UNRESTRICTED STATE COMPLEXITY OF BINARY OPERATIONS ON REGULAR AND IDEAL LANGUAGES

JANusz A. BRZOZOWSKI(A,B) CORWIN Sinnamon(A,B)

(A) David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
brzozo@uwaterloo.ca (J. A. BRZOZOWSKI) sinncore@gmail.com (C. Sinnamon)

ABSTRACT

We study the state complexity of binary operations on regular languages over different alphabets. It is known that if $L'_m$ and $L_n$ are languages of state complexities $m$ and $n$, respectively, and restricted to the same alphabet, the state complexity of any binary boolean operation on $L'_m$ and $L_n$ is $mn$, and that of product (concatenation) is $m2^n - 2^{n-1}$. In contrast to this, we show that if $L'_m$ and $L_n$ are over different alphabets, the state complexity of union and symmetric difference is $(m+1)(n+1)$, that of difference is $mn + m$, that of intersection is $mn$, and that of product is $m2^n + 2^{n-1}$. We also study unrestricted complexity of binary operations in the classes of regular right, left, and two-sided ideals, and derive tight upper bounds. The bounds for product of the unrestricted cases (with the bounds for the restricted cases in parentheses) are as follows: right ideals $m + 2^{n-2} + 2^{n-1} + 1 (m + 2^{n-2})$; left ideals $mn + m + n (m + n - 1)$; two-sided ideals $m + 2n (m + n - 1)$. The state complexities of boolean operations on all three types of ideals are the same as those of arbitrary regular languages, whereas that is not the case if the alphabets of the arguments are the same. Finally, we update the known results about most complex regular, right-ideal, left-ideal, and two-sided-ideal languages to include the unrestricted cases.

Keywords: boolean operation, concatenation, different alphabets, left ideal, most complex language, product, quotient complexity, regular language, right ideal, state complexity, stream, two-sided ideal, unrestricted complexity

1. Motivation

Formal definitions are postponed until Section 2.

The first comprehensive paper on state complexity was published in 1970 by A. N. Maslov [21, p. 1373], but this work was unknown in the West for many years. Maslov wrote:

An important measure of the complexity of [sets of words representable in finite automata] is the number of states of a minimal representing automaton. ... if $T(A)$ and $T(B)$ are representable in automata $A$ and $B$ with $m$ and $n$ states respectively ..., then:

(B) This work was supported by the Natural Sciences and Engineering Research Council of Canada grant No. OGP0000871.
(i) $T(A) \cup T(B)$ is representable in an automaton with $mn$ states;

(ii) $T(A) \cdot T(B)$ is representable in an automaton with $(m-1) \cdot 2^n + 2^{n-1}$ states .

In this formulation these statements are false: we will show that union may require $(m+1)(n+1)$ states and product (concatenation), $m2^n + 2^{n-1}$ states. However, Maslov must have had in mind languages over the same alphabet, in which case the statements are correct.

The second comprehensive paper on state complexity was published by S. Yu, Q. Zhuang and K. Salomaa \cite{Yu} in 1994. Here the authors wrote:

(i) \ldots for any pair of complete $m$-state DFA $A$ and $n$-state DFA $B$ defined on the same alphabet $\Sigma$, there exists a DFA with at most $m2^n - 2^{n-1}$ states which accepts $L(A)L(B)$.

(ii) \ldots $m \cdot n$ states are \ldots sufficient \ldots for a DFA to accept the intersection (union) of an $m$-state DFA language and an $n$-state DFA language.

The first statement includes the same-alphabet restriction, but the second omits it (presumably it is implied by the context). Here DFA stands for deterministic finite automaton, and complete means that there is a transition from every state under every input letter.

After these two papers appeared many authors studied the state complexity of various operations in various classes of regular languages, always using witnesses restricted to the same alphabet. However, we point out that the same-alphabet restriction is unnecessary: there is no reason why we should not compute the union or product of two languages over different alphabets. In fact, the software package Grail, for instance, (http://www.csit.upei.ca/theory/) allows the user to calculate the result of these operations.

As an example, let us consider the union of languages $L'_2 = \{a, b\}^*b$ and $L_2 = \{a, c\}^*c$ accepted by the minimal complete two-state automata $D'_2$ and $D_2$ of Figure 1, where an incoming arrow denotes the initial state and a double circle represents a final state.

![Figure 1: Two minimal complete DFAs $D'_2$ and $D_2$.](image)

The union of $L'_2$ and $L_2$ is a language over three letters. To find the DFA for $L'_2 \cup L_2$, we view $D'_2$ and $D_2$ as incomplete DFAs, the first missing all transitions under $c$, and the second, under $b$. After adding the missing transitions we obtain DFAs $D'_3$ and $D_3$ shown in Figure 2. Now we can proceed as is usually done in the same-alphabet approach, and use the direct product of $D'_3$ and $D_3$ to find $L'_2 \cup L_2$. 
Here it turns out that six states are necessary to represent $L_2' \cup L_2$, but the state complexity of union is actually $(m + 1)(n + 1)$.

In general, when calculating the result of a binary operation on regular languages with different alphabets, we deal with special incomplete DFAs that are only missing some letters and all the transitions caused by these letters. The complexity of incomplete DFAs has been studied previously by Gao, K. Salomaa, and Yu [16] and by Maia, Moreira, and Reis [20]. However, the objects studied there are arbitrary incomplete DFAs, whereas we are interested only in complete DFAs with some missing letters. Secondly, we study state complexity, whereas the above-mentioned papers deal mainly with transition complexity. Nevertheless, there is some overlap. It was shown in [16, Corollary 3.2] that the incomplete state complexity of union is less than or equal to $mn + m + n$, and that this bound is tight in some special cases. In [20, Theorem 2], witnesses that work in all cases were found. These complexities correspond to our result for union in Theorem 4. Also in [20, Theorem 5], the incomplete state complexity of product is shown to be $m2^n + 2^{n-1} + 1$, and this corresponds to our result for product in Theorem 2.

In this paper we remove the restriction of equal alphabets of the two operands. We prove that the complexity of union and symmetric difference is $(m + 1)(n + 1)$, that of difference is $mn + m$, and that of intersection is $mn$, and that of the product is $m2^n + 2^{n-1}$, if each language’s own alphabet is used. We exhibit a new most complex regular language that meets the complexity bounds for restricted and unrestricted boolean operations, restricted and unrestricted products, star, and reversal, has a maximal syntactic semigroup, and most complex atoms. All the witnesses used here are derived from that one most complex language.

A much shorter version of this paper appeared in [5]. That paper dealt only with unrestricted product and binary boolean operations on regular languages. Here we include a shorter proof of the theorem about unrestricted product of regular languages, and establish the unrestricted complexities of product and binary boolean operations on right, left, and two-sided ideals.

2. Terminology and Notation

If $\Sigma$ is a finite alphabet and $L \subseteq \Sigma^*$, the alphabet of $L$ is the set $\Sigma_L = \{a \in \Sigma \mid uav \in L \text{ for some } u, v \in \Sigma^*\}$. A basic complexity measure of $L$ with alphabet $\Sigma_L$ is...
the number \( n \) of distinct (left) quotients of \( L \) by words in \( \Sigma_L^* \), where a (left) quotient of \( L \) by a word \( w \in \Sigma_L^* \) is \( w^{-1}L = \{ x \mid wx \in L \} \). The number of quotients of \( L \) is its quotient complexity \([9], \kappa(L)\).

Unless otherwise specified, for a regular language \( L \) with alphabet \( \Sigma_L \) we define the complement of \( L \) by \( \overline{L} = \Sigma_L^* \setminus L \). With this definition it is not always true that \( \kappa(L) = \kappa(\overline{L}) \) as \( L \) and \( \overline{L} \) may have different alphabets. For example, if \( L = \{ a, b \}^* \setminus a^* \), then \( \kappa(L) = 2 \) while \( \kappa(\overline{L}) = \kappa(a^*) = 1 \), for the alphabet of \( L \) is \( \{ a \} \) instead of \( \{ a, b \} \).

There is only one way for this to occur: In order for their alphabets to be different, there must be a letter in the alphabet of \( L \) such that every word containing the letter is in the language, so that the letter is not present in \( \overline{L} \). Hence \( \kappa(\overline{L}) \in \{ \kappa(L), \kappa(L) - 1 \} \), and it is usually easy to determine the complexity of \( \overline{L} \) when presented with a specific language \( L \).

Let \( L_n \) be a regular language with quotient complexity \( n \), let \( \circ \) be a unary operation on languages, and let \( L_n^\circ \) be the result of the operation. The quotient complexity of the operation \( \circ \) is the maximal value of \( \kappa(L_n^\circ) \) as a function of \( n \), as \( L_n \) ranges over all regular languages with quotient complexity \( \leq n \).

Let \( L_m \) and \( L_n \) be regular languages of quotient complexities \( m \) and \( n \) that have alphabets \( \Sigma_m \) and \( \Sigma \), respectively, let \( \circ \) be a binary operation on languages, and let \( L_m^\circ C. Sinnamon \)

A deterministic finite automaton (DFA) is a quintuple \( \mathcal{D} = (Q, \Sigma, \delta, q_0, F) \), where \( Q \) is a finite non-empty set of states, \( \Sigma \) 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. We extend \( \delta \) to a function \( \overline{\delta}: Q \times \Sigma^* \to Q \) as usual. A DFA \( \mathcal{D} \) accepts a word \( w \in \Sigma^* \) if \( \delta(q_0, w) \in F \). The language accepted by \( \mathcal{D} \) is denoted by \( L(\mathcal{D}) \). If \( q \) is a state of \( \mathcal{D} \), then the language \( L_q(\mathcal{D}) \) of \( q \) is the language accepted by the DFA \( (Q, \Sigma, \delta, q, \emptyset) \). A state is empty (or dead or a sink state) if its language is empty. Two states \( p \) and \( q \) of \( \mathcal{D} \) are equivalent if \( L_p(\mathcal{D}) = L_q(\mathcal{D}) \). A state \( q \) is reachable if there exists \( w \in \Sigma^* \) such that \( \delta(q_0, w) = q \). A DFA is minimal if all of its states are reachable and no two states are equivalent. Usually DFAs are used to establish upper bounds on the complexity of operations, and also as witnesses that meet these bounds.

The state complexity \([25]\) of a regular language \( L \) is the number of states in a complete minimal DFA with alphabet \( \Sigma_L \) which recognizes the language. This concept is equivalent to quotient complexity of \( L \). For example, the state complexity of the language \( a^* \) is one. There is a two-state minimal DFA with alphabet \( \{ a, b \} \) accepting \( a^* \), but its alphabet is not \( \Sigma_L \).

The quotient/state complexity of an operation gives a worst-case lower bound on the time and space complexities of the operation. For this reason it has been studied extensively; see \([3, 4, 24, 25]\) for additional references. Since we do not use any other measures of complexity in this paper (with the exception of one mention of time and space complexity above), we refer to quotient/state complexity simply as complexity.

If \( \delta(q, a) = p \) for a state \( q \in Q \) and a letter \( a \in \Sigma \), we say there is a transition under \( a \) from \( q \) to \( p \) in \( \mathcal{D} \). The DFAs defined above are complete in the sense that there is exactly one transition for each state \( q \in Q \) and each letter \( a \in \Sigma \). If there is at most
one transition for each \( q \in Q \) and \( a \in \Sigma \), the automaton is an incomplete DFA.

