ such subsets possible, the union being non-empty because L_w is always included.
- (B) If an accepting quotient of L is included, then $\varepsilon \in (L^*)_w$, $(L^*)^\varepsilon_w = \varepsilon$, and $L = (L^*)^\varepsilon_w L_\varepsilon$ is also included. We have $2^l - 1$ non-empty subsets of accepting quotients of L and 2^{n-l-1} subsets of rejecting quotients, since L is not counted.

Adding 1 for $(L^*)_\varepsilon$, we have a total of $2^{n-l} - 1 + (2^l - 1)2^{n-l-1} + 1 = 2^{n-1} + 2^{n-l-1}$.

(II) Suppose L is accepting. Then L has $l + 1$ accepting quotients.
- (A) If there is no accepting quotient, there are $2^{n-l-1} - 1$ non-empty subsets of rejecting quotients.
- (B) If an accepting quotient of L is included, then L is included, and 2^{n-1} subsets can be added to L.

We need not add $(L^*)_\varepsilon$, since $\epsilon \cup LL^* = LL^*$ in this case, and this has already been counted. The total is $2^{n-1} + 2^{n-l} - 1$.

The worst-case bound of $2^{n-1} + 2^{n-l-1}$ occurs in the first case only. □

3.2 Witnesses to bounds for basic operations

Finding witness languages showing that a bound is tight is often challenging. However, once a guess is made, the verification can be done using quotients.

Let $|w|_a$ be the number of a's in w, for $a \in \Sigma$ and $w \in \Sigma^*$. Unary, binary, and ternary languages are languages over a one-, two-, and three-letter alphabet, respectively.

- **Union and Intersection** If we have a bound for intersection, then for union we can use the fact that $\kappa(\overline{K} \cup \overline{L}) = \kappa(\overline{\overline{K} \cup \overline{L}}) = \kappa(K \cap L)$; thus the pair $(\overline{K}, \overline{L})$ is a witness for union. Similarly, given a witness for union, we also have a witness for intersection.

  The upper bound mn for the complexity of intersection was observed in 1957[4] by Rabin and Scott [27]. Binary languages $K = \{w \in \{a,b\}^* \mid |w|_a \equiv m-1 \bmod m\}$ and $L = \{w \in \{a,b\}^* \mid |w|_b \equiv n-1 \bmod n\}$ have quotient complexities m and n, respectively. In 1970 Maslov [22] stated without proof that $K \cup L$ meets this upper bound mn. Yu, Zhuang and K. Salomaa [33], used similar languages $K' = \{w \in \{a,b\}^* \mid |w|_a \equiv 0 \bmod m\}$ and $L' = \{w \in \{a,b\}^* \mid |w|_b \equiv 0 \bmod n\}$ for intersection, apparently unaware of [22]. Hricko, Jirásková and Szabari [11] showed that a complete hierarchy of quotient complexities of binary languages exists between the minimum complexity 1 and the maximum complexity mn. More specifically, it was proved that for any integers m, n, α such that $m \geq 2$, $n \geq 2$ and $1 \leq \alpha \leq mn$, there exist binary[5] languages K and L such that $\kappa(K) = m$, $\kappa(L) = n$, and $\kappa(K \cup L) = \alpha$, and the same holds for intersection.

  For a one-letter alphabet $\Sigma = \{a\}$, Yu showed that the bound can be reached if m and n are relatively prime [31]. The witnesses are $K'' = (a^m)^*$ and $L'' = (a^n)^*$. For other cases, see the paper by Pighizzini and Shallit [26].
- **Set difference** For set difference we have $\kappa(K' \setminus \overline{L'}) = \kappa(K' \cap L')$; thus the pair $(K', \overline{L'})$ is a witness.
- **Symmetric difference** For symmetric difference, let $m, n \geq 1$, let $K = (b^*a)^{m-1}(a \cup b)^*$ and let $L = (a^*b)^{n-1}(a \cup b)^*$. There are mn words of the form $a^i b^j$, where $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$. We claim that all the quotients of $K \oplus L$ by these words are distinct. Let $x = a^i b^j$ and $y = a^k b^l$. If $i < k$, let $u = a^{m-1-k} b^n$. Then $xu \notin K$, $yu \in K$, and $xu, yu \in L$,

[4]The work was done in 1957, but published in 1959.

[5]The proof in [11] is for ternary languages; a proof for the binary case can be found in [10].

showing that $xu \in K \oplus L$, and $yu \notin K \oplus L$, *i.e.*, that $(K \oplus L)_x \neq (K \oplus L)_y$. Similarly, if $j < l$, let $v = a^m b^{n-1-l}$. Then $xv \in K \oplus L$, but $yv \notin K \oplus L$. Therefore all the quotients of $K \oplus L$ by these mn words are distinct.

