On Maximally Commutative Complement of a Regular Language

Paul Cabral Henry Adorna
Department of Computer Science, University of the Philippines
Velasquez Ave., UP Diliman
Quezon City 1101
ha@dcs.upd.edu.ph

ABSTRACT

Let L and K be languages over an alphabet Σ. We call K a *commutative complement* of L, denoted by L^{CC} whenever, for all $u, w \in K$, $uw, wu \notin L$. If there is no other language $H \subseteq \Sigma^*$, such that H is a commutative complement of L that contains K, then K is called a maximal commutative complement of L or L^{CC}_{max}

Not all commutative complement of a regular language is regular. But we showed that maximal commutative complement of a regular language is again a regular language. We provide construction procedure for realizing a maximal commutative complement of a regular language. Finally, as a consequence of our construction, we showed that if the minimum DFA accepting a regular language L has n states, then the maximal commutative complement L^{CC}_{max} of L will be accepted by a DFA with at most n^n number of states.

1. INTRODUCTION

Descriptional complexity is the study of measures of complexity of languages and operations. It is a measure of information required to describe a language. As there are many ways of describing a language (e.g., using regular expressions, finite automata, grammars, etc), there are many different measures of descriptional complexity.

(Deterministic) State complexity is a descriptional complexity measure for regular languages based on the deterministic finite automaton (DFA) model. It is given as the minimum number of states in a DFA that accepts the language. Other descriptional complexity measures for regular languages are regular expression size (the number of alphabetic symbols in a regular expression) and radius (the distance from the start state to the farthest state). Though these different descriptional complexities are equally significant, state complexity remains the most popular complexity measure for regular languages. Ellul [4] gave a survey of different descriptional complexities.

Some operations on regular languages preserve regularity; that is, performing such operation on regular languages results in a regular language. Operations like union and intersection, for example, preserves regularity. The study of state complexity is extended to these operations and many results have been established in this area.

The interest in the study of state complexity is primarily motivated by the desire to have a reliable estimate of the amount of memory required (the space complexity) for a resulting automata when regularity-preserving operations are applied. This is particularly crucial in pattern searching for computer virus strands and its mutants, among others.

There is a significant amount of research lately in state complexity of operations on regular languages. Yu et al. [13] systematically studied the state complexity problems of basic operations on regular languages over a general alphabet as well as over a one-letter alphabet. Later, Yu [12] studied the state complexity of basic operations on finite languages. Campeanu et al. [1] obtained results for unary alphabet in the finite case. Domaratzki [3] examined the state complexity of proportional removals such as $\frac{1}{2}L$, while Rampersad [9] examined L^2 and L^k. Campeanu et al . [2] obtained a tight bound for the state complexity of shuffle of regular languages.

There is also an interest in the study of nondeterministic state complexity [4], average state complexity [6], and descriptional complexity for nonregular languages [10].

In this paper, we introduce a new language operation, namely, *commutative complement.* For a given regular language L, we collect all strings x and y, such that xy and yx are not at all in L. We call this collection *commutative complement of* L, and we denote by L^{CC}. We provide an example illustrating that L^{CC} does not necessarily be regular language for a given regular language L. However, we proved that the maximal commutative complement of L, that is L^{CC}_{max}, for a regular language L is always regular. We use algebraic techniques to show that maximal commutative complement preserved regularity. We illustrate the construction of maximal commutative complement of a regular langauge L. Finally, as a consequence of our construction, we provide state complexity of the maximal commutative complement of a regular language L. The state complexity of L^{CC}_{max} is equal to the number of elements of the monoid $\Sigma^*/\approx_L$. This is equivalent to the number of transformatons in the transition monoid T_M, namely $|Q|^{|Q|}$.

The close connection between finite automata and algebra is very well known. Holzer and König [7] and Krawetz [8], for instance, used the algebraic structure monoid to answer state complexity problems. We will use the same method in this paper.