A nondeterministic finite automaton (NFA) is a 5-tuple \( D = (Q, \Sigma, \delta, I, F) \), where \( Q, \Sigma \) and \( F \) are defined as in a DFA, \( \delta : Q \times \Sigma \to 2^Q \) is the transition function, and \( I \subseteq Q \) is the set of initial states. An \( \varepsilon \)-NFA is an NFA in which transitions under the empty word \( \varepsilon \) are also permitted.

To simplify the notation, without loss of generality we use \( Q_n = \{0, \ldots, n-1\} \) as our basic set of \( n \) elements. A transformation of \( Q_n \) is a mapping \( t : Q_n \to Q_n \). The image of \( q \in Q_n \) under \( t \) is denoted by \( qt \). For \( k \geq 2 \), a transformation (permutation) \( t \) of a set \( P = \{q_0, q_1, \ldots, q_{k-1}\} \subseteq Q_n \) is a \( k \)-cycle if \( q_{0t} = q_1, q_{1t} = q_2, \ldots, q_{(k-1)t} = q_0 \). This \( k \)-cycle is denoted by \( (q_0, q_1, \ldots, q_{k-1}) \), and acts as the identity on the states in \( Q_n \setminus P \). A 2-cycle \((q_0, q_1)\) is called a transposition. A transformation that changes only one state \( p \) to a state \( q \neq p \) and acts as the identity for the other states is denoted by \((p \rightarrow q)\). The identity transformation is denoted by \( 1 \). If \( s, t \) are transformations of \( Q_n \), their composition when applied to \( q \in Q_n \) is defined by \((qs)t\).

The set of all \( n^k \) transformations of \( Q_n \) is a monoid under composition.

We use \( Q_n \) as the set of states of every DFA with \( n \) states, and 0 as the initial state. In any DFA \( D_n = (Q_n, \Sigma, \delta, 0, F) \) each \( a \in \Sigma \) induces a transformation \( \delta_a \) of \( Q_n \) defined by \( q\delta_a = \delta(q, a) \); we denote this by \( a : \delta_a \). For example, when defining the transition function of a DFA, we write \( a : (0, 1) \) to mean that \( \delta(q, a) = q(0,1) \), where the transformation \((0,1)\) acts on state \( q \) as follows: if \( q \) is 0 it maps it to 1, if \( q \) is 1 it maps it to 0, and it acts as the identity on the remaining states.

By a slight abuse of notation we use the letter \( a \) to denote the transformation it induces; thus we write \( qa \) instead of \( q\delta_a \). We extend the notation to sets of states: if \( P \subseteq Q_n \), then \( Pa = \{pa \mid p \in P\} \). We also find it convenient to write \( P \xrightarrow{a} Pa \) to indicate that the image of \( P \) under \( a \) is \( Pa \).

We extend these notions to arbitrary words. For each word \( w \in \Sigma^* \), the transition function induces a transformation \( \delta_w \) of \( Q_n \) by \( w \): for all \( q \in Q_n \), \( q\delta_w = \delta(q, w) \). The set \( T_{D_n} \) of all such transformations by non-empty words is the transition semigroup of \( D_n \) under composition \([23]\). If we also allow the empty word, then we have the transition monoid of \( D_n \), where the empty word induces the identity transformation.

The Myhill congruence \( \approx_{L_n} \) \([22]\) (also known as the syntactic congruence) of a language \( L_n \subseteq \Sigma^* \) is defined on \( \Sigma^* \) as follows: for \( x, y \in \Sigma^* \), \( x \approx_{L_n} y \) if and only if \( wxz \in L_n \iff wyz \in L_n \) for all \( w, z \in \Sigma^* \). The quotient set \( \Sigma^*/\approx_{L_n} \) of equivalence classes of \( \approx_{L_n} \) is a semigroup, the syntactic semigroup \( T_{L_n} \) of \( L_n \).

A related concept is that of the syntactic monoid, which is the quotient set \( \Sigma^*/\approx_{L_n}' \), where \( \approx_{L_n}' \) is \( \approx_{L_n} \) extended to \( \Sigma^* \). It is well known that a language is regular if and only if its syntactic semigroup (monoid) is finite \([22]\) \([23]\). Moreover, many classes of languages have been characterized by the properties of their syntactic semigroups or monoids \([15]\) \([23]\).

If \( D_n \) is a minimal DFA of \( L_n \), then \( T_{D_n} \) is isomorphic to the syntactic semigroup \( T_{L_n} \) of \( L_n \) \([23]\), and we represent elements of \( T_{L_n} \) by transformations in \( T_{D_n} \). The size of this semigroup has been used as a measure of complexity \([4]\) \([13]\) \([17]\) \([19]\). We use transition semigroups rather than monoids because in the latter the empty word always induces the identity transformation. However, in state complexity applications a non-empty word that induces the identity transformation is often needed.
The atom congruence is a left congruence defined as follows: two words \( x \) and \( y \) are equivalent if \( ux \in L \) if and only if \( uy \in L \) for all \( u \in \Sigma^* \). Thus \( x \) and \( y \) are equivalent if \( x \in u^{-1}L \) if and only if \( y \in u^{-1}L \). An equivalence class of this relation is called an atom of \( L \) \cite{12,18}. It follows that an atom is a non-empty intersection of complemented and uncomplemented quotients of \( L \). The number of atoms and their quotient complexities are possible measures of complexity of regular languages \cite{4}.

For more information about atoms and their complexity, see \cite{11,12,18}.

A sequence \( (L_n, n \geq k) = (L_k, L_{k+1}, \ldots) \) of regular languages is called a stream; here \( k \) is some small integer, and the languages in the stream usually have the same form and differ only in the parameter \( n \). For example, \( ((a,b)^*a^n(a,b)^* \mid n \geq 2) \) is a stream. To find the complexity of a binary operation \( \circ \) we need to find an upper bound on this complexity and two streams \( (L'_m, m \geq h) \) and \( (L_n, n \geq k) \) of languages meeting this bound. In general, the two streams are different, but there are many examples where \( L'_m \) “differ only slightly” from \( L_n \); such a language \( L'_n \) is called a dialect \cite{11} of \( L_n \), and is defined below.

Let \( \Sigma = \{a_1, \ldots, a_k\} \) be an alphabet; we assume that its elements are ordered as shown. Let \( \pi \) be a partial permutation of \( \Sigma \), that is, a partial function \( \pi: \Sigma \rightarrow \Gamma \) where \( \Gamma \subseteq \Sigma \), for which there exists \( \Delta \subseteq \Sigma \) such that \( \pi \) is bijective when restricted to \( \Delta \) and undefined on \( \Sigma \setminus \Delta \). We denote undefined values of \( \pi \) by “\( - \)”, that is, we write \( \pi(a) = \_ \), if \( \pi \) is undefined at \( a \).

If \( L \subseteq \Sigma^* \), we denote it by \( L(a_1, \ldots, a_k) \) to stress its dependence on \( \Sigma \). If \( \pi \) is a partial permutation, let \( s_\pi(L(a_1, \ldots, a_k)) \) be the language obtained from \( L(a_1, \ldots, a_k) \) by the substitution \( s_\pi \) defined as follows: for \( a \in \Sigma \), \( a \mapsto \{\pi(a)\} \) if \( \pi(a) \) is defined, and \( a \mapsto \emptyset \) otherwise. The permutational dialect, or simply dialect, of \( L(a_1, \ldots, a_k) \) defined by \( \pi \) is the language \( L(\pi(a_1), \ldots, \pi(a_k)) = s_\pi(L(a_1, \ldots, a_k)) \).

Similarly, let \( D = (Q, \Sigma, \delta, 0, F) \) be a DFA; we denote it by \( D(a_1, \ldots, a_k) \) to stress its dependence on \( \Sigma \). If \( \pi \) is a partial permutation, then the permutational dialect, or simply dialect, \( D(\pi(a_1), \ldots, \pi(a_k)) \) of \( D(a_1, \ldots, a_k) \) is obtained by changing the alphabet of \( D \) from \( \Sigma \) to \( \pi(\Sigma) \), and modifying \( \delta \) so that in the modified DFA \( \pi(a_i) \) induces the transformation induced by \( a_i \) in the original DFA. One verifies that if the language \( L(a_1, \ldots, a_k) \) is accepted by DFA \( D(a_1, \ldots, a_k) \), then \( L(\pi(a_1), \ldots, \pi(a_k)) \) is accepted by \( D(\pi(a_1), \ldots, \pi(a_k)) \).

If the letters for which \( \pi \) is undefined are at the end of the alphabet \( \Sigma \), then they are omitted. For example, if \( \Sigma = \{a, b, c, d\} \) and \( \pi(a) = b, \pi(b) = a, \) and \( \pi(c) = \pi(d) = \_ \), then we write \( L_{n_0}(b, a) \) for \( L_n(b, a, \_ \_ \_ \_, \_ \_ \_ \_) \), etc.

A most complex stream of regular languages is one that, together with some dialect streams, meets the complexity bounds for all boolean operations, product, star, and reversal, and has the largest syntactic semigroup and most complex atoms. In looking for a most complex stream we try to use the smallest possible alphabet sufficient to meet all the bounds. Most complex streams are useful in systems dealing with regular languages and finite automata. One would like to know the maximal sizes of automata that can be handled by the system. In view of the existence of most complex streams, one stream can be used to test all the operations.

\footnote{The bounds on the complexities of operations may not hold for values smaller than \( k \).}
3. Regular Languages

The DFA of Definition 1 will be used for both product and boolean operations on regular languages; this DFA is the 4-input DFA called \( \mathcal{U}_n(a, b, c, d) \) in [4], where it was shown that \( \mathcal{U}_n(a, b, c) \) is a “universal witness”, that is, \( (\mathcal{U}_n(a, b, c) \mid n \geq 3) \) is a most complex regular stream for all common restricted operations. We now prove that \( \mathcal{U}_n(a, b, c, d) \) (renamed \( D_n(a, b, c, d) \) below), together with some of its permutational dialects, is most complex for both restricted and unrestricted operations.

**Definition 1.** For \( n \geq 3 \), let \( D_n = D_n(a, b, c, d) = (Q_n, \Sigma, \delta_n, 0, \{n - 1\}) \), where \( \Sigma = \{a, b, c, d\} \), and \( \delta_n \) is defined by the transformations:

- \( a: (0, \ldots, n - 1) \),
- \( b: (0, 1) \),
- \( c: (n - 1 \rightarrow 0) \), and
- \( d: \emptyset \).

Let \( L_n = L_n(a, b, c, d) \) be the language accepted by \( D_n \).

The structure of \( D_n(a, b, c, d) \) is shown in Figure 3. Recall that we use the set \( Q_n = \{0, \ldots, n - 1\} \) as the state set of every DFA with \( n \) states. When dealing with two DFAs with \( m \) and \( n \) states, respectively, to avoid confusion of the states, we use \( D_m' = (Q_m', \Sigma, \delta', 0', F') \) and \( D_n = (Q_n, \Sigma, \delta, 0, F) \) for the two DFAs, and \( L_m' \) and \( L_n \) for their respective languages. Note that this also allows us to use two different alphabets \( \Sigma' \) and \( \Sigma \).

