Duplication in DNA Sequences *

Masami Ito[1], Lila Kari[2], Zachary Kincaid[2] and Shinnosuke Seki[2]

[1] Department of Mathematics, Faculty of Science, Kyoto Sangyo University, Kyoto, Japan, 603-8555, ito@ksuvx0.kyoto-su.ac.jp
[2] Department of Computer Science, University of Western Ontario, London, Ontario, Canada, N6A 5B7, lila, sseki@csd.uwo.ca, zkincaid@uwo.ca

Abstract. Duplication and repeat-deletion are the basic models of errors occurring during DNA replication from the viewpoint of formal languages. During DNA replication, subsequences of a strand of DNA may be copied several times (duplication) or skipped (repeat-deletion). Iterated duplication and repeat-deletion have been well-studied, but little is known about single-step duplication and repeat-deletion. In this paper, we investigate properties of these operations, such as closure properties of language families in the Chomsky hierarchy, language equations involving these operations. We also make progress towards a characterization of regular languages that are generated by duplicating a regular language.

1 Introduction

Duplication grammars and duplication languages have recently received a great deal of attention in the formal language theory community. Duplication grammars, defined in [12], model duplication using string rewriting systems. Several properties of languages generated by duplication grammars were investigated in [12] and [13]. Another prevalent model for duplication is a unary operation on words [1], [2], [5], [7], [8], [9]. The biological phenomenon which motivates the research on duplication is a common error occurring during DNA replication: the insertion or deletion of repeated subsequences in DNA strands, [3].

A DNA single strand is a string over the DNA alphabet of bases $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. Due to the Watson-Crick complementarity $\mathtt{A}$—$\mathtt{T}$, $\mathtt{C}$—$\mathtt{G}$, two complementary DNA single strands of opposite orientation can bind to each other to form a DNA double strand. DNA replication is the process by which given a "template" DNA strand, an enzyme called DNA polymerase creates a new "nascent" DNA strand that is a complement of the template. To be more precise, a special short DNA single strand called a "primer" is attached to the template as a toe-hold, and then DNA polymerase adds complementary bases to the template strand, one by one, until the entire template strand becomes double-stranded.

* This research was supported by Grant-in-Aid for Scientific Research No. 19-07810 by Japan Society for the Promotion of Sciences and Research Grant No. 015 by Kyoto Sangyo University to M. I., and The Natural Sciences and Engineering Council of Canada Discovery Grant and Canada Research Chair Award to L.K.

It has been observed that errors can happen during this process, the most common of them being repeat insertions and deletions of bases. The "strand slippage model" that was proposed as an explanation of these phenomena suggests that these errors are caused by misalignments between the template and nascent strands during replication. DNA polymerase is not known to have any "memory" to remember which base on the template has been just copied onto the nascent strand, and hence the template and nascent strands can *slip*. As such, the DNA polymerase may copy a part of the template twice (resulting in an insertion) or forget to copy it (deletion). These errors occur most frequently on repeated sequences so that they are appropriately modelled by the rewriting rules $u \to uu$ and $uu \to u$.

The rule $u \to uu$ is a natural model for duplication, and the rule $uu \to u$ models the dual of duplication, which we call *repeat-deletion*. Since strand slippage is responsible for both these operations, it is natural to study both duplication and repeat-deletion. Repeat-deletion has already been extensively studied, e. g. , in [6]. However, the existing literature addresses mainly the iterated application of both repeat-deletion and duplication. This paper investigates the effects of a *single* duplication or repeat-deletion. This restriction introduces subtle new complexities into languages that can be obtained as a duplication or repeat-deletion of a language.

The paper is organized as follows: In Section 2 we define the terminology and notations we use. Section 3 is dedicated to the closure properties of language families of the Chomsky hierarchy under duplication and repeat-deletion. In Section 4, we present and solve language equations based on these operations, and give a constructive method for obtaining maximal solutions. In Section 5 we introduce a generalization of duplication, namely controlled duplication, and investigate characterizations of regular languages that can be obtained by the duplication of a regular language. Lastly, we present some results on the relationship between duplication and primitive words.

2 Preliminaries

We provide definitions for terms and notations to be used throughout the paper. For basic concepts in formal language theory, we refer the reader to [4], [16], [18].

Let Σ be a finite alphabet, Σ^* be the set of words over Σ, and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$, where λ is the empty word. The length of a word $w \in \Sigma^*$ is denoted by $|w|$. For a non-negative integer $n \geq 0$, let $\Sigma^n = \{w \in \Sigma^* \mid |w| = n\}$ and $\Sigma^{\leq n} = \bigcup_{i=0}^{n} \Sigma^i$. A language over Σ is a subset of Σ^*. For a language $L \subseteq \Sigma^*$, the set of all (internal) factors (resp. prefixes, suffixes) of L, are denoted by $\mathrm{F}(L)$ ($\mathrm{Pref}(L)$, $\mathrm{Suff}(L)$). The complement of a language $L \subseteq \Sigma^*$, denoted by L^c, is defined as $L^c = \Sigma^* \setminus L$. We denote the families of all finite languages, regular languages, context-free languages, and context-sensitive languages by FIN, REG, CFL, and CSL, respectively. Note that FIN $\subsetneq$ REG $\subsetneq$ CFL $\subsetneq$ CSL.