The paper is organized as follows: In Section 2, we defined our new (regular) language operations, namely, commutative complement and maximal commutative complement of a language. We also provided examples and give some remarks. Section 3 proved the regularity preservation of maximal commutative comlement of a regular language L. In Section 4, we illustrate how to construct a maximal commutative complement of a regular language L using the results in Section 3. Finally, we end Section 4 by providing the consequence of our construction scheme, namely, the state complexity of the maximal commutative complement of a regular language L.

We assume that the reader is familiar with the concepts in algebra as presented in [5]. Concepts and definitions from [11] are adopted in this paper.

2. DEFINITIONS AND EXAMPLES

DEFINITION 1. *Let L and K be languages over an alphabet Σ. Suppose we have $u, w \in K$, such that $uw, wu \notin L$, then we have K is a set of strings over Σ such that all the catenations of these strings are not at all in L. We will call K the* **commutative complement** *of L, and will be denoted by L^{CC}.*

Let $K' \subseteq K$, where K is commutative complement of L. It is trivial to see that K' is again a commutative complement of L. Because any catenation of any two elements of K' cannot be found in L.

DEFINITION 2. *If there is no other language $H \subseteq \Sigma^*$, such that H is a commutative complement of L that contains K, then K is called a* **maximal commutative complement** *of L, and will be denoted by L^{CC}_{max}.*

EXAMPLE 1. *Let $L = \Sigma^*$. Then the only commutative complement of this language is the trivial language, $\emptyset$. If $L = \emptyset$, then its commutative complement would be all $K \subseteq \Sigma^*$. Σ^* is the maximal commutative complement.*

EXAMPLE 2. *Let L be a set of words over $\Sigma = \{a, b\}$ that ends in a. In particular, define $L = L((a+b)^*a)$. If we take the complement of L, that is $L^C = L((a+b)^*b+\epsilon)$, then we can verify that L^C is a commutative complement of L. And this commutative complement is maximal.*

EXAMPLE 3. *Let $L = \{a^p | p \geq 3, p \text{ is a prime}\}$. It is easy to see that L is a commutative complement of itself.*

The language $K = \{a^{2n+1} | n \geq 0\}$ is also a commutative complement. Since $L \subseteq K$, therefore, L is not maximal.

The language $H = \{a^{2n} | n \geq 0\}$ is another commutative complement of L. It can easily be seen that both K and H are maximal commutative complements.

EXAMPLE 4. *Let L be the regular language accepted by the DFA*
$M = (\{1, 2, 3, \}, \{a, b\}, \delta, 1, \{1\})$ whose transition relation is described in Figure 1 It can be shown that the language L accepted by the automaton in Figure 1 does not have nontrivial commutative complement. In particular, its commutative complement is the $\emptyset$.

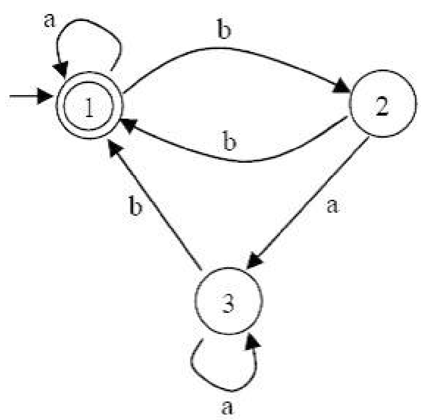

Figure 1: An automaton whose commutative complement is $\emptyset$.

REMARK 1.

- a *Any language L have at least one commutative complement language, that is the trivial language, $\emptyset$.*
- b *Commutative complement of a language L could be its complement, that is $L^{CC} = L^C$ or itself, $L^{CC} = L$.*
- c *Commutative complement of a regular language L is not necessarily regular.*
- d *The maximal commutative complement of a language L is not necessarily unique.*
- e *The maximal commutative complement of a language L could be its complement, that is, $L^{CC}_{max} = L^C$.*

In the rest of the paper, we will only be speaking of regular languages.