For a one-letter alphabet, the witnesses are K'' and L'' as in the case of union above.

- **Other boolean functions** There are six more two-variable boolean functions that depend on both variables: $\overline{K} \cup \overline{L} = \overline{K \cap L}$, $\overline{K} \cap \overline{L} = \overline{K \cup L}$, $\overline{K} \cup L = \overline{K \setminus L}$, $\overline{K} \cap L = L \setminus K$, $K \cup \overline{L} = \overline{L \setminus K}$, and $\overline{K \oplus L}$. The witnesses for these functions can be found using the four functions above.

- **Product** The upper bound of $m2^n - 2^{n-1}$ was given by Maslov in 1970 [22], and he stated without proof that it is tight for binary languages $K = \{w \in \{a,b\}^* \mid |w|_a \equiv m-1 \bmod m\}$ and $L = (a^*b)^{n-2}(a \cup b)(b \cup a(a \cup b))^*$. The bound was refined by Yu, Zhuang and K. Salomaa [33] to $m2^n - k2^{n-1}$, where k is the number of accepting quotients of K. Jirásek, Jirásková and Szabari [12] proved that, for any integers m, n, k such that $m \geq 2$, $n \geq 2$ and $0 < k < m$, there exist binary languages K and L such that $\kappa(K) = m$, $\kappa(L) = n$, and $\kappa(KL) = m2^n - k2^{n-1}$. Furthermore, Jirásková [15] proved that, for all m, n, and α such that either $n = 1$ and $1 \leq \alpha \leq m$, or $n \geq 2$ and $1 \leq \alpha \leq m2^n - 2^{n-1}$, there exist languages K and L with $\kappa(K) = m$ and $\kappa(L) = n$, defined over a growing alphabet, such that $\kappa(KL) = \alpha$.

  For a one-letter alphabet, mn is a tight bound for product if m and n are relatively prime [33]. The witnesses are $K = (a^m)^* a^{m-1}$ and $L = (a^n)^* a^{n-1}$. See also [26].

- **Star** Maslov [22] stated[6] without proof that $\kappa(L^*) \leq 2^{n-1} + 2^{n-2}$, and provided a binary language meeting this bound. Three cases were considered by Yu, Zhuang and K. Salomaa [33]:

  * $n = 1$. If $L = \emptyset$, then $\kappa(L) = 1$ and $\kappa(L^*) = 2$. If $L = \Sigma^*$, then $\kappa(L^*) = 1$.
  * $n = 2$. $L = \{w \in \{a,b\}^* \mid |w|_a \equiv 1 \bmod 2\}$ has $\kappa(L) = 2$, and $\kappa(L^*) = 3$.
  * $n > 2$. Let $\Sigma = \{a,b\}$. Then $L = (b \cup a\Sigma^{n-1})^* a \Sigma^{n-2}$ has n quotients, one of which is accepting, and $\kappa(L^*) = 2^{n-1} + 2^{n-2}$. This example is different from Maslov's.

  Moreover, Jirásková [13] proved that, for all integers n and α with either $1 = n \leq \alpha \leq 2$, or $n \geq 2$ and $1 \leq \alpha \leq 2^{n-1} + 2^{n-2}$, there exists a language L over a 2^n–letter alphabet such that has $\kappa(L) = n$ and $\kappa(L^*) = \alpha$.

  For a one-letter alphabet, $n^2 - 2n + 2$ is a tight bound for star [33]. The witness is $L'' = (a^n)^* a^{n-1}$. See also [26].

[6]The bound is incorrectly stated as $2^{n-1} + 2^{n-2} - 1$, but the example is correct.

4 Generalization of "non-returning" state

A quotient L_w of a language L is *uniquely reachable* if $L_x = L_w$ implies that $x = w$. If L_{wa} is uniquely reachable for $a \in \Sigma$, then so is L_w. Thus, if L has a uniquely reachable quotient, then L itself is uniquely reachable by the empty word, *i.e.*, the minimal automaton of L is *non-returning*[7]. Thus the set of uniquely reachable quotients of L is a tree with root L, if it is non-empty.

We now apply the concept of uniquely reachable quotients to boolean operations and product.