3.1. Product of Regular Languages

**Theorem 2 (Product of Regular Languages).** For \( m, n \geq 3 \), let \( L_m' \) (respectively, \( L_n \)) be a regular language with \( m \) (respectively, \( n \)) quotients over an alphabet \( \Sigma' \) (respectively, \( \Sigma \)). Then \( \kappa(L_m', L_n) \leq m2^n + 2^{n-1} \), and this bound is met by \( L_m'(a, b, -, c) \) and \( L_n(b, a, -, d) \) of Definition 1.

**Proof.** First we derive the upper bound. Let \( D_m' = (Q_m', \Sigma', \delta', 0', F') \) and \( D_n = (Q_n, \Sigma, \delta, 0, F) \) be DFAs of arbitrary regular languages \( L_m' \) and \( L_n \), respectively. We use the normal construction of an \( \varepsilon \)-NFA \( N \) to recognize \( L_m' \) by introducing an \( \varepsilon \)-transition from each final state of \( D_m' \) to the initial state of \( D_n \), and changing all final states of \( D_m' \) to non-final states. This is illustrated in Figure 4, where, in this example, \( (m - 1)' \) is the only final state of \( D_m' \). We then determinize \( N \) using the subset construction to get the DFA \( D \) for \( L_m' L_n \).
Suppose $D'_m$ has $k$ final states, where $1 \leq k \leq m - 1$. We will show that $D$ can have only the following types of states: (a) at most $(m - k)2^n$ states $\{p'\} \cup S$, where $p' \in Q'_m \setminus F'$, and $S \subseteq Q_n$, (b) at most $k2^{n-1}$ states $\{p',0\} \cup S$, where $p' \in F'$ and $S \subseteq Q_n \setminus \{0\}$, and (c) at most $2^n$ states $S \subseteq Q_n$.

Because $D'_m$ is deterministic, there can be at most one state $p'$ of $Q'_m$ in any reachable subset. If $p' \notin F'$, it may be possible to reach any subset of states of $Q_n$ along with $p'$, and this accounts for (a). If $p' \in F'$, then any set of reachable states of $D$ must contain 0 and possibly any subset of $Q_n \setminus \{0\}$, giving (b). It may also be possible to reach any subset $S$ of $Q_n$ by applying an input that is not in $\Sigma'$ to $\{0',1\}$, and we have (c). Altogether, there are at most $(m - k)2^n + k2^{n-1} + 2^n = (2m - k)2^{n-1} + 2^n$ reachable subsets. This expression reaches its maximum when $k = 1$, and so there are at most $m2^n + 2^n$ states in $D$.

![Figure 4: An NFA for the product of $L'_m(a,b,-,c)$ and $L_n(b,a,-,d)$.](image)

We prove that the bound is met by the witnesses of Figure 4. We use the following result to show that all the states in the subset construction are reachable.

Suppose $D'_m = (Q'_m, \Sigma, \delta', 0', \{f'\})$ is a minimal DFA of $L'_m$, $f' \neq 0'$, and $D_n = (Q_n, \Sigma, \delta, 0, F)$ is a minimal DFA of $L_n$. Moreover, assume that the transition semigroups of $D'_m$ and $D_n$ are groups.

**Lemma 3 (Sylvie Davies, personal communication).** If all the sets of the form $\{p'\}$, $p' \in Q'_m \setminus \{f'\}$, and $\{0', q\}$, $q \in Q_n$, are reachable, then so are all sets of the form

$$\{p'\} \cup S, p' \in Q'_m \setminus \{f'\}, S \subseteq Q_n \text{ and } \{f',0\} \cup S, S \subseteq Q_n \setminus \{0\}. \quad (*)$$

We now prove that the conditions of the lemma apply to our case. The initial state in the subset automaton is $\{0'\}$, state $\{p'\}$ is reached by $a^p$ if $p < m - 1$, and $\{(m-1)',0\}$ is reached by $a^m$. Also, $\{0',1\}$ is reached by $a^m$.

- If $n$ is odd, from $\{0',1\}$ we reach $\{0',q\}$, $q \in Q_n$, by words in $(bb)^*$.
If \( n \) is even, from \( \{0', 1\} \) we reach \( \{0', q\} \) with \( q \) odd by words in \((bb)^*\).

- From \( \{0', n-1\} \) we reach \( \{0', 0\} \) by \( ab \).
- From \( \{0', 0\} \) we reach \( \{0', q\} \) with \( q \) even by \((bb)^*\).

Since \( \{p', q\} \), \( p' \in Q'_m \setminus \{f'\} \) and \( \{0', q\} \), \( q \in Q_n \) are reachable, so are all the sets of form \((\cdot \cdot\cdot)^*\) by the Lemma.

For distinguishability, note that only state \( q \) accepts \( w_q = b^{n-1-q} \) in \( D_n \). Hence, if two states of the product have different sets \( S \) and \( S' \) and \( q \in S \oplus S' \), then they can be distinguished by \( w_q \). State \( \{p'\} \cup S \) is distinguished from \( S \) by \( ca^{n-1-m}b^{n-1} \).

If \( p < q \), states \( \{p'\} \cup S \) and \( \{q'\} \cup S \) are distinguished as follows: Use \( ca^{n-1-q} \) to reach \( \{(p + m - 1 - q)'\} \) from \( p' \) and \( \{(m - 1)'\} \) from \( q' \). The reached states are distinguishable since they differ in their subsets of \( Q_n \).

### 3.2. Boolean Operations on Regular Languages

Suppose \( A, B \subseteq U \), where \( U \) is some universal set. A binary operation \( \circ : \mathcal{P}(U) \times \mathcal{P}(U) \to \mathcal{P}(U) \) is boolean if, for any \( x \in U \), whether \( x \) is included in \( A \circ B \) depends only on the membership of \( x \) in \( A \) and \( B \). Thus there are sixteen binary boolean operations, corresponding to the number of ways of filling out the truth table below.

<table>
<thead>
<tr>
<th>( x \in A )</th>
<th>( x \in B )</th>
<th>( x \in A \circ B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>-</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>-</td>
</tr>
</tbody>
</table>

A boolean operation is proper if it is not constant and does not depend on only one variable. There are ten proper boolean operations, given below.

\[
\begin{align*}
L'_m \cup L_n &\quad \overline{\overline{L_m}} \cap \overline{L_n} \\
\overline{L_m} \cup \overline{L_n} &\quad L'_m \cap \overline{L_n} = L'_m \setminus L_n \\
\overline{L_m} \cup \overline{L_n} &\quad L'_m \cap L_n = L_n \setminus L'_m \\
\overline{L_m} \cup \overline{L_n} &\quad L'_m \cap L_n \\
\end{align*}
\]

Although the complement of a regular language \( L_n \) is usually taken with respect to \( \Sigma^* \), where \( \Sigma \) is the alphabet of \( L_n \), the list above requires that \( \overline{L_m} \) denotes the complement of \( L_n \) in a specific universal set \( U \) which contains both \( L'_m \) and \( L_n \). We wish for \( U \) to be the set of all words over some alphabet, and it is most natural to have \( U = (\Sigma' \cup \Sigma)^* \), where \( \Sigma' \) is the alphabet of \( L'_m \). Thus, contrary to its usual meaning, every use of complement in the list of operations above is taken with respect to \((\Sigma' \cup \Sigma)^*\).

We study the complexities of four proper boolean operations only: union \( (L'_m \cup L_n) \), symmetric difference \( (L'_m \oplus L_n) \), difference \( (L'_m \setminus L_n) \), and intersection \( (L'_m \cap L_n) \). From these four it is generally a straightforward exercise to deduce the complexity of any other operation: The complexity of \( L_n \setminus L'_m \) is determined by symmetry with
and from De Morgan’s laws we have \( L_m' \cup L_n = L_m' \cap L_n, \) 
\( L_m' \cap L_n = L_m' \cup L_n, \) 
\( L_m' \backslash L_n = L_m' \cup L_n, \) 
\( L_m \backslash L_n = L_m' \cup L_n, \) 
\( L_m' \cap L_n = L_m' \cup L_n, \) \( L_m' \backslash L_n = L_m' \cup L_n, \) and \( L_m \oplus L_n = L_m' \oplus L_n. \) As discussed in Section 2, \( \kappa(L) \) and \( \kappa(L') \) differ by at most 1 for any regular language \( L, \) and for any specific witness one can easily determine the discrepancy; for this reason we leave it as an exercise to verify that our witnesses meet the upper bounds for all ten proper operations using the bounds for the four operations that we address explicitly.

It turns out that the witnesses that we used for unrestricted product also work for unrestricted boolean operations.

**Theorem 4 (Boolean Operations on Regular Languages).** For \( m, n \geq 3, \) let \( L_m' \) (respectively, \( L_n \)) be a regular language with \( m \) (respectively, \( n \)) quotients over an alphabet \( \Sigma, \) (respectively, \( \Sigma' \)). Then the complexity of union and symmetric difference is \((m + 1)(n + 1)\) and this bound is met by \( L_m'(a, b, - , c) \) and \( L_n(b, a, - , d); \) the complexity of difference is \( mn + m, \) and this bound is met by \( L_m'(a, b, - , c) \) and \( L_n(b, a); \) the complexity of intersection is \( mn \) and this bound is met by \( L_m'(a, b) \) and \( L_n(b, a). \)

**Proof.** Let \( D_m' = (Q_m', \Sigma', \delta', 0', F') \) and \( D_n = (Q_n, \Sigma, \delta, 0, F) \) be minimal DFAs for arbitrary regular languages \( L_m' \) and \( L_n \) with \( m \) and \( n \) quotients, respectively. To calculate an upper bound for the boolean operations assume that \( \Sigma' \setminus \Sigma \) and \( \Sigma \setminus \Sigma' \) are non-empty; this assumption results in the largest upper bound. We add an empty state \( 0' \) to \( D_m' \) to send all transitions under the letters from \( \Sigma \setminus \Sigma' \) to that state; thus we get an \((m + 1)\)-state DFA \( D_{m,0'} \). Similarly, we add an empty state \( 0 \) to \( D_n \) to get \( D_{n,0}. \) Now we have two DFAs over the same alphabet, and an ordinary problem of finding an upper bound for the boolean operations on two languages over the same alphabet, *except that these languages both have empty quotients*. It is clear that \((m + 1)(n + 1)\) is an upper bound for all four operations, but it can be improved for difference and intersection. Consider the direct product \( P_{m,n} \) of \( D_{m,0'} \) and \( D_{n,0}. \)

For difference, all \( n + 1 \) states of \( P_{m,n} \) that have the form \((0', q), \) where \( q \in Q_n \cup \{0\} \) are empty. Hence the bound can be reduced by \( n \) states to \( mn + m + 1. \) However, the empty states can only be reached by words in \( \Sigma' \setminus \Sigma \) and the alphabet of \( L_m' \setminus L_n \) is a subset of \( \Sigma' \); hence the bound is reduced further to \( mn + m. \)

For intersection, all \( n \) states \((0', q), \) \( q \in Q_n, \) and all \( m \) states \((p', 0), \) \( p' \in Q_m \), are equivalent to the empty state \((0', 0), \) thus reducing the upper bound to \( mn + 1. \) Since the alphabet of \( L_m' \cap L_n \) is a subset of \( \Sigma' \cap \Sigma, \) these empty states cannot be reached and the bound is reduced to \( mn. \)

To prove that the bounds are tight, we start with \( D_n(a, b, c, d) \) of Definition 4. For \( m, n \geq 3, \) let \( D'_m(a, b, c, d) \) be the dialect of \( D_m'(a, b, c, d) \) where \( c \) plays the role of \( d \) and the alphabet is restricted to \( \{a, b, c\}, \) and let \( D_n(b, a, - , d) \) be the dialect of \( D_n(a, b, c, d) \) in which \( a \) and \( b \) are permuted, and the alphabet is restricted to \( \{a, b, d\}; \) see Figure 5. We complete the two DFAs by adding empty states, and then construct the direct product of the new DFAs as illustrated in Figure 4 for \( m = 3 \) and \( n = 4. \)

If we restrict both DFAs to the alphabet \( \{a, b\}, \) we have the usual problem of determining the complexity of two DFAs over the same alphabet. By 2, Theorem 1, all \( mn \)
states of the form \((p', q), p' \in Q'_m, q \in Q_n\), are reachable and pairwise distinguishable by words in \(\{a, b\}^*\) for all proper boolean operations if \((m, n) \notin \{(3, 4), (4, 3), (4, 4)\}\). For our application, the three exceptional cases were verified by computation.