A word $w \in \Sigma^+$ is said to be *primitive* if $w = v^n$ implies that $n = 1$, *i.e.*, $w = v$. A word $v \in \Sigma^+$ is called a *conjugate* of w if $v = xy$ and $w = yx$ for some $x, y \in \Sigma^*$.

For a finite automaton $A = (Q, \Sigma, \delta, s, F)$ (where Q is a state set, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $s \in Q$ is the start state, and $F \subseteq Q$ is a set of final states), let $\mathcal{L}(A)$ denote the language accepted by A. We extend δ to $\hat{\delta} : Q \times \Sigma^* \to 2^Q$ as follows: (1) $\hat{\delta}(q, \lambda) = \{q\}$ for $q \in Q$ and (2) $\hat{\delta}(q, wa) = \cup_{p \in \hat{\delta}(q,w)} \delta(p, a)$ for $q \in Q$, $w \in \Sigma^*$, and $a \in \Sigma$. For $P_1, P_2 \subseteq Q$, we define an automaton $A_{(P_1,P_2)} = (Q \cup s_0, \Sigma, \delta', s_0, P_2)$, where $s_0 \notin Q$ is a new start state and $\delta' = \delta \cup (s_0, \lambda, P_1)$. Hence, $\mathcal{L}(A_{(P_1,P_2)}) = \{w \mid \hat{\delta}(p_1, w) \cap P_2 \neq \emptyset \text{ for some } p_1 \in P_1\}$. If P_i is the singleton set $\{p_i\}$, then we may simply write p_i for $i \in \{1, 2\}$.

The aim of this paper is to investigate two operations that are defined on words and languages: *duplication* and *repeat-deletion*. The unary duplication operation is defined for a word $u \in \Sigma^*$ as follows:

$$u^{\heartsuit} = \{v \mid u = xyz, v = xyyz \text{ for some } x, z \in \Sigma^*, y \in \Sigma^+\}.$$

The duplication operation is extended to a language $L \subseteq \Sigma^*$ as $L^{\heartsuit} = \bigcup_{u \in L} u^{\heartsuit}$. Some authors, e.g., [2] require the duplicated factor y to be in a finite set of words called the *duplication scheme*. We discuss a generalization of duplication schemes which we call *controlled duplication* in Section 5.

We also define another unary operation based on the dual of the $\heartsuit$ operation. We call this operation *repeat-deletion* and denote it by $\spadesuit$, which is defined for a word $v \in \Sigma^*$ as follows:

$$v^{\spadesuit} = \{u \mid v = xyyz, u = xyz \text{ for some } x, z \in \Sigma^*, y \in \Sigma^+\}.$$

As above, for a given language $L \subseteq \Sigma^*$, we define $L^{\spadesuit} = \bigcup_{v \in L} v^{\spadesuit}$.

Previous work focused on the reflexive transitive closure of the duplication operation, which we will refer to as duplication closure. In this paper, all occurrences of $\heartsuit$ and $\spadesuit$ refer to the *single step* variations of the duplication and repeat-deletion, respectively.

3 Closure Properties

Much of the work on duplication closure has been concerned with determining which of the families of languages on the Chomsky hierarchy are closed under this operation. It is known that on a binary alphabet the family of regular languages is closed under duplication closure. In contrast, on a bigger alphabet it is still closed under n-bounded duplication closure for $n \leq 2$ but not closed under n-bounded operation closure for any $n \geq 4$. The family of context-free languages is closed under (uniformly) bounded duplication closure. The readers are referred to [5] for these results.

It is a natural first step to determine these closure properties under (single step) duplication. In this section, we show that the family of regular languages is closed under repeat-deletion but not duplication, the family of context-free

languages is not closed under either operation, and the family of context-sensitive languages is closed under both operations.

The following two propositions are due to [17] (without proofs).

Proposition 1. REG *is not closed under duplication.*

Proposition 2. CFL *is not closed under duplication.*

The proof of Proposition 1 requires an alphabet that is at least binary. As we shall see in Section 5, this bound is strict. That is, the family of regular languages over a unary alphabet is closed under duplication. In addition, we have:

Proposition 3. CSL *is closed under duplication.*

In the following, we consider the closure properties of the language families on the Chomsky hierarchy under repeat-deletion. Our first goal is to prove that the family of regular languages is closed under repeat-deletion. For this purpose, we define the following binary operation $\natural$ on languages $L, R \subseteq \Sigma^*$:

$$L \natural R = \{xyz \mid xy \in L, yz \in R, y \neq \lambda\}.$$

Proposition 4. REG *is closed under* $\natural$.