Theorem 3. *Suppose $\kappa(K) = m$, $\kappa(L) = n$, K and L have m_u and n_u uniquely reachable quotients, respectively, and there are r words w_i such that both K_{w_i} and L_{w_i} are uniquely reachable. If $\circ$ is a boolean operator, then*

$$\kappa(K \circ L) \leq mn - (\alpha + \beta + \gamma), \text{ where} \tag{17}$$

$$\alpha = r(m+n) - r(r+1);\ \beta = (m_u - r)(n - (r+1));\ \gamma = (n_u - r)(m - m_u - 1). \tag{18}$$

If K has k accepting quotients, t of which are uniquely reachable, and s rejecting uniquely reachable quotients, then

$$\kappa(KL) \leq m2^n - k2^{n-1} - s(2^n - 1) - t(2^{n-1} - 1). \tag{19}$$

Proof: Suppose the quotients of K and L are $K_1, \dots, K_m$ and $L_1, \dots, L_n$, respectively. Without loss of generality, we can assume that this numbering is such that $K_{w_i} = K_i$ and $L_{w_i} = L_i$, for each $i = 1, \dots, r$. Moreover, assume that the remaining $m_u - r$ uniquely reachable quotients of K are numbered $K_{r+1}, \dots, K_{m_u}$, and the remaining $n_u - r$ uniquely reachable quotients of L are $L_{m_u+1}, \dots, L_{m_u+(n_u-r)}$.

Because of (12), the number of quotients of $K \circ L$ is bounded from above by the number of pairs (K_w, L_w). Consider first $K_{w_1} = K_1$ and $L_{w_1} = L_1$. Since K_1 can appear only with L_1 in any pair, we know that the $n - 1$ pairs of the form (K_1, L_j), $j > 1$, and the $m - 1$ pairs of the form (K_i, L_1), $i > 1$ will never appear. Next, if K_2 and L_2 are both uniquely reachable, then we know that the $n - 2$ pairs (K_2, L_j), $j > 2$, and $m - 2$ pairs (K_i, L_1), $i > 2$ will not appear. Finally, the $n - r$ pairs (K_r, L_j), $j > r$, and the $m - r$ pairs (K_i, L_r), $i > r$, will not appear. Thus, we have the following reduction due to the r pairs: $\alpha = m - 1 + \dots + m - r + n - 1 + \dots + n - r = r(m + n) - r(r + 1)$.

We have now examined r of the m_u quotients that are uniquely reachable in K. For each of the remaining $m_u - r$ uniquely reachable quotients of K we can eliminate $n-(r+1)$ pairs of the form (K_i, L_j), $r+1 < i < m_u$, where $K_w = K_i$, but $L_w \neq L_j$. This yields the second part of the reduction: $\beta = (m_u - r)(n - (r+1))$.

For each of the $n_u - r$ uniquely reachable quotients of L, we can eliminate $m - m_u - 1$ pairs of the form (K_i, L_j), $r + 1 < i < m_u$, where $L_w = L_j$, but $K_w \neq K_i$, obtaining the third reduction: $\gamma = (n_u - r)(m - m_u - 1)$.

[7]The term "non-returning" suggests that once a state is left it cannot be visited again. However, such non-returning states are not necessarily uniquely reachable.

For the product, each quotient of KL corresponds to one of the m quotients of K together with a subset of the n quotients of L. If K_w is rejecting, then it can appear with 2^n subsets of quotients of L. This gives $(m-k)2^n$ such possible quotients of KL. If a rejecting quotient of K is uniquely reachable, then it can appear with only one subset. Hence there is a savings of $s(2^n-1)$. If K_w is accepting, then L is always present in $(KL)_w$. Hence there are at most $k2^{n-1}$ such possible quotients of KL. But, if t of the accepting quotients of K are uniquely reachable, then each can appear with only one subset of quotients of L that does not include L, for a saving of $t(2^{n-1}-1)$. Altogether, we have at most $(m-k)2^n - s(2^n-1) + k2^{n-1} - t(2^{n-1}-1)$ quotients of KL, as claimed. □

The following observation was stated for union and intersection of finite languages in [31]; we add the suffix-free case:

Corollary 4. *If K and L are non-empty and finite or suffix-free and $\kappa(K) = m > 1$, $\kappa(L) = n > 1$, then $\kappa(K \circ L) \leq mn - (m+n-2)$.*

Proof: The quotients K_ε and L_ε are uniquely reachable if K and L are finite. This also holds if K and L are suffix-free. For suppose $L_\varepsilon = L_w$. Since L is non-empty, we have $\varepsilon \in L_x$ for some $x \in \Sigma^*$. Then also $\varepsilon \in L_{wx}$ and both wx and x are in L, contradicting suffix-freeness. □

The bound $mn-(m+n-2)$ for union of suffix-free languages was shown to be tight for quinary languages by Han and Salomaa [7]. It is also tight for the binary languages $K = a((ba^*)^{m-3}b)^*(ba^*)^{m-3}$ and $L = a((a \cup b)^{n-3}b)^*(a \cup b)^{n-3}$, as shown recently by Jirásková and Olejár [17].