3. MAXIMAL COMMUTATIVE COMPLEMENTS

We will prove in this section that maximal commutative complement is a regularity preserving operation.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. This DFA induces a monoid in a natural way: for any $w \in \Sigma^*$, define a transformation $\delta_w : Q \to Q$ by $\delta_w(q) = \delta(q, w)$, for all $q \in Q$. For transformations δ_w and δ_v, we define their composition as follows: $(\delta_w \circ \delta_v)(q) = (\delta_w \delta_v)(q) = \delta_w(\delta_v(q))$. With δ_ϵ as the identity element, the set of all transformations δ_w together with the composition operator forms a monoid, called the transition monoid of M. Let T_M denote the monoid induced by the DFA M.

Since Q is finite, then T_M is also finite with at most $|Q|^{|Q|}$ number of elements. It is not hard to see that $\delta_{wv} = \delta_v \delta_w$, for all $w, v \in \Sigma^*$. Hence, T_M is generated by $\{ \delta_a \in Q^Q \mid a \in \Sigma \}$ [8].

Let $w, v \in \Sigma^*$. Define the relation $\approx_L$ where

$$w \approx_L v \Leftrightarrow \delta_w = \delta_v.$$

Clearly, $\approx_L$ is an equivalence relation on Σ^*. For $w \in \Sigma^*$, we denote the equivalence class of w by $[\delta_w]$. Note that $v \in [\delta_w]$ if and only if $\delta_v = \delta_w$. The equivalence class $[\delta_\epsilon]$ contains only one element, that is the empty word ϵ.

It is not hard to see that $\langle \Sigma^* / \approx_L, \bullet \rangle$, where for every $[\delta_w], [\delta_v] \in \Sigma^* / \approx_L$, we have $[\delta_w] \bullet [\delta_v] = [\delta_{vw}]$, forms a monoid.

Remark 2. *The mapping* $h \colon \langle \Sigma^* / \approx_L, \bullet \rangle \to \langle T_M, \circ \rangle$, *given by* $h([\delta_w]) = \delta_w$ *is an isomorphism.*

Now we will prove that maximal commutative complement of a regular language is also a regular language. First, we prove the following preliminary results.

Lemma 1. *The equivalence classes in* $\Sigma^* / \approx_L$ *are regular languages.*

Proof. For each element $[\delta_w]$ of $\Sigma^* / \approx_L$, we define a DFA $A = (T_M, \Sigma, \Delta, \delta_\epsilon, \{\delta_w\})$, where $\Delta \colon T_M \times \Sigma \to T_M$ is given by $\Delta(\delta_v, a) = \delta_a \delta_v$. To see that A accepts $[\delta_w]$, let $x \in [\delta_w]$. Then, $\delta_x = \delta_w$. By extending Δ to $\hat{\Delta} \colon T_M \times \Sigma^* \to T_M$, we have $\hat{\Delta}(\delta_\epsilon, x) = \delta_x \delta_\epsilon = \delta_x = \delta_w$, an accept state of A. It is easy to see that, $x \notin [\delta_w]$ iff $\hat{\Delta}(\delta_\epsilon, x) \neq \delta_w$. □

□

Lemma 2. *Let* L *be accepted by the DFA* $M = (Q, \Sigma, \delta, q_0, F)$. *Then,* $w \in L$ *if and only if* $\delta_w(q_0) \in F$.

The quotient $\Sigma^* / \approx_L$ separates words in Σ^* according to how they would be traced in a DFA. This means that words of similar behavior are grouped into one equivalent classes. So we ask, which elements of this quotient are commutative complements?

Theorem 1. *Let* L *be a regular language accepted by a DFA* M. *The equivalence class* $[\delta_w]$ *is a commutative complement of* L *if and only if* $\delta_w^2(q_0) \notin F$.