Proof. Let $L_1, L_2 \in$ REG. Let $\# \notin \Sigma$ and let h be defined by $h(a) = a$ for $a \in \Sigma^*$ and $h(\#) = \lambda$. Let $L_1' = L_1 \leftarrow \{\#\} = \{u\#v \mid uv \in L_1\}$ ($\leftarrow$ denotes the insertion operation) and $L_2' = L_2 \leftarrow \{\#\}$. Moreover, let $\overline{L_1} = L_1'\#\Sigma^*$ and let $\overline{L_2} = \Sigma^*\#L_2'$. Then $L_1 \natural L_2 = h(\overline{L_1} \cap \overline{L_2})$. Since REG is closed under insertion, concatenation, intersection, and homomorphism, $L_1 \natural L_2$ is regular. □

For a regular language L, there is a finite automaton $A = (Q, \Sigma, \delta, s, F)$ such that $\mathcal{L}(A) = L$. Recall that for any state $q \in Q$, $\mathcal{L}(A_{(s,q)}) = \{w \mid q \in \hat{\delta}(s, w)\}$ and $\mathcal{L}(A_{(q,F)}) = \{w \mid \hat{\delta}(q, w) \cap F \neq \emptyset\}$. Intuitively, $\mathcal{L}(A_{(s,q)})$ is the set of words accepted "up to q", and $\mathcal{L}(A_{(q,F)})$ is the set of words accepted "after q" so that $\mathcal{L}(A_{(s,q)})\mathcal{L}(A_{(q,F)}) \subseteq L$ is the set of words in L which have a derivation that passes through state q.

Lemma 1. *Let L be a regular language and $A = (Q, \Sigma, \delta, s, F)$ be a finite automaton accepting L. Then $L^{\spadesuit} = \bigcup_{q \in Q} \mathcal{L}(A_{(s,q)}) \natural \mathcal{L}(A_{(q,F)})$.*

Proof. Let $L' = \bigcup_{q \in Q} \mathcal{L}(A_{(s,q)}) \natural \mathcal{L}(A_{(q,F)})$. First we prove that $L^{\spadesuit} \subseteq L'$. Let $\alpha \in L^{\spadesuit}$. Then there exists a decomposition $\alpha = xyz$ for some $x, y, z \in \Sigma^*$ such that $xyyz \in L$ and $y \neq \lambda$. Since A accepts $xyyz$, there exists some $q \in Q$ such that $q \in \hat{\delta}(s, xy)$ and $\hat{\delta}(q, yz) \cap F \neq \emptyset$. By construction, $xy \in \mathcal{L}(A_{(s,q)})$ and $yz \in \mathcal{L}(A_{(q,F)})$. This implies that $xyz \in \mathcal{L}(A_{(s,q)}) \natural \mathcal{L}(A_{(q,F)})$, from which we have $L^{\spadesuit} \subseteq L'$.

Conversely, if $\alpha \in L'$, then there exists $q \in Q$ such that $\alpha \in \mathcal{L}(A_{(s,q)}) \natural \mathcal{L}(A_{(q,F)})$. We can decompose α into xyz for some $x, y, z \in \Sigma^*$ such that $xy \in \mathcal{L}(A_{(s,q)})$, $yz \in \mathcal{L}(A_{(q,F)})$, and $y \neq \lambda$. Since $\mathcal{L}(A_{(s,q)})\mathcal{L}(A_{(q,F)}) \subseteq L$, we have that $xyyz$ belongs to L. It follows that $\alpha = xyz \in L^{\spadesuit}$ and $L' \subseteq L^{\spadesuit}$. □

The following is an immediate consequence of Proposition 4 and Lemma 1.

Proposition 5. REG *is closed under repeat-deletion.*

In contrast, the family of context-free languages is not closed under repeat-deletion, despite the following proposition.

Proposition 6. CFL *is closed under* $\natural$ *with regular languages.*

Lemma 2. CFL *is not closed under* $\natural$.

Proof. Let $L_1 = \{a^i\#b^i\$ \mid i \geq 0\}$ and $L_2 = \{\#b^j\$c^j \mid j \geq 0\}$. Although L_1 and L_2 are CFLs, $L_1 \natural L_2 = \{a^i\#b^i\$c^i \mid i \geq 0\}$, which is not context-free. □

Proposition 7. CFL *is not closed under repeat-deletion.*

Proof. Let $L = \{a^i\#b^i\#b^jc^j \mid i,j \geq 0\}$, which is context-free. Then $L^{\spadesuit} \cap a^*\#b^*c^* = \{a^i\#b^jc^j \mid i,j \geq 0, i \leq j\}$, which is not context free. Since CFL is closed under intersection with regular languages, and since $L^{\spadesuit} \cap a^*\#b^*c^*$ is not context-free, we conclude that $L^{\spadesuit}$ is not context-free. □

Proposition 8. CSL *is closed under repeat-deletion.*

In summary, the following closure properties of duplication, repeat-deletion, and the $\natural$ operation hold:

	$\heartsuit$	$\spadesuit$	$\natural$	$\natural$ with regular
FIN	Y	Y	Y	N
REG	N	Y	Y	Y
CFL	N	N	N	Y
CSL	Y	Y	Y	Y

4 Language Equations

We now consider the language equation problem posed by duplication: for a given language $L \subseteq \Sigma^*$, can we find a language $X \subseteq \Sigma^*$ such that $X^{\heartsuit} = L$? In the following, we show that if L is a regular language and there exists a solution to $X^{\heartsuit} = L$, then we can compute a maximal solution. We note that the solution to the language equation is not unique in general.