To prove that the remaining states are reachable, observe that \((\emptyset', 0) \xrightarrow{d} (\emptyset', 0)\) and \((\emptyset', 0) \xrightarrow{b} (\emptyset', q), q \in Q_n\). Symmetrically, \((\emptyset', 0) \xrightarrow{c} (\emptyset', 0)\) and \((\emptyset', 0) \xrightarrow{a} (p', 0), p' \in Q'_m\). Finally, \((\emptyset', n-1) \xrightarrow{c} (\emptyset', 0)\), and all \((m+1)(n+1)\) states of the direct product are reachable.
Remark 5. In the restricted case the complexity of every one of the ten binary
boolean functions in $mn$. In the unrestricted case one verifies that we have

$$\kappa(L'_m \cup L_n) = \kappa(L'_m \cap L_n) = \kappa(L'_m \oplus L_n) = \kappa(L'_m \oplus \overline{L_n}) = (m + 1)(n + 1),$$

$$\kappa(L'_m \cup \overline{L_n}) = mn + m + 1, \ \kappa(L'_m \cup \overline{L_n}) = mn + n + 1,$$

$$\kappa(L'_m \cap \overline{L_n}) = mn + m, \ \kappa(L'_m \cap L_n) = mn + n,$$

$$\kappa(L'_m \cap \overline{L_n}) = mn + 1, \ \kappa(L'_m \cap L_n) = mn.$$
3.3. Most Complex Regular Languages

We now update the result of [4] to include the unrestricted case.

**Theorem 6 (Most Complex Regular Languages).** For each \( n \geq 3 \), the DFA of Definition [4] is minimal and its language \( L_n(a, b, c, d) \) has complexity \( n \). The stream \( (L_m(a, b, c, d) \mid m \geq 3) \) with dialect streams \( (L_n(a, b, -, c) \mid n \geq 3) \) and \( (L_n(b, a, - , d) \mid n \geq 3) \) is most complex in the class of regular languages. In particular, it meets all the complexity bounds below, which are maximal for regular languages. In several cases the bounds can be met with a reduced alphabet.

1. The syntactic semigroup of \( L_n(a, b, c) \) has cardinality \( n^n \).
2. Each quotient of \( L_n(a) \) has complexity \( n \).
3. The reverse of \( L_n(a, b, c) \) has complexity \( 2^n \), and \( L_n(a, b, c) \) has \( 2^n \) atoms.
4. For each atom \( A_S \) of \( L_n(a, b, c) \), the complexity \( \kappa(A_S) \) satisfies:
   \[
   \kappa(A_S) = \begin{cases} 
   2^n - 1, & \text{if } S \in \{\emptyset, Q_n\}; \\
   1 + \sum_{s=1}^{|S|} \sum_{y=1}^{n-|S|} \binom{n}{y} (n-y), & \text{if } \emptyset \subseteq S \subseteq Q_n.
   \end{cases}
   \]
5. The star of \( L_n(a, b) \) has complexity \( 2^{n-1} + 2^{n-2} \).
6. **Product**
   - **Restricted case:** The product \( L'_m(a, b, c) L_n(a, b, c) \) has complexity \( m2^n - 2^{n-1} \).
   - **Unrestricted case:** The product \( L'_m(a, b, -, c) L_n(b, a, -, d) \) has complexity \( m2^n + 2^{n-1} \).
7. **Boolean operations**
   - **Restricted case:** The complexity of \( L'_m(a, b) \circ L_n(b, a) \) is \( mn \).
   - **Unrestricted case:** The complexity of \( L'_m(a, b, -, c) \circ L_n(b, a, -, d) \) is \( (m+1)(n+1) \) if \( \circ \in \{\cup, \oplus\} \), that of \( L'_m(a, b, -, c) \setminus L_n(b, a) \) is \( mn + m \), and that of \( L'_m(a, b) \cap L_n(b, a) \) is \( mn \).

*Proof.* The proofs for the restricted case can be found in [4], and the claims for the unrestricted case were proved in the present paper, in Theorems 2 and 4. \( \Box \)

**Proposition 7 (Marek Szykuła, personal communication).** At least four letters are required for a most complex regular language. In particular, four letters are needed for union: two letters to reach all pairs of states in \( Q'_m \times Q_n \), one in \( \Sigma' \setminus \Sigma \) for pairs \( (p', q) \) with \( p' \in Q'_m \), and one in \( \Sigma \setminus \Sigma' \) for pairs \( (\emptyset, q) \) with \( q \in Q_n \).

****

A non-empty language \( L \) is a right ideal (left ideal, two-sided ideal, respectively) if \( L = L \Sigma^* \) (\( L = \Sigma^* L \), \( L = \Sigma^* \Sigma^* \), respectively). Ideals are fundamental objects in semigroup theory. We study regular ideals only.
Ideals appear in the area of pattern matching \cite{14}. For this application, a text is represented by a word $w$ over some alphabet $\Sigma$. A pattern can be an arbitrary language $L$ over $\Sigma$ described by a regular expression. An occurrence of a pattern represented by $L$ in text $w$ is a triple $(u, x, v)$ such that $w = uxv$ and $x$ is in $L$. Searching text $w$ for words in $L$ is equivalent to looking for prefixes of $w$ that belong to the language $\Sigma^* L$, which is the left ideal generated by $L$.

Algorithms such as that of Aho and Corasick \cite{1} can be used to determine all possible occurrences of words from a finite set $L$ in a given input $w$. For example, in a Unix-style editor such as sed, one can find all the words ending in $x$ (that is, all the words of the left ideal $\Sigma^* x$) that occur in $w$; all the words beginning with $x$ (that is, all the words of the right ideal $x \Sigma^*$) that occur in $w$; and all the words that have $x$ as a factor (that is, all the words of the two-sided ideal $\Sigma^* x \Sigma^*$) that occur in $w$.

The complexities of restricted basic operations on ideals were studied in \cite{9}. The sizes of transition semigroups of minimal DFAs accepting ideal languages were determined in \cite{10, 13}. Atoms of ideals were analyzed in \cite{7}. Most complex right ideals for restricted operations were studied in \cite{6}, and left and two-sided ideals were added in \cite{8}. In this paper we add the results for unrestricted binary operations.

4. Right Ideals

A stream of right ideals that have the largest syntactic semigroups was introduced in \cite{13}. A different stream was used in \cite{6} and shown to be most complex for restricted operations; it was also studied in \cite{7, 8}. Here we modify the stream of \cite{6} by adding an input $e$ that induces the identity transformation. We find the unrestricted state complexity of product and boolean operations of this stream together with some of its permutational dialects.

**Definition 8.** For $n \geq 3$, let $\mathcal{D}_n = \mathcal{D}_n(a, b, c, d, e) = (Q_n, \Sigma, \delta_n, 0, \{n - 1\})$, where $\Sigma = \{a, b, c, d, e\}$, and $\delta_n$ is defined by the transformations $a: (0, \ldots, n - 2)$, $b: (1, \ldots, n - 2)$, $c: (n - 2 \rightarrow 0)$, $d: (n - 2 \rightarrow n - 1)$, and $e: \bot$. Let $L_n = L_n(a, b, c, d, e)$ be the language accepted by $\mathcal{D}_n$. The structure of $\mathcal{D}_n$ is shown in Figure 7.

![Figure 7: Minimal DFA $\mathcal{D}_n(a, b, c, d, e)$ of Definition 8.](image-url)
4.1. Product of Right Ideals

**Theorem 9 Product of Right Ideals.** For \( m, n \geq 3 \), let \( L'_m \) (respectively, \( L_n \)) be an arbitrary right ideal with \( m \) (respectively, \( n \)) quotients over an alphabet \( \Sigma' \), (respectively, \( \Sigma \)). Then \( \kappa(L'_m L_n) \leq m + 2^n - 2 + 2^{n-1} + 1 \), and this bound is met by the right ideals \( L'_m(a, b, -, d, e) \) and \( L_n(a, b, -, d, c) \) of Definition 5.

**Proof.** Let \( D'_m = (Q'_m, \Sigma', \delta', 0', \{(m-1)'\}) \) and \( D_n = (Q_n, \Sigma, \delta, 0, \{n-1\}) \) be DFAs of arbitrary right ideals \( L'_m \) and \( L_n \), respectively. We can safely assume that they each have a unique final state because every right ideal has a single final quotient accepting all words over its alphabet. We use the standard construction of the NFA for the product \( L'_m L_n \). We bound the complexity of the product by counting the reachable and distinguishable sets in the subset construction on this NFA. The \( m - 1 \) sets \( \{p'\} \), where \( p' \in Q'_m \) is non-final, as well as \( \{(m-1)', 0\} \) may be reachable. From \( \{(m-1)', 0\} \) we may reach all \( 2^n - 1 \) sets \( \{(m-1)', 0\} \cup S \) for \( S \subseteq Q_n \setminus \{0\} \). The \( 2^n \) sets \( S \subseteq Q_n \) may also be reachable by using a letter in \( \Sigma \setminus \Sigma' \). So far, there are at most \( m - 1 + 2^n - 1 + 2^n \) reachable sets. However, in the DFA obtained by the subset construction from the NFA, states \( \{n - 1\} \cup S \) for \( S \subseteq Q_{n-1} \) are all equivalent, since they accept all words in \( \Sigma^* \) and go to \( \emptyset \) by letters in \( \Sigma \setminus \Sigma' \). Similarly, states \( \{(m-1)', 0, n - 1\} \cup S \) are all equivalent because they all accept \( \Sigma^* \) and go to \( \{(m-1)', 0\} \) by letters in \( \Sigma \setminus \Sigma' \). This leaves at most \( m + 2^{n-2} + 2^n + 1 \) distinguishable states.

![Diagram](image)

Figure 8: An NFA for product of right ideals \( L'_m(a, b, -, d, e) \) and \( L_n(a, b, -, d, c) \).