Proof. Let $x, y \in [\delta_w]$. Let $[\delta_w]$ is a commutative complement of L. Lemma 2, $xy \notin L$ if and only if $\delta_{xy}(q_0) \notin F$. Now, $x, y \in [\delta_w]$ implies that we have $\delta_x = \delta_y = \delta_w$. So that $\delta_{xy}(q_0) = \delta_y \delta_x(q_0) = \delta_w \delta_w(q_0) = \delta_w^2(q_0) \notin F$.

For the converse, let $x, y \in [\delta_w] \subseteq L$. Since x and y are both in $[\delta_w]$, we have, $\delta_x = \delta_y = \delta_w$. This implies that $\delta_x \delta_y = \delta_w \delta_w = \delta_w^2$. By hypothesis, $\delta_w^2(q_0) \notin F$, then $\delta_x \delta_y(q_0) = \delta_{xy}(q_0) \notin F$, too. By Lemma 2, $xy \notin L$. That $yx \notin L$ follows similarly. Therefore, $[\delta_w]$ is a commutative complement. □ □

Theorem 2. *Let* L *be a regular language. Let* $w \in \Sigma^*$. *Then,* $\{w\}$ *is a commutative complement of* L *if and only if* $[\delta_w]$ *is a commutative complement of* L.

Proof. Let $w \in \Sigma^*$ and L a regular language. $\{w\}$ is a commutative complement of L if and only if $ww = w^2 \notin L$ if and only if $\delta_{w^2}(q_0) \notin F$ if and only if $\delta_w \delta_w(q_0) = \delta_w^2(q_0) \notin F$ if and only if $[\delta_w]$ is a commutative complement of L. □ □

Theorem 3. *Let* L *be accepted by a DFA* $M = (Q, \Sigma, \delta, q_0, F)$, *let* $[\delta_w]$ *be a commutative complement of* L, *and suppose the* $v \notin [\delta_w]$. *Then the following are equivalent.*

a. $[\delta_w] \cup \{v\}$ *is a commutative complement of* L.

b. $[\delta_w] \cup [\delta_v]$ *is a commutative complement of* L.

c. $\delta_w \delta_v(q_0) \notin F$ *and* $\delta_v \delta_w(q_0) \notin F$.

Proof. We will prove the theorem in the following manner: $a \Rightarrow b \Rightarrow a$ and $b \Leftrightarrow c$.

$(a \Rightarrow b)$: Let $y \in [\delta_v]$. Then $\delta_y = \delta_v$. Since $[\delta_w] \cup \{v\}$ is a commutative complement of L, we have $wv \notin L$. By Lemma 2, $\delta_{wv}(q_0) \notin F$. But $\delta_{wv} = \delta_v \delta_w = \delta_y \delta_w = \delta_{wy}$. Hence, $\delta_{wv}(q_0) = \delta_{wy}(q_0) \notin F$. Since y was taken arbitrarily, it follows that $[\delta_w] \cup [\delta_v]$ is a commutative complement of L.

$(b \Rightarrow a)$: If $[\delta_w] \cup [\delta_v]$ is a commutative complement of L and, the fact that $v \in [\delta_v]$, then the result is obvious.

$(b \Leftrightarrow c)$: Without loss of generality, suppose that $x \in [\delta_w]$ and $y \in [\delta_v]$. This means that $\delta_x = \delta_w$ and $\delta_y = \delta_v$. That $[\delta_w] \cup [\delta_v]$ is a commutative complement of L means that neither of xy nor yx are elements of L. By Lemma 2, $xy \notin L$. and $yx \notin L$ if and only if $\delta_{xy}(q_0) \notin F$ and $\delta_{yx}(q_0) \notin F$. So that $\delta_{xy}(q_0) = \delta_y \delta_x(q_0) = \delta_v \delta_w(q_0) \notin F$. Similarly, $\delta_{yx}(q_0) = \delta_x \delta_y(q_0) = \delta_w \delta_v(q_0) \notin F$. □ □

Let us denote by Φ_L the class of elements of $\Sigma^* / \approx_L$ that are commutative complements of L. Clearly, if $[\delta_w] \in \Phi_L$ and $[\delta_v] \notin \Phi_L$, then $[\delta_w] \cup [\delta_v]$ is not a commutative complement of L. We have the following consequence of Theorem 3.