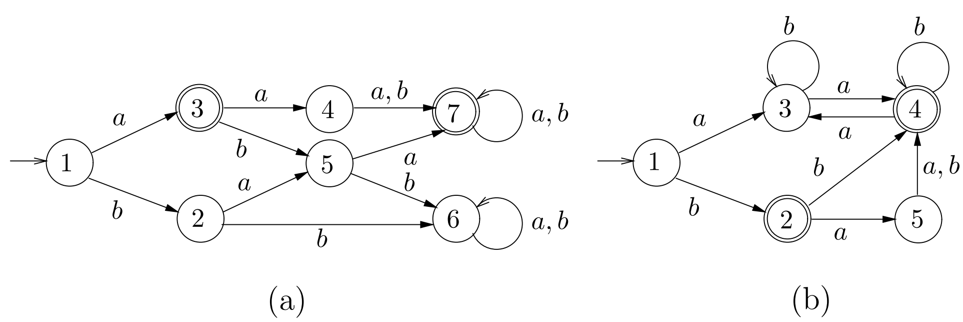

Figure 1. Illustrating unique reachability.

Example 5. The automaton of Fig. 1 (a) accepting K has $m = 7$ and four uniquely reachable states: 1, 2, 3, and 4. The automaton of Fig. 1 (b) accepting L has $n = 5$ and three uniquely reachable states: 1, 2, and 5. In pairs $(1,1)$ and $(2,2)$ both states are reachable by the same word (ε and b, respectively); hence $r = 2$.

The $m \times n = 7 \times 5$ table of all pairs is shown below, where uniquely reachable states are in boldface type. We have $\alpha = 18$, where the removed pairs are all the pairs in the first two rows and columns, except $(1,1)$ and $(2,2)$. Next, $\beta = 4$, and we remove the pairs $(3,4)$, $(3,5)$, $(4,3)$ and $(4,5)$ from rows 3 and 4. Finally, $\gamma = 2$, and we remove the pairs $(6,5)$ and $(7,5)$ from column 5.

$$\begin{array}{ccccc}
(\mathbf{1},\mathbf{1}) & (1,2) & (1,3) & (1,4) & (1,5) \\
(2,1) & (\mathbf{2},\mathbf{2}) & (2,3) & (2,4) & (2,5) \\
(3,1) & (3,2) & (\mathbf{3},3) & (3,4) & (3,5) \\
(4,1) & (4,2) & (4,3) & (\mathbf{4},4) & (4,5) \\
(5,1) & (5,2) & (5,3) & (5,4) & (5,\mathbf{5}) \\
(6,1) & (6,2) & (6,3) & (6,4) & (6,5) \\
(7,1) & (7,2) & (7,3) & (7,4) & (7,5)
\end{array}$$

Altogether, we have removed 24 states from $K \circ L$, leaving 11 possibilities. The minimal automaton of $K \cup L$ has 8 states. Notice that state 7 corresponds to the quotient Σ^*. Since $\Sigma^* \cup L_w = \Sigma^*$ for all w, we need to account for only one pair $(7,x)$, and we could remove the remaining four pairs. However, we have already removed pair $(7,5)$ by Theorem 3. Hence, there are only three pairs left to remove, and we have an automaton with 8 states. More will be said about the effects of Σ^* later.

It is also possible to use Theorem 3 if K has some uniquely reachable quotients and L has none, or when L is completely unknown. If $n_u = 0$, then $r = 0$, $\alpha = 0$, $\beta = m_u(n-1)$, and $\gamma = 0$. Then, for any L,

$$\kappa(K \circ L) \leq mn - m_u(n-1). \tag{20}$$

For example, for any L with $n = 101$ and K as in Fig. 1 (a), $\kappa(K \cap L) \leq 307$, instead of the general bound 707.

Let K and L be the automata of Fig. 1 (a) and (b), respectively. Then the general bound on $\kappa(KL)$ is 192. Here $s = 3$ (states 1, 2, and 4), and $t = 1$ (state 3). By Theorem 3 the bound is reduced by $93 + 15 = 108$ to 84. The actual quotient complexity of KL is 14.

The general bound for LK is 512, the reduced bound is 195, and the actual quotient complexity is 12. ■

5 Languages with ε, Σ^+, $\emptyset$, or Σ^* as quotients

In this section we consider the effects of the presence of special quotients in a language. In particular, we study the quotients ε, Σ^+, $\emptyset$, and Σ^*.

Theorem 6. *If $\kappa(K) = m$, $\kappa(L) = n$, and K and L have $k > 0$ and $l > 0$ accepting quotients, respectively, then*

(I) *If K and L have ε as a quotient, then*
- $\kappa(K \cup L) \leq mn - 2$.
- $\kappa(K \cap L) \leq mn - (2m + 2n - 6)$.
- $\kappa(K \setminus L) \leq mn - (m + 2n - k - 3)$.
- $\kappa(K \oplus L) \leq mn - 2$.

(II) *If K and L have Σ^+ as a quotient, then*
- $\kappa(K \cap L) \le mn - 2$.
- $\kappa(K \cup L) \le mn - (2m + 2n - 6)$.
- $\kappa(K \setminus L) \le mn - (2m + l - 3)$.
- $\kappa(K \oplus L) \le mn - 2$.