To show that this bound can be met, consider two dialects of the DFA of Definition 5 shown in Figure 8. Here \( \Sigma' = \{a, b, d, e\} \) and \( \Sigma = \{a, b, c, d\} \). Set \( \{p'\} \) for \( p' \in Q'_{m-1} \) is reached by \( a^n \), and \( \{(m-1)', 0\} \) is reached by \( a^{n-2}d \).
Any set \( \{(m-1)',0,q_1,q_2,\ldots,q_k\} \) with \( 0 < q_1 < q_2 < \cdots < q_k < n-1 \) and \( k \geq 1 \) is reached from \( \{(m-1)',0,q_2,q_3-q_1,\ldots,q_k-q_1-1\} \) by \( ab^{n-1} \); hence all sets \( \{(m-1)',0,q_2,q_3-q_1,\ldots,q_k-q_1-1\} \) are reachable. Set \( S = \{q_1,q_2,\ldots,q_k\} \) with \( 0 < q_1 < q_2 < \cdots < q_k < n-1 \) and \( k \geq 1 \) is reachable from \( \{(m-1)',0,q_2-q_1,\ldots,q_k-q_1\} \) by \( ca^{n} \), and \( \emptyset \) is reached from \( \{0\} \) by \( e \); hence all sets \( S \subseteq Q_{n-1} \) are reachable. Sets containing \( n-1 \) are easily reached from these sets using \( a \) and \( d \). However, in the DFA obtained from the NFA, the states \( S \subseteq Q_n \) that contain \( n-1 \) all accept \( \Sigma^* \) and are sent to the empty state by \( e \); hence they are all equivalent. Similarly, the states \( \{(m-1)',0\} \cup S \) that contain \( n-1 \) all accept \( \{a,b,c,d\}^* \) and are sent to \( \{(m-1)',0\} \) by \( e \); hence they are also equivalent.

The remaining states are pairwise distinguishable. States \( \{p'\} \) and \( \{q'\} \) with \( 0 \leq p < q \leq n-2 \) are distinguished by \( a^{n-2}qda^{n-2}d \), and \( \{p'\} \) is distinguished from \( \{(m-1)',0\} \cup S \) or from \( S \), where \( \emptyset \subseteq S \subseteq Q_n \), by \( (ad)^{n-1} \). Two states \( \{(m-1)',0\} \cup S \) and \( \{(m-1)',0\} \cup T \) with \( q \in S \oplus T \) are distinguished by \( a^{n-2}qda \), as are two states \( S \) and \( T \) with \( q \in S \oplus T \). A set \( \{(m-1)',0\} \cup S \) is distinguishable from \( T \), where \( S,T \subseteq Q_n \), by \( ca^{n-2}d \). Every state is distinguishable from \( \emptyset \) by a word in \( \{a,d\}^* \). Thus all \( m + 2n^2 + 2^{n-1} + 1 \) states are pairwise distinguishable.

\[ \square \]

4.2. Boolean Operations on Right Ideals

**Theorem 10 (Boolean Operations on Right Ideals).** For \( m,n \geq 3 \), the unrestricted complexities of boolean operations on right ideals are the same as those for arbitrary regular languages. In particular, the right ideals \( L_m(a,b,-,d,e) \) and \( L_n(e,c,-,d,a) \) of Definition 8 meet the bound \( (m+1)(n+1) \) for union and symmetric difference, \( L'_m(a,b,-,d,e) \) and \( L_n(e,-,-,d,a) \) meet the bound \( mn + m \) for difference, and \( L'_m(a,-,-,d,e) \) and \( L_n(e,-,-,d,a) \) meet the bound \( mn \) for intersection.

**Proof.** We show that the bounds of Theorem 4 for arbitrary regular languages are met by DFAs \( D_m(a,b,-,d,e) \) and \( D_n(e,c,-,d,a) \) of the right ideals of Definition 8.

To compute \( L'_m(a,b,-,d,e) \circ L_n(e,c,-,d,a) \), where \( \circ \) is a boolean operation, add an empty state \( \emptyset' \) to \( D_m(a,b,-,d,e) \), and send all the transitions from any state of \( Q'_m \) under \( c \) to \( \emptyset' \). Similarly, add an empty state \( \emptyset \) to \( D_n(e,c,-,d,a) \) together with appropriate transitions; now the alphabets of the resulting DFAs are the same. The direct product of \( L'_m(a,b,-,d,e) \) and \( L_n(e,c,-,d,a) \) is illustrated in Figure 9 for \( m = n = 4 \).

We first check that all \( (m+1)(n+1) \) states of the direct product are reachable. State \( (p',q) \in Q'_m \times Q_n \) is reachable by \( a^p e^q \). State \( ((m-1)',q) \) for \( q \in Q_{n-1} \) is reached from \( ((m-2)',0) \) by \( de^q \), state \( (p',n-1) \) for \( p' \in Q_{m-1} \) is similarly reached from \( (0',n-2) \), and \( ((m-1)',n-1) \) is reached from \( ((m-2)',n-2) \) by \( d \). Hence the states of \( Q'_m \times Q_n \) are reachable; the remaining states involve \( \emptyset' \) or \( \emptyset \) and are reached from states of \( Q'_m \times Q_n \) using \( b \) and \( c \) as seen in Figure 9.

We now check distinguishability, which depends on the final states of the DFA.
Union  The final states of the direct product for union are \(((m - 1)', q)\) for \(q \in Q_n \cup \{\emptyset\}\) and \((p', n - 1)\) for \(p' \in Q'_m \cup \{\emptyset'\}\). States that differ in the first coordinate are distinguished by words in \(ba^*d\), and states that differ in the second coordinate are distinguished by words in \(ce^*d\); hence all \((m + 1)(n + 1)\) states are distinguishable.

Symmetric Difference  The final states for symmetric difference are the same as those for union, except that \(((m - 1)', n - 1)\) is non-final. All states are pairwise distinguishable as in union.

Difference  The final states for difference are \(((m - 1)', q)\) for \(q \neq n - 1\). The alphabet of \(L'_m(a, b, -, d, e) \setminus L_n(e, c, -, d, a)\) is \(\{a, b, d, e\}\); hence we can omit \(c\) and delete all states \((\emptyset', q)\) and be left with a DFA recognizing the same language. The remaining states are distinguished by words in \(ba^*d\) if they differ in the first coordinate, or by words in \(a^*da^*d\) if they differ in the second coordinate. The bound for difference is also met by \(L'_m(a, b, -, d, e) \setminus L_n(e, -, -, d, a)\).

Intersection  For intersection, the only final state is \(((m - 1)', n - 1)\). The alphabet of \(L'_m(a, b, -, d, e) \cap L_n(e, c, -, d, a)\) is \(\{a, d, e\}\); hence we can omit \(b\) and \(c\) and delete all states \((p', \emptyset)\) and \((\emptyset', q)\). The remaining \(mn\) states are pairwise distinguishable by words in \(e^*da^*d\). The bound for intersection is also met by \(L'_m(a, -, -, d, e) \setminus L_n(e, -, -, d, a)\).

One verifies that the complexities of all ten binary boolean functions on right ideals are as given in Remark \[\text{remark}^5\].
4.3. Most Complex Right Ideals

Theorem 11 (Most Complex Right Ideals). For each \( n \geq 3 \), the DFA of Definition 8 is minimal and \( L_n(a, b, c, d, e) \) is a right ideal of complexity \( n \). The stream \( (L_n(a, b, c, d, e) \mid n \geq 3) \) with some dialect streams is most complex in the class of right ideals. In particular, it meets all the bounds below. In several cases the bounds can be met with a reduced alphabet.

(i) The syntactic semigroup of \( L_n(a, b, c, d) \) has cardinality \( n^{n-1} \).

(ii) The quotients of \( L_n(a, - , - , d) \) have complexity \( n \) except for the final quotient which has complexity 1.

(iii) The reverse of \( L_n(a, - , - , d) \) has complexity \( 2^n - 1 \), and \( L_n(a, - , - , d) \) has \( 2^n - 1 \) atoms.

(iv) Each atom \( A_S \) of \( L_n(a, b, c, d) \) has maximal complexity:

\[
\kappa(A_S) = \begin{cases} 
2^{n-1}, & \text{if } S = Q_n; \\
1 + \sum_{x=1}^{\left\lfloor \frac{|S|}{2} \right\rfloor} \sum_{y=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} (n-1-x) \cdot (n-1-y), & \text{if } \emptyset \subsetneq S \subsetneq Q_n.
\end{cases}
\]

(v) The star of \( L_n(a, - , - , d) \) has complexity \( n + 1 \).

(vi) Product

(a) Restricted case:

\( L'_m(a, b, - , d) L_n(a, b, - , d) \) has complexity \( m + 2^n - 2 \).

(b) Unrestricted case:

\( L'_m(a, b, - , d, e) L_n(a, b, - , d, c) \) has complexity \( m + 2^n - 2 + 1 \).

(vii) Boolean operations

(a) Restricted case:

The complexity of \( \circ \) is \( mn \) if \( \circ \in \{ \cap, \oplus \} \), \( mn - (m-1) \) if \( \circ = \setminus \), and \( mn - (m + n - 2) \) if \( \circ = \cup \), and these bounds are met by \( L'_m(a, b, - , d) \circ L_n(b, a, - , d) \).

(b) Unrestricted case:

The complexity of \( L'_m(a, b, - , d, e) \circ L_n(e, c, - , d, a) \) is the same as for arbitrary regular languages: \( (m+1)(n+1) \) if \( \circ \in \{ \cup, \oplus \} \), \( mn + m \) if \( \circ = \setminus \), and \( mn \) if \( \circ = \cap \). The bound for difference is also met by \( L'_m(a, b, - , d, e) \setminus L_n(e, - , - , d, a) \) and the bound for intersection by \( L'_m(a, - , - , d, e) \cap L_n(e, - , - , d, a) \).

Proof. The restricted complexity results were proved in [5], and the unrestricted results for product and boolean operations were proved in Theorems 9 and 10. □

5. Left Ideals

The following stream of left ideals was defined in [13], where it was conjectured that its DFAs have maximal transition semigroups. This conjecture was proved in [10], and it was shown in [5] that this stream is most complex for restricted operations. We prove that it is also most complex in the unrestricted case.
Definition 12. For \( n \geq 4 \), let \( D_n = D_n(a, b, c, d, e) = (Q_n, \Sigma, \delta_n, 0, \{ n - 1 \}) \), where \( \Sigma = \{ a, b, c, d, e \} \), and \( \delta_n \) is defined by transformations \( a: (1, \ldots, n - 1) \), \( b: (1, 2) \), \( c: (n - 1 \rightarrow 1) \), \( d: (n - 1 \rightarrow 0) \), \( e: (Q_n \rightarrow 1) \). Let \( L_n = L_n(a, b, c, d, e) \) be the language accepted by \( D_n \). The structure of \( D_n(a, b, c, d, e) \) is shown in Figure 10.

Figure 10: Minimal DFA of left ideal of Definition 12

5.1. Product of Left Ideals

Theorem 13 (Product of Left Ideals). For \( m, n \geq 4 \), let \( L'_m \) (respectively, \( L_n \)) be an arbitrary left ideal with \( m \) (respectively, \( n \)) quotients over an alphabet \( \Sigma' \), (respectively, \( \Sigma \)). Then \( \kappa(L'_m, L_n) \leq mn + m + n \), and this bound is met by \( L'_m(a, b, \ldots, d, e) \) and \( L_n(a, d, c, \ldots, e) \) of Definition 12.

Proof. We first derive an upper bound on the complexity of \( L'_m, L_n \), where \( L'_m \) is any regular language over \( \Sigma' \) and \( L_n \) is any left ideal over \( \Sigma \). If \( \Sigma' \cup \Sigma' \cup \Sigma' \cup \Sigma \) are non-empty, we consider \( L'_m \) and \( L_n \) to be over \( \Sigma' \cup \Sigma' \cup \Sigma' \cup \Sigma \) by adding empty quotients.