Corollary 1. *Let* $[\delta_w], [\delta_v], [\delta_z] \in \Phi_L$, *and suppose that the union* $[\delta_w] \cup [\delta_v]$ *is a commutative complement of* L. *Then,* $[\delta_w] \cup [\delta_v] \cup [\delta_z]$ *is a commutative complement of* L *if and only if both* $[\delta_w] \cup [\delta_z]$ *and* $[\delta_v] \cup [\delta_z]$ *are commutative complements of* L.

It is imperative that we can form larger commutative complements by taking unions of elements of Φ_L. Starting with a commutative complement equivalence class, we add (by taking unions) another equivalence class from Φ_L and use Corollary 1 to check wether the union is a commutative complement. If yes, we have a larger commutative complement class. Otherwise, we discard this union and proceed to evaluate the remaining equivalence classes in Φ_L. Now, we can say that $|\Phi_L| \leq |\Sigma^*/\approx_L| \leq |T_M| = |Q|^{|Q|}$. Hence, we will eventually exhaust all elements of Φ_L. We claim that the resulting union is a maximal commutative complement of L.

THEOREM 4. *Let L be a regular language. Suppose that $[\delta_w] \cup [\delta_v]$ is a commutative complement of L. If for all other eqivalence classes $[\delta_z] \in \Phi_L$, the union $[\delta_w] \cup [\delta_v] \cup [\delta_z]$ is not a commutative complement of L, then $[\delta_w] \cup [\delta_v]$ is the maximal commutative complement of L.*

PROOF. Suppose we assume the opposite, that is $[\delta_w] \cup [\delta_v]$ is not maximal. Then there must exist a larger commutative complement H that properly contains $[\delta_w] \cup [\delta_v]$; that is $H \supset [\delta_w] \cup [\delta_v]$. Let $x \in H$ and $x \notin [\delta_w] \cup [\delta_v]$. Since $\{x\} \subset H$, then $\{x\}$ must be a commutative complement of L. And by Theorem 2, $[\delta_x]$ must also be a commutative complement of L. Clearly, $[\delta_w] \cup [\delta_v] \cup \{x\}$ is a commutative complement of L. By Theorem 3, $[\delta_w] \cup [\delta_v] \cup [\delta_x]$ must also be a commutative complement of L. Then by contrapositive, the result follows. □ □

We will now show that all maximal commutative complements of L are union of some equivalence classes in Φ_L.

THEOREM 5. *Let L be the language accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ and let K be a maximal commutative complement of L. Then*

$$K = \bigcup_{\exists \delta_w \in \Phi_L} [\delta_w].$$

PROOF. Let $v \in K$. Since K is a commutative complement of L, then so are $\{v\}$ and $[\delta_v]$. Hence, to show that $K = \bigcup_{\exists \delta_w \in \Phi_L} [\delta_w]$, it suffices to show that $[\delta_v] \subseteq K$, for all $v \in K$.

Suppose there is an element x in $[\delta_v]$ not found in K. Since $x \in [\delta_v]$, we have $\delta_x = \delta_v$. For all $v, w \in K$, we know that $wv, vw \notin L$. Now, $wv, vw \notin L \Rightarrow \delta_{wv}(q_0), \delta_{vw}(q_0) \notin F \Rightarrow \delta_v\delta_w(q_0), \delta_w\delta_v(q_0) \notin F \Rightarrow \delta_w\delta_x, \delta_x\delta_w(q_0) \notin F$. Hence, $K \cup \{x\}$ is a commutative complement much larger than K, which is absurd, since K is a maximal commutative complement. □ □

Finally, we give our main result, that follows from Lemma 1, Theorem 5 and the closure property of regular languages under the operation union.