Example 1. $\{aaa, aaaa, aaaaa\}^{\heartsuit} = \{aaa, aaaaa\}^{\heartsuit} = \{a^i : 4 \leq i \leq 10\}$

In view of the fact that a language equation may have multiple solutions, we define an equivalence relation $\sim_{\heartsuit}$ on languages as follows:

$$X \sim_{\heartsuit} Y \Leftrightarrow X^{\heartsuit} = Y^{\heartsuit}.$$

For the same reason, we define an equivalence relation $\sim_{\spadesuit}$ as follows:

$$X \sim_{\spadesuit} Y \Leftrightarrow X^{\spadesuit} = Y^{\spadesuit}.$$

Lemma 3. *The equivalence classes of $\sim_\heartsuit$ are closed under arbitrary unions. That is, if $[X] \in 2^{\Sigma^*}/\sim_\heartsuit$ and if $\Xi \subseteq [X]$ ($\Xi \neq \emptyset$), then $\bigcup_{L\in\Xi} L \in [X]$.*

Corollary 1. *For an equivalence class $[X] \in 2^{\Sigma^*}/\sim_\heartsuit$, there exists a unique maximal element $X_{\max}$ with respect to the set inclusion partial order defined as follows:*

$$X_{\max} = \bigcup_{L\in[X]} L.$$

We provide a way to construct the maximum element of a given equivalence class. First, we prove a more general result.

Proposition 9. *Let $L \subseteq \Sigma^*$, and let $f, g : \Sigma^* \to 2^{\Sigma^*}$ be any functions such that $u \in g(v) \Leftrightarrow v \in f(u)$ for all $u, v \in \Sigma^*$. If a solution to the language equation $\bigcup_{x\in X} f(x) = L$ exists, then the maximum solution (with respect to the set inclusion partial order) is given by $X_{\max} = \left(\bigcup_{y\in L^c} g(y)\right)^c$.*

Proof. For two languages $X, Y \subseteq \Sigma^*$ such that $\bigcup_{x\in X} f(x) = L$ and $\bigcup_{y\in Y} f(y) = L$, $\bigcup_{z\in X\cup Y} f(z) = L$ holds. Hence the assumption implies the existence of $X_{\max}$.

($\subseteq$) Suppose $\exists w \in g(v) \cap X_{\max}$ for some $v \in L^c$. This means that $v \in f(w)$. However, $f(w) \subseteq \bigcup_{x\in X_{\max}} f(x) = L$, and hence $v \in L$, a contradiction. ($\supseteq$) Suppose that $\exists w \in X_{\max}^c \cap (\bigcup_{y\in L^c} g(y))^c$. If $f(w) \subseteq L$, then $w \in X_{\max}$ (by the maximality of $X_{\max}$). Otherwise, $\exists v \in f(w) \cap L^c$. This implies that $w \in g(v) \subseteq \bigcup_{y\in L^c} g(y)$. In both cases, we have a contradiction. Therefore, we have $X_{\max}^c = \bigcup_{y\in L^c} g(y)$, *i.e.*, $X_{\max} = (\bigcup_{y\in L^c} g(y))^c$. □

Lemma 4. *Let $u, v \in \Sigma^*$. Then $u \in v^\heartsuit$ if and only if $v \in u^\spadesuit$.*

Proof. ($\Rightarrow$) If $u \in v^\heartsuit$, then there exist $x, z \in \Sigma^*$ and $y \in \Sigma^+$ such that $v = xyz$ and $u = xyyz$. Then $u^\spadesuit$ contains $xyz = v$. ($\Leftarrow$) If $v \in u^\spadesuit$, then there exist $x', z' \in \Sigma^*$ and $y' \in \Sigma^+$ such that $v = x'y'z'$ and $u = x'y'y'z'$. Then $x'y'y'z' = u \in v^\heartsuit$. □

Proposition 9 and Lemma 4 imply the following corollaries.

Corollary 2. *Let $L \subseteq \Sigma^*$. If there exists a language $X \subseteq \Sigma^*$ such that $X^\spadesuit = L$, then the maximum element $X_{\max}$ of $[X]_{\sim_\spadesuit}$ is given by $((L^c)^\heartsuit)^c$.*

Corollary 3. *Let $L \subseteq \Sigma^*$. If there exists a language $X \subseteq \Sigma^*$ such that $X^\heartsuit = L$, then the maximum element $X_{\max}$ of $[X]_{\sim_\heartsuit}$ is given by $((L^c)^\spadesuit)^c$.*

Proposition 10. *Let L, X be regular languages satisfying $X^\heartsuit = L$. Then it is decidable whether X is the maximal solution for this language equation.*

Proof. Since REG is closed under repeat-deletion and complement, the maximum solution of $X^\heartsuit = L$ given in Corollary 3, $((L^c)^\spadesuit)^c$, is regular. Since the equivalence problem for regular languages is decidable, it is decidable whether a given solution to the duplication language equation is maximal. □

Due to the fact that the family of regular languages is not closed under duplication, we cannot obtain a similar decidability result for the repeat-deletion language equation, $X^{\spadesuit} = L$. This motivates our investigation in the next section of a necessary and sufficient condition for the duplication of a regular language to be regular.