We count the number of distinct quotients of \( L'_m, L_n \). Consider \( w^{-1}(L'_m, L_n) \) for a word \( w \in (\Sigma' \cup \Sigma)^* \). We decompose \( w \) as \( u_1u_2 \cdots u_kv \) where \( u_i \cdots u_i \in L'_m \) for \( i = 1, \ldots, k \) and no other prefix of \( w \) is in \( L'_m \). Then \( wx \in L'_m, L_n \) if and only if \( u_i+1 \cdots u_kv \in L_n \) for some \( 1 \leq i \leq k \), or \( x = yz \) such that \( wy \in L'_m \) and \( z \in L_n \); hence

\[
w^{-1}(L'_m, L_n) = \bigcup_{i=1}^{k} (u_{i+1} \cdots u_kv)^{-1}L_n \cup (w^{-1}L'_m)L_n.
\]

Notice that \( v^{-1}L_n \subseteq (u_kv)^{-1}L_n \subseteq (u_{k-1}u_kv)^{-1}L_n \subseteq \cdots \subseteq (u_{i+1} \cdots u_kv)^{-1}L_n \) as long as \( u_{i+1} \cdots u_kv \in \Sigma^* \), since \( L_n \) is a left ideal with respect to \( \Sigma \). Alternatively if \( u_{i+1} \cdots u_kv \) contains a letter from \( \Sigma' \cup \Sigma' \cup \Sigma' \cup \Sigma \) then \( (u_{i+1} \cdots u_kv)^{-1}L_n = \emptyset \). Thus, either \( \bigcup_{i=1}^{k} (u_{i+1} \cdots u_kv)^{-1}L_n = \emptyset \), or there exists a minimal \( j \in \{ 1, \ldots, k \} \) such that \( u_{j+1} \cdots u_kv \in \Sigma^* \) in which case \( \bigcup_{i=1}^{j} (u_{i+1} \cdots u_kv)^{-1}L_n = (u_{j+1} \cdots u_kv)^{-1}L_n \). Hence \( \bigcup_{i=1}^{k} (u_{i+1} \cdots u_kv)^{-1}L_n \) is one of the \( n + 1 \) quotients of \( L_n \). Each quotient of the product may therefore be written as \( K \cup K'L_n \) for quotients \( K' \) of \( L'_m \) and \( K \) of \( L_n \). Since \( L'_m \) and \( L_n \) have at most \( m + 1 \) and \( n + 1 \) quotients...
respectively, their product can have no more than \((m+1)(n+1)\) distinct quotients. Recall that \(L_n\) and \(\emptyset\) are both quotients of \(L_n\): in the case that \(K'\) is final, the sets \(L_n \cup K'L_n\) and \(\emptyset \cup K'L_n\) are equal. As \(L'_m\) has at least one final quotient, we have the upper bound of \(mn + m + n\). By comparison, if \(L'_m\) and \(L_n\) use the same alphabet then their product has complexity at most \(m + n - 1\). 

The dialects \((L'_m(a, b, -, d, e) \mid m \geq 4)\) and \((L_n(a, d, c, -, e) \mid n \geq 4)\) of the left ideal stream of Definition 12 meet the upper bound for product. To prove this we apply the usual NFA construction for product. This NFA is illustrated in Figure 11 for \(m = n = 4\).

![NFA for product of left ideals](image)

The subset construction yields sets \(\{p'\} \cup S\) where \(p' \in Q'_m\) and \(S \subseteq Q_n\). However it is impossible to reach any set containing two or more elements of \(Q_n \setminus \{0\}\) since they are only reachable from \(0\) by applying \(e\), and doing so sends all of \(Q_n\) to \(1\). Moreover, in the DFA obtained from the NFA by the subset construction, the states \(\{p', q\}\) are equivalent to \(\{p', 0, q\}\) for each \(p' \in Q'_m\) and \(q \in Q_n\), and states \(\{q\}\) and \(\{0, q\}\) are similarly equivalent. Hence we consider only the states \(\{p'\}\) for \(p' \in Q'_m - \{0\}\) for \(q \in Q_n\), \(\{p', 0, q\}\) for \(p' \in Q'_m\) and \(q \in Q_n\), and \(\emptyset\); a total of \(mn + m + n\) states.

We check reachability of these states. State \(\{0'\}\) is initial and \(\{p'\}\) is reached by \(e a^{p-1}\) for \(1 \leq p \leq m - 2\). State \(\{(m - 1)', 0\}\) is reached by \(e a^{m-2}\), and \(\{p', 0\}\) is then reached by \(a^p\) for \(1 \leq p \leq m - 1\), or by \(d\) for \(p' = 0'\). Applying \(e\) from \(\{(m - 1)', 0\}\) reaches \(\{1', 1\}\), and we reach \(\{1', 0, 1\}\) by \(e a^{(m-1)(n-1)}\). State \(\{p', 0, 2\}\) is then reached by \((ad)^{p-2}a\) for \(2 \leq p \leq m - 1\), and \(\{1', 0, 2\}\) is reached from \(\{1', 0, 1\}\) by \(d\). Now states of the form \(\{p', 0, q\}\) for \(p \in Q'_m \setminus \{0'\}\) and \(q \in Q_n \setminus \{0\}\) are reached by words in \(a^*\) from states of the form \(\{p', 0, 2\}\). State \(\{0', 0, q\}\) is reachable from \(\{(m - 1)', 0, 1\}\) by \(da^{q-1}\), and \(\{0, q\}\) is reached from \(\{0', 0, 1\}\) by \(e a^{q-1}\). Finally \(\emptyset\) is reached from \(\{0'\}\) by \(c\). Thus, all states are reachable.

All non-empty states are distinguished from \(\emptyset\) by \(e a^{m-2}e a^{n-2}\). States \(\{p'_1\}\) and \(\{p'_2\}\) where \(p_1 < p_2\) are distinguished by \(a^{m-1}p_1 e a^{n-2}\), and states \(\{0, q_1\}\) and \(\{0, q_2\}\) where \(q_1 < q_2\) are distinguished by \(a^{n-1}q_2\). Two states which differ on \(Q'_m\) are reduced to \(\{p'_1\}\) and \(\{p'_2\}\) by \(b\), and two states which differ on \(Q_n\) are reduced to \(\{0, q_1\}\) and \(\{0, q_2\}\) by \(c\) or \(ac\). Hence \(L'_m(a, b, -, d, e)\) and \(L_n(a, d, c, -, e)\) meet the
upper bound for product. □

5.2. Boolean Operations on Left Ideals

Theorem 14 (Boolean Operations on Left Ideals). For $m, n \geq 4$, the unrestricted complexities of boolean operations on left ideals are the same as those for arbitrary regular languages. In particular, the left ideals $L'_m(a, -, c, d, e)$ and $L_n(a, b, e, -, c)$ of Definition 12 meet the bound $(m+1)(n+1)$ for union and symmetric difference, $L'_m(a, -, c, d, e)$ and $L_n(a, -, e, -, c)$ meet the bound $mn + m$ for difference, and $L'_m(a, -, c, -, e)$ and $L_n(a, -, e, -, c)$ meet the bound $mn$ for intersection.

Proof. The upper bounds on the complexity of boolean operations for left ideals are the same as for regular languages. We show that the left ideals $L'_m(a, -, c, d, e)$ and $L_n(a, b, e, -, c)$ of Definition 12 meet these bounds.

The direct product for union is illustrated in Figure 12 for the case $m = n = 4$. Let $S = Q'_m \times Q_n$; when the languages have the same alphabet the direct product only contains the $mn$ states of $S$. It was proved in [8] that $L'_m(a, -, c, -, e) \circ L_n(a, -, e, -, c)$ has complexity $mn$ for each proper boolean operation $\circ \in \{\cup, \oplus, \setminus, \cap\}$; hence all the states of $S$ are reachable and pairwise distinguishable by words in $\{a, c, e\}^*$ for each operation. The remaining states $(p', \emptyset)$ and $(\emptyset', q)$ in the unrestricted direct product are reachable using $b$ and $d$ – which are not present in both alphabets.

Figure 12: Direct product for union of $L'_4(a, b, c, -, e)$ and $L_4(a, d, e, -, c)$ shown partially.
It remains to determine which states are distinguishable for each operation. Let

\[ H = \{ (\emptyset', q) \mid q \in Q_n \} \text{ and } V = \{ (p', \emptyset) \mid p' \in Q'_m \}. \]

**Union** The final states are \( \{ ((m-1)', q) \mid q \in Q_n \cup \{ \emptyset \} \} \) and \( \{ (p', n-1) \mid p' \in Q'_m \cup \{ \emptyset' \} \} \). Every state in \( V \) accepts a word with a \( b \), whereas no state in \( H \) accepts such words. Similarly, every state in \( H \) accepts a word with a \( d \), whereas no state in \( V \) accepts such words. Every state in \( Q'_m \times Q_n \) accepts a word with a \( b \) and a word with a \( d \). State \( (\emptyset', \emptyset) \) accepts no words at all. Hence any two states chosen from different sets (the sets being \( S, H, V \), and \( \{ (\emptyset', \emptyset) \} \)) are distinguishable. States in \( H \) are distinguishable by words in \( a^* \), as are those in \( V \). Therefore all \( (m+1)(n+1) \) states are pairwise distinguishable.

**Symmetric Difference** The final states here are all the final states for union except \( ((m-1)', n-1) \). The rest of the argument is the same as for union.

**Difference** The final states now are \( \{ ((m-1)', q) \mid q \notin n-1 \} \). The \( n \) states of the form \( (\emptyset', q) \), \( q \in Q_n \), are now equivalent to the empty state \( (\emptyset', \emptyset) \). The remaining states are pairwise distinguishable by the arguments used for union. Hence we have \( mn + m + 1 \) distinguishable states. However, since \( b \) is not in the alphabet of the difference it can be omitted along with the empty states \( (\emptyset', q) \), \( q \in Q_n \cup \{ \emptyset \} \) (reachable only by \( b \)), giving a bound of \( mn + n \). This bound can be reached by \( L'_m(a, -, c, d, e) \) and \( L_n(a, -, e, -, c) \).

**Intersection** Here only \( ((m-1)', n-1) \) is final and all states \( (p', \emptyset), p' \in Q'_m \), and \( (\emptyset', q), q \in Q_n \), are equivalent to \( (\emptyset', \emptyset) \), leaving \( mn + 1 \) distinguishable states. However, the empty state can be reached only by \( b \) or \( d \), which are not in the alphabet of \( L'_m(a, -, c, d, e) \cap L_n(a, b, e, -, c) \). Hence the correct bound is \( mn \) and it is reached by \( L'_m(a, -, c, -, e) \) and \( L_n(a, -, e, -, c) \).

The complexities of all ten binary boolean functions on left ideals are given in Remark 6.

5.3. Most Complex Left Ideals

We now update the results from [8] to include the unrestricted cases.

**Theorem 15** (Most Complex Left Ideals). For each \( n \geq 4 \), the DFA of Definition 12 is minimal and its language \( L_n(a, b, c, d, e) \) is a left ideal of complexity \( n \). The stream \( (L_m(a, b, c, d, e) \mid m \geq 4) \) with some dialect streams is most complex in the class of left ideals. In particular, it meets all the complexity bounds below, which are maximal for left ideals. In several cases the bounds can be met with a reduced alphabet.