THEOREM 6. *Maximal commutative complement of a regular language is regular.*

4. CONSTRUCTION OF MAXIMAL COMMUTATIVE COMPLEMENT

We will illustrate our result in Section 3 by constructing a maximal commutative complement of a given regular language.

Let L be the regular language accepted by the DFA $M = (\{1,2,3,4\}, \{a,b\}, \delta, 1, \{3,4\})$ whose transition relation is given by the table below and whose state diagram is shown in Figure 2:

δ	a	b
1	1	2
2	3	4
3	1	2
4	3	4

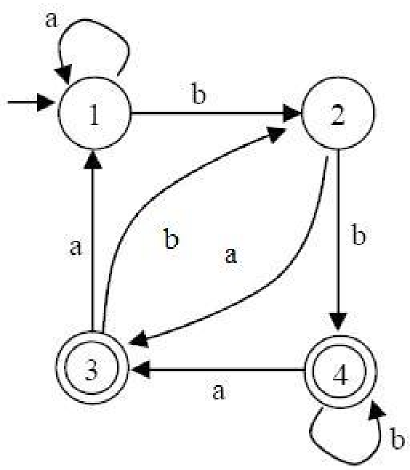

Figure 2: DFA that accepts the regular language L.

This transition function induces a transition monoid T_M that is generated by the set $\{\delta_a, \delta_b\}$, where $\delta_a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 1 & 3 \end{pmatrix}$ and $\delta_b = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 2 & 4 \end{pmatrix}$. Then we can have the following $T_M = \{\delta_\epsilon, \delta_a, \delta_b, \delta_{aa}, \delta_{bb}, \delta_{ab}, \delta_{ba}\}$ whose multiplication table is shown below:

$\circ$	δ_ϵ	δ_a	δ_b	δ_{aa}	δ_{ab}	δ_{ba}	δ_{bb}
δ_ϵ	δ_ϵ	δ_a	δ_b	δ_{aa}	δ_{ab}	δ_{ba}	δ_{bb}
δ_a	δ_a	δ_{aa}	δ_{ba}	δ_{aa}	δ_{ba}	δ_{aa}	δ_{ba}
δ_b	δ_b	δ_{ab}	δ_{bb}	δ_{ab}	δ_{bb}	δ_{ab}	δ_{bb}
δ_{aa}	δ_{aa}	δ_{aa}	δ_{aa}	δ_{aa}	δ_{aa}	δ_{aa}	δ_{aa}
δ_{ab}	δ_{ab}	δ_{ab}	δ_{ab}	δ_{ab}	δ_{ab}	δ_{ab}	δ_{ab}
δ_{ba}	δ_{ba}	δ_{ba}	δ_{ba}	δ_{ba}	δ_{ba}	δ_{ba}	δ_{ba}
δ_{bb}	δ_{bb}	δ_{bb}	δ_{bb}	δ_{bb}	δ_{bb}	δ_{bb}	δ_{bb}

For each element of $[\delta_w]$ of $\Sigma^*/\approx_L$, we check whether or not $\delta_w^2(1) \notin \{3,4\}$ (Theorem 1). So that in this case, $\Phi_L = \{[\delta_\epsilon], [\delta_a], [\delta_{aa}], [\delta_{ab}]\}$. Now, $\delta_\epsilon\delta_w = \delta_w\delta_\epsilon = \delta_w$ and $\delta_a(1) =$

$1, \delta_{aa}(1) = 1, \delta_{ab}(1) = 2$, which are not accept states, implying that $[\delta_\epsilon] \cup [\delta_a], [\delta_\epsilon] \cup [\delta_{aa}]$, and $[\delta_\epsilon] \cup [\delta_{ab}]$ are also commutative complement of L (Theorem 3). Similarly, since $\delta_a \delta_{ab} = \delta_{ba}$ and $\delta_{ba}(1) = 3 \in \{3, 4\}$, the union $[\delta_a \cup [\delta_{ab}]$ is not a commutative complement. So that only $[\delta_\epsilon] \cup [\delta_a] \cup [\delta_{aa}]$ and $[\delta_\epsilon] \cup [\delta_{aa}] \cup [\delta_{ab}]$ are the larger commutative complements (Corollary 1). Having exhausted all the elements of Φ_L, we say that these two are maximal commutative complements of L (Theorem 4), and these are the only maximal commutative complements of L (Theorem 5). Furthermore, these two maximal commutative complements are regular (Theorem 6).