5 Controlled Duplication

In Section 4 we showed that for a given language $L \subseteq \Sigma^*$, the maximal solution of the repeat-deletion language equation $X^{\spadesuit} = L$ is given by $((L^c)^{\heartsuit})^c$. However, unlike the duplication language equation, we do not have an efficient algorithm to compute this language due to the fact that the family of regular languages is not closed under duplication. This motivates "controlling" the duplication in such a manner that duplications can occur only for some specific words. In this section, we first introduce a *controlled duplication*, together with some of its basic properties. Then we propose a possible way of characterizing regular languages whose duplication can be controlled so as to generate regular languages, and give partial answers in several particular cases.

For languages $L, C \subseteq \Sigma^*$, we define the duplication of L using the control set C as follows:

$$L^{\heartsuit(C)} = \{xyyz \mid xyz \in L, y \in C\}.$$

Note that this "controlled" duplication operation can express two variants of duplication that appear in previous literature ([8], [9]), namely uniform and length-bounded duplication. Indeed, using the notations in [8], we have $D^1_{\{n\}}(L) = L^{\heartsuit(\Sigma^n)}$ and $D^1_{\{0,1,\ldots,n\}}(L) = L^{\heartsuit(\Sigma^{\leq n})}$.

The following two lemmata are basic properties of *controlled duplication*.

Lemma 5. *Let $L, C_1, C_2 \subseteq \Sigma^*$. If $C_1 \subseteq C_2$, then $L^{\heartsuit(C_1)} \subseteq L^{\heartsuit(C_2)}$.*

Lemma 6. *Let $L, C_1, C_2 \subseteq \Sigma^*$. Then $L^{\heartsuit(C_1 \cup C_2)} = L^{\heartsuit(C_1)} \cup L^{\heartsuit(C_2)}$.*

Let L be a language and C be a control set. We say that a word $w \in C$ is *useful with respect to* L if $w \in \mathrm{F}(L)$; otherwise, it is called *useless with respect to* L. The control set C is said to *contain an infinite number of useful words with respect to* L if $|\mathrm{F}(L) \cap C| = \infty$.

Lemma 7. *Let $L \subseteq \Sigma^*$ be a language, $C \subseteq \Sigma^*$ be a control set, and C' be the set of all useless words in C with respect to L. Then $L^{\heartsuit(C)} = L^{\heartsuit(C \setminus C')}$.*

Proposition 11. *For a regular language $L \subseteq \Sigma^*$ and a regular control set $C \subseteq \Sigma^*$, it is decidable whether C contains an infinite number of useful words with respect to L.*

For a regular language L and a control set C, we now investigate a necessary and sufficient condition for $L^{\heartsuit(C)}$ to be regular. A sufficient condition is a corollary of the following result in [2]. A family of languages is called a *trio* if it is

closed under λ-free homomorphism, inverse homomorphisms, and intersections with regular languages. Note that both the families of regular languages and of context-free languages are trio.

Theorem 1 ([2]). *Any trio is closed under duplication with a finite control set.*

Corollary 4. *Let $L \subseteq \Sigma^*$ be a regular language and $C \subseteq \Sigma^*$. If there exists a finite control set $C' \subseteq \Sigma^*$ such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$, then $L^{\heartsuit(C)}$ is regular.*

Results in [15] that state that infinite repetitive languages cannot be even context-free indicate that the converse of Corollary 4 may also be true. Hence, in the remainder of this section we shall investigate the following claim:

Claim. Let $L \subseteq \Sigma^*$ be a regular language and $C \subseteq \Sigma^*$ be a control set. If $L^{\heartsuit(C)}$ is regular then there exist a finite control set $C' \subseteq \Sigma^*$ such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.

As shown in the following example, this claim generally does not hold.

Example 2. Let $\Sigma = \{a, b\}$, $L = ba^+b$, and $C = ba^+ \cup a^+b$. We can duplicate a prefix ba^i of a word $ba^jb \in L$ $(i \leq j)$ to obtain a word $ba^iba^jb \in L^{\heartsuit(C)}$. In the same way, the duplication of a suffix $a^\ell b$ of a word ba^kb $(k \geq \ell)$ results in a word $ba^kba^\ell b \in L^{\heartsuit(C)}$. Thus $L^{\heartsuit(C)} = ba^+ba^+b$. Note that L and $L^{\heartsuit(C)}$ are regular. However there exists no finite control set C' satisfying $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. This is because ba^+ba^+b can have arbitrary long repetitions of a's, and hence arbitrary long control factors are required to generate it.

Nevertheless this claim holds for several interesting cases: the case where L is finite or C contains at most a finite number of useful words with respect to L, the case of a unary alphabet $\Sigma = \{a\}$, the case $L = \Sigma^*$, and the case where the control set is "marked", i.e. there exists $a \in \Sigma$ such that $C \subseteq a(\Sigma \setminus \{a\})^*a$. In the following, we prove the direct implication of the claim for these cases (the reverse one is clear from Corollary 4).

The first case we consider is when L is finite. Then $L^{\heartsuit(C)}$ is finite and hence regular. Since $\mathrm{F}(L)$ is finite, by letting $C' = C \cap \mathrm{F}(L)$, $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. Thus the claim holds in this case. Moreover, even for an infinite L, we can reach the same conclusion if C contains at most a finite number of useful words with respect to L because C', defined as above, is finite.

Next, we consider the case of a unary alphabet. We omit the proof that is mainly based on number theory arguments.