(III) *If K and L have $\emptyset$ as a quotient, then*
- $\kappa(K \cap L) \le mn - (m + n - 2)$.
- $\kappa(K \setminus L) \le mn - n + 1$.

(IV) *If K and L have Σ^* as a quotient, then*
- $\kappa(K \cup L) \le mn - (m + n - 2)$.
- $\kappa(K \setminus L) \le mn - m + 1$.

(V)
- *If L has ε as a quotient, then $\kappa(L^R) \le 2^{n-2} + 1$.*
- *If L has Σ^+ as a quotient, then $\kappa(L^R) \le 2^{n-2} + 1$.*
- *If L has $\emptyset$ as a quotient, then $\kappa(L^R) \le 2^{n-1}$.*
- *If L has Σ^* as a quotient, then $\kappa(L^R) \le 2^{n-1}$.*
- *Moreover, the effect of these quotients on complexity is cumulative. For example, if L^R has both $\emptyset$ and Σ^*, then $\kappa(L^R) \le 2^{n-2}$, if L^R has both $\emptyset$ and Σ^+, then $\kappa(L^R) \le 2^{n-3} + 1$, etc.*

Proof:

(I) The quotient of ε by every non-empty word is $\emptyset$; thus, if L has ε, it also has $\emptyset$. Since $\varepsilon \cup \varepsilon = \emptyset \cup \varepsilon = \varepsilon \cup \emptyset = \varepsilon$, we subtract two quotients for union.

For intersection, for any K_u and L_v, $K_u \cap \emptyset = \emptyset \cap L_v = \emptyset$. This eliminates $m-1+n-1$ possibilities. Moreover, if K_u, L_v are rejecting, then $K_u \cap \varepsilon = \varepsilon \cap L_v = \emptyset$. This removes another $m-k-1+n-l-1$ possibilities. If K_u, L_v are accepting, then $K_u \cap \varepsilon = \varepsilon \cap L_v = \varepsilon$, and $k-1+l-1$ quotients are removed. Altogether, $\kappa(K \cap L) \le mn - (2m + 2n - 6)$.

For set difference, since K has $\emptyset$ and $\emptyset \cap \overline{L_w} = \emptyset$ for all w, this saves $n-1$ quotients. Since K has ε, if $\varepsilon \in \overline{L_w}$, then $\varepsilon \cap \overline{L_w} = \varepsilon$; otherwise $\varepsilon \cap \overline{L_w} = \emptyset$. This saves another $n-1$ quotients. If L has ε and $\emptyset$, then $\overline{L}$ has Σ^+ and Σ^*. For each rejecting quotient K_w, we have $K_w \cap \Sigma^* = K_w \cap \Sigma^+$. This saves another $(m-k)-1$ quotients of $K \setminus L$.

Since $\emptyset \oplus \varepsilon = \varepsilon \oplus \emptyset = \varepsilon$, and $\varepsilon \oplus \varepsilon = \emptyset \oplus \emptyset = \emptyset$ we can subtract two quotients for symmetric difference.

(II) The proofs are dual to those of (I):

The quotient of Σ^+ by every non-empty word is Σ^*; thus, if L has Σ^+, it also has Σ^*. Since $\Sigma^+ \cap \Sigma^+ = \Sigma^* \cap \Sigma^+ = \Sigma^+ \cap \Sigma^* = \Sigma^+$, we subtract two quotients for intersection.

For union, for any K_u and L_v, $K_u \cup \Sigma^* = \Sigma^* \cup L_v = \Sigma^*$. This eliminates $m-1+n-1$ possibilities. Moreover, if K_u, L_v are accepting, then $K_u \cup \Sigma^+ = \Sigma^+ \cup L_v = \Sigma^*$. This removes another $k-1+l-1$

possibilities. If K_u, L_v are rejecting, then $K_u \cup \Sigma^+ = \Sigma^+ \cup L_v = \Sigma^+$, removing $m-k-1+n-l-1$ quotients.

For set difference, if L has Σ^+ and Σ^*, then $\overline{L}$ has ε and $\emptyset$. Since $K_w \cap \emptyset = \emptyset$ for all w, this saves $m-1$ quotients. Also, if $\varepsilon \in K_w$, then $K_w \cap \varepsilon = \varepsilon$, and otherwise $K_w \cap \varepsilon = \emptyset$. This saves another $m-1$ quotients. Finally, for each accepting quotient L_w, we have $\Sigma^* \cap \overline{L_w} = \Sigma^+ \cap \overline{L_w}$. This saves another $l-1$ quotients.

Since $\Sigma^* \oplus \Sigma^+ = \Sigma^+ \oplus \Sigma^* = \varepsilon$ and $\Sigma^* \oplus \Sigma^* = \Sigma^+ \oplus \Sigma^+ = \emptyset$, we can subtract two quotients for symmetric difference.