We will construct the DFA for the commutative complement $K = [\delta_\epsilon] \cup [\delta_a] \cup [\delta_{aa}]$. (The construction for the DFA accepting $[\delta_\epsilon] \cup [\delta_{aa}] \cup [\delta_{ab}]$ is done similarly.) By Lemma 1, the DFA $M = (T_M, \Sigma, \Delta, \delta_\epsilon, \{\delta_\epsilon, \delta_a, \delta_{aa}\})$ accepts K where $\Delta(\delta_w, a) = \delta_a \delta_w$, for all $a \in \Sigma$. We provide below the table for Δ, that is the transition function table.

Δ	a	b
δ_ϵ	δ_a	δ_b
δ_a	δ_{aa}	δ_{ab}
δ_b	δ_{ba}	δ_{bb}
δ_{aa}	δ_{aa}	δ_{ab}
δ_{ab}	δ_{ba}	δ_{bb}
δ_{ba}	δ_{aa}	δ_{ab}
δ_{bb}	δ_{ba}	δ_{bb}

Using algorithms that converts DFA to a regular expression, we see that $K = L(\epsilon + a + (a+b)^* aa)$, while $L = L((a+b)^* b(a+b))$. Indeed, is a commutative complement of L, the language of words whose second to the last letter is b. (The other maximal commutative complement of L is $[\delta_\epsilon] \cup [\delta_{aa}] \cup [\delta_{ab}] = L(\epsilon + (a+b)^* aa + (a+b)^*)ab))$.

We note that the DFA produced here, as shown in Figure 3, is not with minimal number of states. For all $w \in \Sigma^*$, it is easy to see that $\Delta(\delta_b, w) \in F$ if and only if $\Delta(\delta_{bb}, w) \in F$ if and only if $\Delta(\delta_{ab}, w) \in F$, so that the given DFA can still be reduced to 5-state DFA.

We end this section by the following result on the upper bound of the state complexity L^{CC}_{max}.

THEOREM 7. *If the minimun DFA accepting a regular language L has n states, then the maximal commutative complement K of L will be accepted by a DFA with at most* n^n *number of states.*

This result which follows from series of results in Section 3, provides the above naive upper bound. Note that this upper bound is the cardinality of the monoid $\Sigma^* / \approx_L$, or equivalently, this is the number of transformations in the transition monoid T_M, which is $|Q|^{|Q|}$.

5. CONCLUSION

We have defined and introduced a new language operation which we called *commutative complement* L^{CC} of a language