Proposition 12. *Let $\Sigma = \{a\}$ be a unary alphabet, $L \subseteq \Sigma^*$ be a regular language, and $C \subseteq \Sigma^*$ be an arbitrary language. Then $L^{\heartsuit(C)}$ is regular, and there exists a finite control set $C' \in \mathrm{FIN}$ such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

By letting $C = \Sigma^*$, Proposition 12 implies that the family of regular languages is closed under duplication when Σ is unary.

Thirdly we prove that the claim holds for the case when $L = \Sigma^*$ (Corollary 5). This requires the following known two lemmata.

Lemma 8 ([10]). *For a primitive word p, any conjugate word of p is primitive.*

Lemma 9 ([11]). *Let p and q be primitive words with $p \neq q$ and let $i, j \geq 2$. Then $p^i q^j$ is primitive.*

For a language $C \subseteq \Sigma^*$, we define $\mathrm{Dup}(C) = \{ww \mid w \in C\}$.

Proposition 13. *Let $C \subseteq \Sigma^*$. Then $\Sigma^*\mathrm{Dup}(C)\Sigma^*$ is regular if and only if there exists a finite language C' such that $\Sigma^*\mathrm{Dup}(C')\Sigma^* = \Sigma^*\mathrm{Dup}(C)\Sigma^*$.*

Proof. The proof of 'if'-part is obvious since $\Sigma^*\mathrm{Dup}(C')\Sigma^*$ is regular. Now consider the proof of 'only if'-part. Assume $L = \Sigma^*\mathrm{Dup}(C)\Sigma^*$ is regular and consider the regular language $L \cap (\Sigma^* \setminus L\Sigma^+) \cap (\Sigma^* \setminus \Sigma^+L)$. All words in this language have a representation ww for some $w \in C$. Hence there exists $C' \subseteq C$ such that $\mathrm{Dup}(C') = L \cap (\Sigma^* \setminus L\Sigma^+) \cap (\Sigma^* \setminus \Sigma^+L)$. Notice that for any $w \in C$ there exist $w' \in C'$ and $x, y \in \Sigma^*$ such that $ww = xw'w'y$. Therefore, $\Sigma^*\mathrm{Dup}(C)\Sigma^* = \Sigma^*\mathrm{Dup}(C')\Sigma^*$.

Suppose C' is infinite. Then there exists a word $uu \in \mathrm{Dup}(C')$ with length twice that of the pumping lemma constant for $\mathrm{Dup}(C')$. So by the pumping lemma, there exists a decomposition $uu = u_1u_2u_3u_1u_2u_3$, of uu such that $u_1, u_3 \in \Sigma^*$, $u_2 \in \Sigma^+$ and $u_1u_2^iu_3u_1u_2u_3 \in \mathrm{Dup}(C')$ for any $i \in \mathbb{N}$. Notice that for any $i \in \mathbb{N}$, $u_1u_2^iu_3u_1u_2u_3$ is not primitive because it is in $\mathrm{Dup}(C')$. Consider the case $i \geq 3$. By Lemma 8, $u_2^{i-1}(u_2u_3u_1)^2$ is not primitive. Then Lemma 9 implies that u_2 and $u_2u_3u_1$ share a primitive root, say $p \in \Sigma^+$. We may now write $u_2 = p^n$ and $u_2u_3u_1 = p^m$ for some $n, m \geq 1$. Hence $u_2^{i-1}(u_2u_3u_1)^2 = p^{n(i-1)+2m}$. From Lemma 8, it follows that $u_1u_2^iu_3u_1u_2u_3 = q^{n(i-1)+2m}$, where q is a conjugate word of p. Now we have that $u_1u_2^iu_3u_1u_2u_3 = q^{n(i-1)+2m}$ is a proper prefix (and suffix) of $u_1u_2^{i+1}u_3u_1u_2u_3 = q^{ni+2m}$, which contradicts the definition of $\mathrm{Dup}(C')$. Thus C' must be finite. □

Lemma 10. *Let $C \subseteq \Sigma^*$. Then $(\Sigma^*)^{\heartsuit(C)} = \Sigma^*\mathrm{Dup}(C)\Sigma^*$.*

Proof. Let $w \in (\Sigma^*)^{\heartsuit(C)}$. Then there exist $x, y, z \in \Sigma^*$ such that $y \in C$ and $w = xyyz$. Thus, $w \in \Sigma^*\mathrm{Dup}(C)\Sigma^*$. Conversely, let $v \in \Sigma^*\mathrm{Dup}(C)\Sigma^*$. Then v is of the form $xyyz$ such that $x, z \in \Sigma^*$ and $yy \in \mathrm{Dup}(C)$ (so, $y \in C$). The duplication of y in $xyz \in \Sigma^*$ results in $xyyz = v$, and hence $v \in (\Sigma^*)^{\heartsuit(C)}$. □

The following corollary is a consequence of Proposition 13 and Lemma 10. In fact, this corollary asserts the claim in the case when $L = \Sigma^*$.