(III) Since $K_w \cap \emptyset = \emptyset \cap \overline{L_w} = \emptyset \cap \emptyset$, we can subtract $(m-1)+(n-1)$ quotients. For difference, since $\emptyset \cap \overline{L_w} = \emptyset$ for all w, this saves $n-1$ quotients.

(IV) Dual to III.

(V) The subset construction is used to convert the nondeterministic automaton $\mathcal{N}^R$, which is the reverse of the quotient automaton $\mathcal{A}$ recognizing L, to a deterministic automaton $\mathcal{A}^R$ recognizing L^R. In that construction, the state of $\mathcal{N}^R$ corresponding to $\emptyset$ is not reachable from the set of initial states, and the state corresponding to ε appears only in the initial set of states, but is not reachable by any non-empty word from that set. Dually, the state corresponding to Σ^* appears in all the sets reachable from the set of initial states, and Σ^+ appears in all the sets reachable from the set of initial states, except the set of initial states.

□

Corollary 7. *If K and L are both non-empty and both suffix-free with $\kappa(K) = m$ and $\kappa(L) = n$, then $\kappa(K \cap L) \le mn - 2(m+n-3)$.*

Proof: We have shown in Corollary 4 that $\kappa(K \cap L) \le mn - (m+n-2)$ by removing all pairs (K, L_u) and (K_v, L), where $L_u \ne L$ and $K_v \ne K$. If K is suffix-free, then it must have $\emptyset$ as a quotient [7], and the same holds for L. Thus all pairs of the form $(K_w, \emptyset)$ and $(\emptyset, L_w)$ are equivalent to $(\emptyset, \emptyset)$. We have already removed $(K, \emptyset)$ and $(\emptyset, L)$ by unique reachability. Hence we can remove a further $(m+n-2)-2$ quotients, for a total of $2(m+n-3)$. □

It is shown in [7] that the bound can be reached with

$$K = \{\#w \mid w \in \{a,b\}^*, |w|_a \equiv 0 \bmod m-2\},$$

$$L = \{\#w \mid w \in \{a,b\}^*, |w|_b \equiv 0 \bmod n-2\}.$$

It was recently proved in [17] that this bound can be reached by the binary languages given after Corollary 4.

Proposition 8. *If $\kappa(L) = n \ge 3$, L has $l > 0$ accepting quotients, and L has ε as a quotient, then $\kappa(L^*) \le 2^{n-3} + 2^{n-l-1} + 1$.*

Proof: If L has ε, then it also has $\emptyset$. From (14), every quotient of L^* by a non-empty word is a union of a non-empty subset of quotients of L, followed by L^*. We have two cases:

(I) Suppose L is rejecting.

(A) If no accepting quotient is included, then there are $2^{n-l-1} - 1$ non-empty subsets of non-empty rejecting quotients plus the subset consisting of the empty quotient alone, for a total of 2^{n-l-1}.

(B) If an accepting quotient is included in the subset, then so is L. We can add the subset $\{\varepsilon\}$ or any non-empty subset S of accepting quotients that does not contain ε, since $S \cup \{\varepsilon\}$ is equivalent to S. Thus we have 2^{l-1} subsets of accepting quotients. To this we can add 2^{n-l-2} rejecting subsets, since the empty quotient and L need not be counted. The total is $2^{l-1}2^{n-l-2} = 2^{n-3}$.

Adding 1 for $(L^*)_\varepsilon$, we have a total of $2^{n-3} + 2^{n-l-1} + 1$.

(II) Suppose L is accepting. Since $n \geq 3$, we have $L \neq \varepsilon$.

(A) If there is no accepting quotient, there are 2^{n-l-1} subsets, as before.

(B) If an accepting quotient is included, then L is included and L itself is sufficient to guarantee that $(L^*)_w$ is accepting. Since $L \cup \varepsilon = L \cup \emptyset = L$, we also exclude ε and $\emptyset$. Thus any one of the 2^{n-3} subsets of the remaining quotients can be added to L.

The total is $2^{n-3} + 2^{n-l-1}$. We need not add $(L^*)_\varepsilon$, since it is LL^* which has been counted already.

The worst-case bound of $2^{n-3} + 2^{n-l-1} + 1$ occurs in the first case only. □

6 Prefix-free languages

As another example of the application of the quotient methods, we now derive the bounds for basic operations on prefix-free languages [8].

Proposition 9. *If K and L are both prefix-free and non-empty with $\kappa(K) = m$ and $\kappa(L) = n$, then*

(I) $\kappa(K \cup L) \leq mn - 2$,

(II) $\kappa(K \oplus L) \leq mn - 2$,

(III) $\kappa(K \cap L) \leq mn - (2m + 2n - 6)$,

(IV) $\kappa(KL) \leq m + n - 2$,

(V) $\kappa(L^*) \leq n$.