1. The syntactic semigroup of \( L_n(a, b, c, d, e) \) has cardinality \( n^{n-1} + n - 1 \). Moreover, fewer than five inputs do not suffice to meet this bound.
2. Each quotient of \( L_n(a, -, -, d, e) \) has complexity \( n \).
3. The reverse of \( L_n(a, -, c, d, e) \) has complexity \( 2^{n-1} + 1 \), and \( L_n(a, -, c, d, e) \) has \( 2^{n-1} + 1 \) atoms.
For each atom $A_S$ of $L_n(a, b, c)$, the complexity $\kappa(A_S)$ satisfies:

$$\kappa(A_S) = \begin{cases} n, & \text{if } S = Q_n; \\ 2^{n-1}, & \text{if } S = \emptyset; \\ 1 + \sum_{x=1}^{|S|} \sum_{y=1}^{n-|S|} \binom{n-1}{x} \binom{n-1-x}{y}, & \text{otherwise}. \end{cases}$$

(v) The star of $L_n(a, -, -, -, e)$ has complexity $n + 1$.

(vi) Product

(A) Restricted case:

The product $L'_m(a, -, -, -, e)L_n(a, -, -, -, e)$ has complexity $m + n - 1$.

(B) Unrestricted case:

The product $L'_m(a, b, -, d, e)L_n(a, d, c, -, e)$ has complexity $mn + m + n$.

(vii) Boolean operations

(A) Restricted case:

For any proper binary boolean function $\circ$, the complexity of $L'_m(a, -, c, -, e)$ $\circ L_n(a, -, c, -, e)$ is $mn$.

(B) Unrestricted case:

The complexity of $L'_m(a, -, c, d, e) \circ L_n(a, b, e, -, c)$ is the same as for arbitrary regular languages: $(m+1)(n+1)$ if $\circ \in \{\cup, \oplus\}$, $mn+m$ if $\circ = \setminus$, and $mn$ if $\circ = \cap$. The bound for difference is also met by $L'_m(a, -, c, d, e) \setminus L_n(a, -, c, -, e)$ and the bound for intersection by $L'_m(a, -, c, -, e) \cap L_n(a, -, c, -, e)$.

Proof. The proofs for the restricted cases can be found in [8], and the claims about the unrestricted complexities are proved in Theorems 13 and 14. \qed

6. Two-Sided Ideals

The following stream of two-sided ideals was defined in [13], where it was conjectured that the DFAs in this stream have maximal transition semigroups. This conjecture was proved in [10], and the stream was shown to be most complex for restricted operations in [8]. We prove that it is also most complex in the unrestricted case.

Definition 16. For $n \geq 5$, let $D_n = D_n(a, b, c, d, e, f) = (Q_n, \Sigma, \delta, 0, \{n-1\})$, where $\Sigma = \{a, b, c, d, e, f\}$, and $\delta_n$ is defined by the transformations $a: (1, 2, \ldots, n-2)$, $b: (1, 2)$, $c: (n-2 \to 1)$, $d: (n-2 \to 0)$, $e: (Q_{n-1} \to 1)$, and $f: (1 \to n-1)$. The structure of $D_n(a, b, c, d, e, f)$ is shown in Figure 13.

6.1. Product of Two-Sided Ideals

Theorem 17 (Product of Two-Sided Ideals). For $m, n \geq 5$, let $L'_m$ (respectively, $L_n$) be any two-sided ideal with $m$ (respectively, $n$) quotients over an alphabet $\Sigma'$ (respectively, $\Sigma$). Then $\kappa(L'_m \cup L_n) \leq m + 2n$, and this bound is met by the two-sided ideals $L'_m(a, b, -, -, e, f)$ and $L_n(a, c, -, -, e, f)$ of Definition 16.
Proof. We first derive an upper bound on the complexity of $L'_m L_n$, where $L'_m$ is a right ideal over $\Sigma'$ and $L_n$ is a left ideal over $\Sigma$; as two-sided ideals are both right and left ideals, this upper bound will apply. In taking their product, we consider the languages over the union of their alphabets. If $L'_m$ and $L_n$ use different alphabets then they each have an empty quotient when taken over the combined alphabet; hence $L'_m$ and $L_n$ have up to $m + 1$ and $n + 1$ quotients respectively.

We count the number of quotients $w^{-1}(L'_m L_n)$ where $w \in (\Sigma' \cup \Sigma)^*$. Write $w = u_1 u_2 \cdots u_k v$ where $u_1 \cdots u_i \in L'_m$ for $i = 1, \ldots, k$ and no other prefix of $w$ is in $L'_m$. Then $wz \in L'_m L_n$ if and only if $u_{i+1} \cdots u_k vz \in L_n$ for some $1 \leq i \leq k$, or $x = yz$ is such that $wy \in L'_m$ and $z \in L_n$; hence

$$w^{-1}(L'_m L_n) = \bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L_n \cup (w^{-1} L'_m) L_n.$$ 

Consider the case where $w \in \Sigma^*$. If $k = 0$, then $w^{-1}(L'_m L_n)$ simplifies to $(w^{-1} L'_m) L_n$; there are at most $m + 1$ quotients of this form since $w^{-1} L'_m$ is one of the non-final quotients of $L'_m$. Assume $k \geq 1$; then $w^{-1} L'_m = \Sigma^*$ since $L'_m$ is a right ideal with respect to $\Sigma'$. Hence

$$w^{-1}(L'_m L_n) = \bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L_n \cup \Sigma^* L_n.$$ 

Moreover there are only $n + 1$ possible values for $\bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L_n$: Notice that $v^{-1} L_n \subseteq (u_k v)^{-1} L_n \subseteq (u_{k-1} u_k v)^{-1} L_n \subseteq \cdots \subseteq (u_{i+1} \cdots u_k v)^{-1} L_n$ as long as $u_{i+1} \cdots u_k v \in \Sigma^*$, since $L_n$ is a left ideal with respect to $\Sigma$. Alternatively if $u_{i+1} \cdots u_k v$ contains a letter from $\Sigma' \setminus \Sigma$ then $(u_{i+1} \cdots u_k v)^{-1} L_n = \emptyset$. Thus, either $\bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L = \emptyset$, or there exists a minimal $j \in \{1, \ldots, k\}$ such that $u_{j+1} \cdots u_k v \in \Sigma^*$ in which case $\bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L_n = (u_{j+1} \cdots u_k v)^{-1} L_n$. Hence $\bigcup_{i=1}^{k} (u_{i+1} \cdots u_k v)^{-1} L_n$ is one of the $n + 1$ quotients of $L_n$. 

---

Figure 13: Minimal DFA $D_n(a, b, c, d, e, f)$ of Definition 16.
Now $w^{-1}(L_m^0 L_n) = K \cup \Sigma^+ L_n$, which has at most $n + 1$ distinct values as $K$ is a quotient of $L_n$. However $L_n \cup \Sigma^+ L_n = \emptyset \cup \Sigma^+ L_n$, so there are only $n$ distinct values. So far, we have counted a total of $(m-1) + n$ quotients of the product.

Now suppose $w \notin \Sigma^+$; then $w^{-1}L_m^0 = \emptyset$. By the same reasoning we have $w^{-1}(L_m^0 L_n) = K \cup (w^{-1}L_m^0)L_n = K$ for some quotient $K$ of $L_n$. There are at most $n + 1$ quotients of $L_n$, yielding an upper bound of $(m-1) + n + (n+1) = m+2n$ quotients of $L_m^0 L_n$.

We prove this bound is tight using the dialect streams $(L_m^0(a, b, -, -, e, f) \mid m \geq 5)$ and $(L_n(a, c, -, -, e, f) \mid n \geq 5)$ of Definition 10. We apply the usual NFA construction for product. This NFA is illustrated in Figure 14 for $m = n = 5$.

The subset construction yields sets $\{p\} \cup S$ where $p \in Q_m$ and $S \subseteq Q_n$, as well as sets $S \subseteq Q_n$, although many sets of this form are not reachable here. In $D_m^0$, states $p \in Q_{m-1}$ are not reachable from $(m-1)p$ and so all sets $\{p\} \cup S$ where $S \neq \emptyset$ are unreachable. In $D_n$, states of $\{1, \ldots, n-2\}$ are only reachable from 0 using $e$, which maps $Q_n \setminus \{n-1\}$ to 1, and no other state is reachable from $n-1$; this restricts $S$ to contain only one state from $\{1, \ldots, n-2\}$. Thus, the only potentially reachable sets are $\{p\}$ for $p \in Q_{m-1}$, $(m-1)'0, q$ and $(m-1)'0, q, n-1$ for $q \in Q_{n-1}$, $\{0, q\}$ and $\{0, q, n-1\}$, $\{q\}$ and $\{q, n-1\}$ for $q \in Q_{n-1}$, and $\emptyset$.

We reach sets $\{p\}$ by $ea^{p-1}$ for $1 \leq p \leq m - 2$, and $(m-1)'0$ by $e$. From $(m-1)'0$ we reach $(m-1)'0, q$ by $ea^{q-1}$ for $1 \leq q \leq n-2$. Set $(m-1)'0, n-1$ is reached from $(m-1)'0, 1$ by $f$, and $(m-1)'0, q, n-1$ is then reached by $ea^{q-1}$. From $(m-1)'0$ we reach $(0) \cup S$ by $e^2$. Set $(q)$ is reached from $(0)$ by $ea^{q-1}$ for $q \in \{1, \ldots, n-2\}$, and $\{q, n-1\}$ is reached from $(0, n-1)$ by the same word. Finally, $\emptyset$ is reached from $(0')$ by $e$.

Figure 14: NFA for product of two-sided ideals.
In the DFA obtained from the NFA by the subset construction, many of these subsets represent equivalent states. All sets containing \( n - 1 \) and not \((m - 1)’\) accept \( \{a, c, e, f\}^* \) and are mapped to \( \emptyset \) by \( b \); hence they are equivalent. Similarly, all sets containing \((m - 1)’\) and \( n - 1 \) accept \( \{a, c, e, f\}^* \) and are mapped to \( \{(m - 1)’, 0\} \) by \( b \); hence they are equivalent. Moreover, sets \( \{0, q\} \) and \( \{q\} \) are equivalent for \( 1 \leq q \leq n - 2 \), since any word that maps \( b \) to \( 0 \) to \( n - 1 \) also maps \( q \) to \( n - 1 \). Therefore, we only need to consider sets \( \{p’\} \) for \( p’ \in Q_{m-1}’ \), \( \{(m - 1)’, 0, q\} \) for \( q \in Q_{n-1} \), and \( \{0, q\} \) for \( q \in Q_{n-1} \) as well as \( \{(m - 1)’, n - 1\}, \{n - 1\}, \emptyset \). Every reachable set is equivalent to one of these \( m + 2n \) sets. We prove that they are pairwise distinguishable.