Corollary 5. *Let $C \subseteq \Sigma^*$. Then $(\Sigma^*)^{\heartsuit(C)}$ is regular if and only if there exists a finite subset $C' \subseteq C$ such that $(\Sigma^*)^{\heartsuit(C')} = (\Sigma^*)^{\heartsuit(C)}$.*

The last case we consider is that of marked duplication, where given a word w in $L^{\heartsuit(C)}$, we can deduce or at least guess the factor whose duplication generates w from a word in L according to some mark of a control set C. Here we consider a mark which shows the beginning and end of a word in C, that is, $C \subseteq \#(\Sigma \setminus \{\#\})^*\#$ for some character $\#$. For a strongly-marked duplication, where $\# \notin \Sigma$ and $L \subseteq \Sigma^*\#\Sigma^*\#\Sigma^*$, we can easily show that the existence of a finite control set provided $L^{\heartsuit(C)}$ is regular using the pumping lemma for the regular language.

Hence we consider the case when the mark itself is a character in Σ, say $\# = a$ for some $a \in \Sigma$.

We introduce several needed notions related to controlled duplication. Let $L \subseteq \Sigma^*$ be a language and $C \subseteq \Sigma^*$ be a control set. For a word $w \in L^{\heartsuit(C)}$, we call a tuple (x, y, z) a *dup-factorization of w with respect to L and C* if $w = xyyz$, $xyz \in L$, and $y \in C$. When L and C are clear from the context, we simply say that (x, y, z) is a dup-factorization of w. For $y \in C$, if there are $x, z \in \Sigma^*$ such that (x, y, z) is a dup-factorization of w, then we call y a *dup-factor* of w.

Proposition 14. *Let Σ be a finite alphabet of more than one character, $L \subseteq \Sigma^*$ be a regular language, and $C \subseteq a(\Sigma \setminus \{a\})^*a$ for some $a \in \Sigma$. Then $L^{\heartsuit(C)}$ is regular if and only if there exists a finite language C' such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

Proof. We consider following two syntactic equivalence relations:

$$\equiv_L = \{(u, v) \mid \forall x, y \in \Sigma^*, xuy \in L \Leftrightarrow xvy \in L\},$$
$$\equiv_{\heartsuit} = \{(u, v) \mid \forall x, y \in \Sigma^*, xuy \in L^{\heartsuit(C)} \Leftrightarrow xvy \in L^{\heartsuit(C)}\},$$

and define $\equiv = \equiv_L \cap \equiv_{\heartsuit}$. Since both L and $L^{\heartsuit(C)}$ are regular, $C/\equiv$ is finite. Let $\Gamma_2 = \{[c] \in C/\equiv \mid |[c]| \leq 2\}$. Using induction on the number of dup-factorizations, we prove that (i) $\Gamma_2 \neq \emptyset$, and (ii) any word in $L^{\heartsuit(C)}$ has a dup-factor which is in an equivalence class in Γ_2.

Firstly, we consider a word w in $L^{\heartsuit(C)}$ which has the smallest number of dup-factorizations among the elements of $L^{\heartsuit(C)}$. Suppose that no dup-factor of w is in equivalence classes in Γ_2. Let (x, aya, z) be a dup-factorization of w for some $x, y, z \in \Sigma^*$. Then there exists $ay'a \in C$ such that $ay'a \equiv aya$, $y' \neq y$, and $ay'a \notin \mathrm{Suff}(x)$. Let $w' = xay'aayaz$. This is in $L^{\heartsuit(C)}$, and hence w' must have a dup-factorization, say $(\alpha, a\beta a, \gamma)$ for some $\alpha, \beta, \gamma \in \Sigma^*$. Due to the fact that y', y, β do not contain any a, $(a\beta a)^2$ is either (1) a factor of x, (2) a factor of z, or (3) $\beta = y$ and $a\beta a \in \mathrm{Pref}(z)$. Here we consider only the case (1), and let $x = \alpha(a\beta a)^2\gamma'$, $\gamma = \gamma' ay' aayaz$. Then $w' = \alpha(a\beta a)^2\gamma \in L^{\heartsuit(C)} \Rightarrow \alpha a\beta a\gamma' ay' aayaz \in L \Rightarrow \alpha a\beta a\gamma'(aya)^2 z \in L \Rightarrow \alpha(a\beta a)^2\gamma'(aya)^2 z = w \in L^{\heartsuit(C)}$, and hence $\alpha, a\beta a, \gamma'(aya)^2 z)$ is a dup-factorization of w. This means that a dup-factorization $(\alpha, a\beta a, \gamma)$ of w' induces a dup-factorization $(\alpha_0, a\beta a, \gamma_0)$ of w, where a single occurrence of y' in either α or γ is replaced by y to obtain α_0 and γ_0. The original dup-factorization (x, aya, z) of w cannot be obtained this way. Hence w' has a smaller number of dup-factorizations than w, a contradiction. Thus w has a dup-factor which is in an equivalence class in Γ_2, and hence $\Gamma_2 \neq \emptyset$.

Now we assume that all words in $L^{\heartsuit(C)}$ with at most n dup-factorizations have a dup-factor which is in an equivalence class in Γ_2. Suppose that there were $v \in L^{\heartsuit(C)}$ with $n+1$ dup-factorizations and without any dup-factor which is in the equivalence class of size at most 2. Then we can construct a word v' as above which has at most n dup-factorizations but does not satisfy the assumption, which is a contradiction. □

Note that the property of a control set required in this proof is that none of its elements "overlap" with each other. That is, we can use a similar proof to

settle a more general case where a control set is non-overlapping and an infix code. (See [18] for definitions.)