Proof: If L is non-empty, then it has at least one accepting quotient L_w. If L is also prefix-free, then $\varepsilon \in L_w$ implies $L_w = \varepsilon$. Thus both K and L have ε as a quotient. Parts (I)–(III) follow directly from Theorem 6.

For (IV), consider Equation (13). If $K = \varepsilon$, then $m = 2$, $KL = L$ and $\kappa(KL) = n$. If $n = 1$, then L can only be Σ^*, which is not prefix-free. If $n \geq 2$, then $n \leq m + n - 2$. Assume now that $\varepsilon \notin K$; hence the term $K^\varepsilon L_w$ is missing in (13). Consider any $w \neq \varepsilon$. If $\varepsilon \notin K_w$, $K_w \neq \emptyset$, and $\varepsilon \in K_u$ for some proper prefix u of $w = uv$, then K cannot be prefix-free, because it contains u and wx, for some $x \in \Sigma^+$. Thus, if K_w is rejecting and non-empty, then $(KL)_w = K_w L$; there are $m-2$ such quotients, the remaining two quotients being ε and $\emptyset$. If $K_w = \varepsilon$, then $(KL)_w = L$. Then $(KL)_{wx} = L_x$, for all $x \in \Sigma^*$, and there are n such quotients. If $K_w = \emptyset$, then $(KL)_w = \emptyset$, and this quotient has already been counted in the case where $K_w = \varepsilon$ and $L_x = \emptyset$. Thus the total is at most $m+n-2$.

For (V), we have $(L^*)_\varepsilon = L^*$, and for $w \in \Sigma^+$, consider Equation (14). By the argument we used in (IV), if L_w is rejecting and non-empty, then $(L^*)_w = L_w L^*$, and there are $n-2$ such quotients. If L_w is accepting, then $L_w = \varepsilon$, and $(L^*)_w = L_w L^* = L^* = (L^*)_\varepsilon$. Finally, if $L_w = \emptyset$, then $(L^*)_w$ may be empty, if for every prefix u of w, L_u is rejecting. Thus the bound is n. $\square$

It is shown in [8] that the bounds are tight: Let $K = (a^{m-2})^* b$ and $L = (a^{n-2})^* b$. Then $\kappa(KL) = m+n-2$, and $\kappa(L^*) = n$. If $\Sigma = \{a, b, c\}$ and

$$K = \{wc \mid w \in \{a,b\}^* \text{ and } |w|_a \equiv 0 \bmod m\},$$

$$L = \{wc \mid w \in \{a,b\}^* \text{ and } |w|_b \equiv 0 \bmod n\},$$

then $\kappa(K \cap L) = mn - (2m + 2n - 6)$. Finally, if $\Sigma = \{a, b, c, d\}$ and

$$K = \{wc \mid w \in \{a,b,d\}^* \text{ and } |w|_a \equiv 0 \bmod m\},$$

$$L = \{wd \mid w \in \{a,b,c\}^* \text{ and } |w|_b \equiv 0 \bmod n\},$$

then $\kappa(K \cup L) = mn - 2$.

7 Conclusions

Quotients provide a uniform approach for finding upper bounds for the complexity of operations on regular languages, and for verifying that particular languages meet these bounds. It is hoped that this is a step towards a theory of complexity of languages and automata.

Acknowledgments I am very grateful to Galina Jirásková for correcting several errors in early versions of this paper, suggesting better examples, improving proofs, and helping me with references, in particular, with the early work on complexity. I thank Sheng Yu for his help with references, and for answering many of my questions on complexity. I also thank Baiyu Li, Shengying Pan, and Jeff Shallit for their careful reading of the manuscript.

References

[1] J. C. Birget. Intersection of regular languages and state complexity. *ACM SIGACT News,* 22(2), 49, 1991.

[2] H. Bordihn, M. Holzer, M. Kutrib. State complexity of NFA to DFA conversion for subregular language families. *Proc. DCFS 2008*, C. Câmpeanu, G. Pighizzini, (eds.), University of Prince Edward Island, 85–96, 2006.

[3] J. Brzozowski. *Regular expression techniques for sequential circuits.* PhD Dissertation, Department of Electrical Engineering, Princeton University, 242 pp., 1962.

[4] J. Brzozowski. Derivatives of regular expressions. *J. ACM,* 11, 481–494, 1964.

[5] C. Câmpeanu, K. Salomaa, K. Culik II, S. Yu. State complexity of basic operations on finite languages. *Proc. WIA 1999*, *LNCS*, 2214, 60–70, 2001.

[6] H. Gruber, M. Holzer. Finite automata, digraph connectivity, and regular expression size. *Proc. ICALP 2008, Part II*, L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, I. Walukiewicz, (eds.), *LNCS,* 5126, 39–50, 2008.