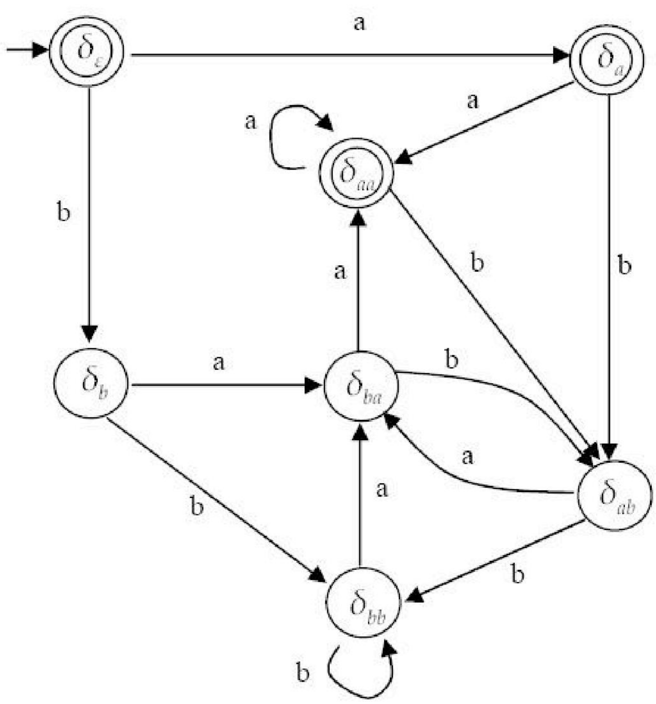

Figure 3: DFA for the commutative complement $K = [\delta_\epsilon] \cup [\delta_a] \cup [\delta_{aa}]$**.**

L. Although, we have examples showing that not for all language L, that L^{CC} will be regular, its *maximal commutative complement* that is L^{CC}_{max} will always be regular.

Theorem 7, which directly follows from the construction method we proposed provided us some upper bound the state complexity of the DFA for L^{CC}_{max}, for a regular language L. Although the construction characterizes some properties of maximal commutative complements, it is however inadequate in estimating the state complexity of a commutative complement. It is just too large an estimate. Note that in Section 4, our DFA for L has 4 states, while our resulting DFA for L^{CC}_{max} will finally have a minimum of 5 states.

Hence we ask, is there a much tighter bound than n^n for the state complexity of maximal commutative complements? Or is n^n the best possible? If yes, is there a language L whose maximal commutative complement requires at least n^n states?

In this paper, we considered only regular languages. Also, it would perhaps be interesting to see L^{CC} of some non-regular L.

6. REFERENCES

[1] C. Campeanu, K. Culik, K. Salomaa, S. Yu: State Complexity of Basic Operations on Finite Languages. Proceedings of the Fourth International Workshop on Implementing Automata VIII 1-11, 1999, LNCS 2214, Springer- Verlag, 2001, pp. 60-70.

[2] C. Campeanu, K. Salomaa, S. Yu: Tight lower bound for the state complexity of shuffle of regular languages. Journal of Automata, Languages, and Combinatorics 7, (2002) 3, pp. 303-310.

[3] M. Domaratzki: State Complexity of Proportional Removals. Journal of Automata, Languages, and Combinatorics 7 (2002), pp. 455 468.

[4] K. Ellul: Descriptional Complexity Measures of Regular Languages. Masters Thesis, University of Waterloo, 2004.

[5] T. W. Hungerford: Algebra. Springer-Verlag, 1974

[6] H. Gruber and M. Holzer: Results on the Average State and Transition Complexity of Finite Automata Accepting Finite Languages. To appear in Proceedings of the 8th Workshop on Descriptional Complexity of Formal Systems, 2006.

[7] M. Holzer and B. Knig: On Deterministic Finite Automata and Syntactic Monoid Size. In Proc. Deveopments in Language Theory 2002, M. Ito and M. Toyoma, Eds., vol. 2450 of Lecture Notes in Computer Science, Springer- Verlag, 2003, 229-240.

[8] B. Krawetz: Monoids and the State Complexity of the Operation root(L) . Masters Thesis, University of Waterloo, 2003.

[9] N. Rampersad: The state complexity of L^2 and L^k . Information Processing Letters 98, (2006) 6, pp. 231-234.

[10] J. Shallit and Y. Breitbart: Automaticity I: Properties of a Measure of Descriptional Complexity. Journal of Computer and System Sciences 53, (1996), pp. 10-25.

[11] M. Sipser: Introduction to the Theory of Computation. PWS Publishing Company, Boston, 1997.

[12] S. Yu: State Complexity of Regular Languages, Journal of Automata, Languages, and Combinatorics 6 (2001), pp. 221-324.

[13] S. Yu, Q. Zhuang, and K. Salomaa: The State Complexities of Some Basic Operations on Regular Languages. Theoretical Computer Science 125 (1994), pp. 315-328.