All non-empty sets are distinguished from \( \emptyset \) by \( e f e f \). States \( \{p_1’\} \) and \( \{p_2’\} \) where \( p_1 < p_2 \) are distinguished by \( a^{m-1}p_2 f e f \). Sets \( \{(m - 1)’, 0, q_1\} \) and \( \{(m - 1)’, 0, q_2\} \) for \( q_1 < q_2 \) are distinguished by \( a^{m-1}p_2 f \); sets \( \{0, q_1\} \) and \( \{0, q_2\} \) are similarly distinguished. State \( \{p\} \) is distinguished from any state containing \( q \in Q_n \) by \( e f \). Finally, \( \{(m - 1)’, 0\} \cup S_1 \) is distinguished from \( S_2 \subseteq Q_n \) by \( b e f \). Thus all \( m + 2n \) states are distinguishable.

\[ \square \]

6.2 Boolean Operations on Two-Sided Ideals

**Theorem 18 (Boolean Operations on Two-Sided Ideals).** For \( m, n \geq 5 \), the unrestricted complexities of boolean operations on two-sided ideals are the same as those for arbitrary regular languages. In particular, the two-sided ideals \( L_m’(a, b, c, e, f) \) and \( L_n(a, e, d, -b, f) \) of Definition 16 meet the bound \((m+1)(n+1)\) for union and symmetric difference, \( L_m’(a, b, c, e, f) \) and \( L_n(a, e, d, -b, f) \) meet the bound \( mn + m \) for difference, and \( L_m’(a, b, -, e, f) \) and \( L_n(a, e, d, -b, f) \) meet the bound \( mn \) for intersection.

\[ \begin{align*}
\text{Proof.} & \text{ The upper bounds on the complexity of boolean operations for two-sided ideals are the same as for regular languages. We show that the two-sided ideals } \\
& \text{ } L_m’(a, b, c, -e, f) \text{ and } L_n(a, e, d, -b, f) \text{ of Definition 14 meet these bounds.} \\
& \text{The direct product for union is illustrated in Figure 13. We check that all } (m+1)(n+1) \text{ states are reachable. First consider the states of } Q_{m-1}’ \times Q_{n-1}. \text{ State } (0’, 0) \text{ is initial, } (0’, q) \text{ is reached by } ba^{q-1}, \text{ and } (p’, 0) \text{ is reached by } e^{p-1}. \text{ State } (1’, q) \text{ is reached from } (0’, q) \text{ by } e^2, \text{ and } (p’, 1) \text{ is reached from } (p’, 0) \text{ by } b^2. \text{ State } (p’, q) \text{ with } q \geq p \text{ is reached from } (1’, q) \text{ by } a^{p-1}. \text{ By symmetry all states } (p’, q) \text{ with } p \geq q \text{ are also reachable; hence the states of } Q_{m-1}’ \times Q_{n-1} \text{ are reachable. The remaining states of } \{(m - 1)’, 0\} \times Q_n \cup \{\emptyset\} \text{ and } Q_m’ \cup \{0\} \times \{n - 1, \emptyset\} \text{ are easily seen to be reachable in Figure 13.} \\
& \text{It remains to determine which states are distinguishable for each operation. Let } \\
& H = \{(0’, q) | q \in Q_n\} \text{ and } V = \{(p’, \emptyset) | p’ \in Q_m’\}. \\
& \text{Union The final states are } \{(m - 1)’, q\} \text{ and } \{p’ \in Q_{m}’ \cup \{0\}\}. \text{ Every state in } V \text{ accepts a word with a } c, \text{ whereas no state in } H \text{ accepts such words. Similarly, every state in } H \text{ accepts a word with a } d, \text{ whereas no state in } V \text{ accepts such words. Every state in } Q_m’ \times Q_n \text{ accepts a} \]
\[ \]
Figure 15: Direct product for union of $L'_5(a, b, c, e, f)$ and $L_5(a, e, d, -, b, f)$ shown partially.

word with a $b$ and a word with a $d$. State $(\emptyset', \emptyset)$ accepts no words at all. Hence
any two states chosen from different sets (the sets being $Q'_m \times Q_n$, $H$, $V$, and
$\{(\emptyset', \emptyset)\}$) are distinguishable. States in $H$ are distinguishable by words in $a^*f$,
and those in $V$ by words in $a^*f$. States that differ in the first coordinate are
sent to distinct states of $V$ by $c$; the exceptions are the pair of states $(1', q_1)$ and
$(2', q_2)$, which are sent to distinct states of $V$ by $ac$. Similarly, states that differ
in the second coordinate are sent to distinct states of $H$ by $d$ or $ad$. Therefore
all $mn + m + n + 1$ states are pairwise distinguishable.

**Symmetric Difference** The final states here are all the final states for union except
$(\emptyset - 1', n - 1)$. The rest of the argument is the same as for union.

**Difference** The final states now are $\{(m - 1)', q | q \neq n - 1\}$. The $n$ states of the
form $(\emptyset', q), q \in Q_n$ are now equivalent to the empty state $\{(\emptyset', \emptyset)\}$; all other
states are non-empty as they accept $cef$. States that differ in the first coordinate
are distinguished by words in $ca^*f$; the exceptions are the pairs of states $(1', q_1)$
and $(2', q_2)$ which are distinguished by words in $aca^*f$. States $(m - 1)', q)$ for
$q \in Q_n$ are distinguished by words in $a^*f$, and $((m - 1)', \emptyset)$ is distinguished
from $((m - 1)', q)$ by $bf$. States that differ in the second coordinate are mapped
by either $ef$ or $eaf$ to distinct states of $\{(m-1),q\} | q \in Q_n \cup \emptyset\}$, and are therefore distinguishable. Hence we have $mn + m + 1$ distinguishable states. However, $d$ is not in the alphabet of the difference, and so $d$ and the empty state can be removed, giving bound $mn + m$ reached by $L_m'(a,b,c,-,e,f)$ and $L_n(a,e,-,-,b,f)$.

**Intersection** Here only $((m-1)',n-1)$ is final and all states $(p',\emptyset), p' \in Q'_m$, and $(\emptyset',q), q \in Q_n$ are equivalent to $\{\emptyset',\emptyset\}$, leaving $mn + 1$ states. All other states are non-empty as they accept $efbf$. States $(p',n-1)$ for $p' \in Q'_m$, as well as states $((m-1),q)$ for $q \in Q_n$, are distinguished by words in $a'f$. States that differ in the first coordinate are mapped by either $bf$ or $baf$ to distinct states of $\{(m-1),q\} | q \in Q_n\}$, and are hence distinguishable. States that differ in the second coordinate are mapped by either $ef$ or $eaf$ to distinct states of $\{(p',n-1) | p' \in Q'_m\}$, and are hence distinguishable. However, $c$ and $d$ are not in the alphabet of the intersection, and so $c$ and the empty state can be removed, giving bound $mn$ reached by $L'_m(a,b,-,-,e,f)$ and $L_n(a,e,-,-,b,f)$.

The complexities of all ten binary boolean functions on two-sided ideals are given in Remark 5.

6.3. Most Complex Two-Sided Ideals

**Theorem 19 (Most Complex Two-Sided Ideals).** For each $n \geq 5$, the DFA of Definition 12 is minimal and its language $L_n(a,b,c,d,e,f)$ is a two-sided ideal of complexity $n$. The stream $(L_n(a,b,c,d,e,f) | n \geq 5)$ with some dialect streams is most complex in the class of regular two-sided ideals. In particular, this stream meets all the complexity bounds listed below, which are maximal for two-sided ideals. In several cases the bounds can be met with a reduced alphabet.

(i) The syntactic semigroup of $L_n(a,b,c,d,e,f)$ has size $n^{n-2} + (n-2)2^{n-2} + 1$. Moreover, fewer than six inputs do not suffice to meet this bound.

(ii) Each quotient of $L_n(a,-,-,d,e,f)$ has complexity $n$.

(iii) The reverse of the language $L_n(a,-,-,d,e,f)$ has complexity $2^{n-1} + 1$, and $L_n(a,-,-,d,e,f)$ has $2^{n-1} + 1$ atoms.

(iv) For each atom $A_S$ of $L_n(a,b,c,d,e,f)$, the complexity $\kappa(A_S)$ satisfies:

$$\kappa(A_S) = \begin{cases} n, & \text{if } S = Q_n; \\
2^{n-2} + n - 1, & \text{if } S = Q_n \setminus \{1\}; \\
1 + \sum_{x=1}^{|S|} \sum_{y=1}^{|S|} (n-2)(n-x-1), & \text{otherwise.} \end{cases}$$

(v) The star of $L_n(a,-,-,e,f)$ has complexity $n + 1$.

(vi) Product

(a) Restricted case:

The product $L_m'(a,-,-,e,f)L_n(a,-,-,e,f)$ has complexity $m + n - 1$. 
(ii) Unrestricted case:
The product \( L'_m(a, b, -, -, e, f) L_n(a, c, -, -, e, f) \) has complexity \( m + 2n \).

(vii) Boolean operations

(A) Restricted case:
For any proper binary boolean function \( \circ \), the complexity of \( L'_m(a, b, -d, e, f) \)
\( \circ L_n(b, a, -d, e, f) \) is maximal. In particular,

1. \( L'_m(a, b, -d, e, f) \cap L_n(b, a, -d, e, f) \) has complexity \( mn \), as does
   \( L'_m(a, b, -d, e, f) \oplus L_n(b, a, -d, e, f) \).

2. \( L'_m(a, b, -d, e, f) \setminus L_n(b, a, -d, e, f) \) has complexity \( mn - (m - 1) \).

3. \( L'_m(a, b, -d, e, f) \cup L_n(b, a, -d, e, f) \) has complexity \( mn - (m + n - 2) \).

(B) Unrestricted case:
The complexity of \( L'_m(a, b, c, -, e, f) \circ L_n(a, e, d, -, b, f) \) is the same as for any regular languages:
\( (m + 1)(n + 1) \) if \( \circ \in \{\cup, \oplus\} \), \( mn + m \) if \( \circ = \| \), and \( mn \) if \( \circ = \cap \). The bound for difference is also met
by \( L'_m(a, b, c, -, e, f) \setminus L_n(a, e, -b, f) \) and the bound for intersection by
\( L'_m(a, b, -, e, f) \cap L_n(a, e, -, b, f) \).

Proof. The proofs for the restricted cases can be found in \([8]\), and the claims about
the unrestricted complexities are proved in Theorems 17 and 18. \( \square \)

7. Conclusions

Two complete DFAs over different alphabets \( \Sigma' \) and \( \Sigma \) are incomplete DFAs over
\( \Sigma' \cup \Sigma \). Each DFA can be completed by adding an empty state and sending all
transitions induced by letters not in the DFA’s alphabet to that state. This results
in an \( (m + 1) \)-state DFA and an \( (n + 1) \)-state DFA. We have shown that the tight
bounds for boolean operations are \( (m + 1)(n + 1) \) for union and symmetric difference,
\( mn + m \) for difference, and \( mn \) for intersection, while the tight bound for product
is \( m2^n + 2^n - 1 \). In the same-alphabet case the tight bound is \( mn \) for all boolean
operations and it is \( m2^n - 2^{n-1} \) for product. In the case of all three types of ideals,
the unrestricted bounds for boolean operations are the same as those for arbitrary
regular languages. The bounds for product are higher than in the restricted case for
all three kinds of ideals.

In summary, the restriction of identical alphabets is unnecessary and leads to
incorrect results. It should be noted that if the two languages in question already
have empty quotients, then making the alphabets the same does not require the
addition of any states, and the traditional same-alphabet methods are correct. This
is the case, for example, for prefix-free, suffix-free and finite languages.

References

30

J. A. Brzozowski, C. Sinnamon