[7] Y.-S. Han, K. Salomaa. State complexity of basic operations on suffix-free regular languages. *Theoret. Comput. Sci.,* 410, 2537–2548, 2009.

[8] Y.-S. Han, K. Salomaa, D. Wood. State complexity of prefix-free regular languages. *Proc. DCFS 2006,* 165–176, 2006.

[9] M. Holzer, M. Kutrib. Descriptional and computational complexity of finite automata. *Proc. LATA 2009,* A. H. Dediu, A. M. Ionescu, C. Martin-Vide, (eds.), *LNCS,* 5457, 23–42, 2009.

[10] M. Hricko. *Finite automata, regular languages, and state complexity.* Master's thesis. P. J. Šafárik University in Košice, Slovakia, 2005.

[11] M. Hricko, G. Jirásková, A. Szabari. Union and intersection of regular languages and descriptional complexity. *Proc. DCFS 2005,* C. Mereghetti, B. Palano, G. Pighizzini, D. Wotschke (eds.), University of Milano, Milano, Italy, 170–181, 2005.

[12] J. Jirásek, G. Jirásková, A. Szabari. State complexity of concatenation and complementation. *Internat. J. Found. Comput. Sci.,* 16, 511–529, 2005.

[13] G. Jirásková. On the state complexity of complements, stars, and reversals of regular languages. *Proc. DLT 2008,* M. Ito, M. Toyama, (eds.), *LNCS* 5257, 431–442, 2008.

[14] G. Jirásková. Magic numbers and ternary alphabet. *Proc. DLT 2009,* V. Diekert, D. Nowotka, (eds.,) *LNCS,* 5583, 300–311, 2009.

[15] G. Jirásková. Concatenation of regular languages and descriptional complexity. *Proc. CSR 2009,* to appear, 2009.

[16] G. Jirásková, A. Okhotin. State complexity of cyclic shift. *RAIRO Theor. Inform. Appl.,* 42, 335–360, 2008.

[17] G. Jirásková, P. Olejár. State complexity of union and intersection of binary suffix-free languages. Manuscript, June 2009.

[18] G. Jirásková, G. Pighizzini. Converting self-verifying automata into deterministic automata. *Proc. LATA 2009,* A. H. Dediu, A. M. Ionescu, C. Martin-Vide, (eds.), *LNCS,* 5457, 458–468, 2009.

[19] E. Leiss. Succinct representation of regular languages by boolean automata. *Theoret. Comput. Sci.,* 13, 323–330, 1981.

[20] O. B. Lupanov. A comparison of two types of finite automata. *Problemy Kibernetiki,* 9, 321–326, 1963 (in Russian). German translation: Über den Vergleich zweier Typen endlicher Quellen. *Probleme der Kybernetik* 6, 328–335, 1966.

[21] Yu. I. Lyubich. Estimates for optimal determinization of nondeterministic autonomous automata. *Sibirskii Matematicheskii Zhurnal,* 5, 337–355, 1964 (in Russian).

[22] A. N. Maslov. Estimates of the number of states of finite automata. *Dokl. Akad. Nauk SSSR,* 194, 1266–1268 (in Russian) 1970. English translation: *Soviet Math. Dokl.* 11 1373–1375, 1970.

[23] F. R. Moore. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Trans. Comput.,* C20 (10), 1211–1214, 1971.

[24] A. Nerode. Linear automaton transformations. *Proc. Amer. Math. Soc.,* 9, 541–544, 1958.

[25] D. Perrin. Finite automata. *Handbook of Theoretical Computer Science,* J. van Leewen, (ed.), Elsevier, B, 1–57,1990.

[26] G. Pighizzini, J. Shallit. Unary language operations, state complexity and Jacobsthal's function. *Internat. J. Found. Comput. Sci.,* 13(1), 145–159, 2002.

[27] M. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. and Dev.* 3, 114–129, 1959.

[28] A. Salomaa, K. Salomaa, S. Yu. State complexity of combined operations. *Theoret. Comput. Science*, 383, 140–152, 2007.

[29] K. Salomaa, S. Yu. On the state complexity of combined operations and their estimation. *Internat. J. Found. Comput. Sci.,* 18(4), 683–698, 2007.

[30] S. Yu. Regular languages. *Handbook of Formal Languages*, G. Rozenberg, A. Salomaa, (eds.), Springer, 1, 41–110, 1997.

[31] S. Yu. State complexity of regular languages. *J. Autom., Lang. and Comb.,* 6(2), 221–234, 2001.

[32] S. Yu, Q. Zhuang. On the state complexity of intersection of regular languages. *ACM SIGACT News,* 22(3), 52–54, 1991.

[33] S. Yu, Q. Zhuang, K. Salomaa. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.* 125, 315–328, 1994.