Corollary 6. *Let L be a regular language and C be a control set such that $L^{\heartsuit(C)}$ is regular. If C is non-overlapping and an infix code, then there exists a finite control set C' such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

Moreover, the proof of Proposition 14 shows that if we let $m = |C/\equiv|$, the size of finite control set C' given there is at most $2|\Gamma_2|$, which is not bigger than $2(m-1)$ because at least one equivalence class in $C/\equiv$ must have infinite cardinality. Finally, we provide a result slightly stronger than Corollary 6.

Corollary 7. *Let L be a regular language and C be a control set. If there exists a finite set $C_1 \subset C$ such that $C \setminus C_1$ is non-overlapping and an infix code, then the regularity of $L^{\heartsuit(C)}$ implies the existence of a finite control set C' such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

6 Duplication and Primitivity

There is evidently a connection between duplication, repeat-deletion, and primitive words, but the nature of this relationship is unclear. This section elucidates some of the properties of this relationship.

Proposition 15 (see, for instance, [14]). *Let $u, v \in \Sigma^+$ such that uv is primitive. Then both $u(uv)^n$ and $v(uv)^n$ are primitive for any $n \geq 2$.*

Proposition 16. *Let $w \in \Sigma^*$ be a non-primitive word. If we duplicate a factor of w which is properly shorter than the primitive root of w, then the resulting word is primitive.*

We can derive the following proposition from Lemma 9.

Proposition 17. *Let $x, y, z \in \Sigma^*$. If xyz is primitive and $xyyz$ is not primitive, then xz is primitive.*

7 Discussion and Future Work

In this paper, we studied duplication and repeat-deletion, two formal language theoretic models of insertion and deletion errors occurring during DNA replication. Specifically, we obtained the closure properties of the families of languages in the Chomsky hierarchy under these operations, the language equations of the form $X^{\heartsuit} = L$ and $X^{\spadesuit} = L$ for a given language L, and the operation of controlled duplication. In addition, we made steps towards finding a necessary and sufficient condition for a controlled duplication of a regular language to be regular.

Two problems for further investigation are: the problem of how to decide for a given language L whether the language equation $X^{\heartsuit} = L$ has a solution, and the problem of finding a necessary condition for the controlled duplication of a regular language to be regular in the general case.

Acknowledgements

We wish to express our gratitude to Dr. Zoltán Ésik for the concise proof of Proposition 4. We would also like to thank Dr. Helmut Jürgensen for discussions about the claim and Dr. Kathleen Hill for extended discussions on the biological motivation for duplication and repeat-deletion.

References

1. Dassow, J., Mitrana, V., Păun, Gh.: On the regularity of duplication closure. Bull. EATCS 69, pp. 133-136 (1999)
2. Dassow, J., Mitrana, V., Salomaa, A.: Operations and language generating devices suggested by the genome evolution. Theoretical Computer Science 270, pp. 701-738 (2002)
3. Garcia-Diaz M., Kunkel, T.A.: Mechanism of a genetic glissando: structural biology of indel mutations. Trends in Biochemical Sciences 31(4), pp. 206-214 (2006)
4. Ito, M.: Algebraic Theory of Automata and Languages. World Scientific Pub. Co. Inc. (2004)
5. Ito, M. Leupold, P., S-Tsuji, K.: Closure of language classes under bounded duplication. In Ibarra, O.H., Dang, Z. (eds.): DLT 2006, LNCS 4036, pp.238-247 (2006)
6. Leupold, P.: Duplication roots. In Harju, T. Karhumäki, J., and Lepistö, A. (eds.): DLT 2007, LNCS4588, pp.290-299 (2007)
7. Leupold, P.: Languages generated by iterated idempotencies and the special case of duplication. Ph.D. thesis, Department de Filologies Romaniques, Facultat de Lletres, Universitat Rovira i Virgili, Tarragona, Spain (2006)
8. Leupold, P., Mitrana, V., Sempere, J.: Formal languages arising from gene repeated duplication. Aspects of Molecular Computing. Essays in Honour of Tom Head on his 70th Birthday, LNCS 2950, pp.297-308. Springer-Verlag, Berlin (2004)
9. Leupold, P., M-Vide, C., Mitrana, V.: Uniformly bounded duplication languages. Discrete Applied Mathematics 146(3), pp.301-310 (2005)
10. Lothaire, M.: Combinatorics on Words, Encyclopedia of Mathematics and its Applications 17, Addison-Wesley Publishing Co. (1983)
11. Lyndon, R.C., Schützenberger, M.P.: On the equation $a^M = b^N c^P$ in a free group. Michigan Mathematical Journal 9, pp.289-298 (1962)
12. M-Vide, C., Păun, Gh.: Duplication grammars. Acta Cybernetica 14, pp.151-164 (1999)
13. Mitrana, V., Rozenberg, G.: Some properties of duplication grammars. Acta Cybernetica 14, pp.165-177 (1999)
14. Reis, C.M., Shyr, H.J.: Some properties of disjunctive languages on a free monoid. Information and Control 37, pp.334-344 (1978)
15. Ross, R., Winklmann, K.: Repetitive strings are not context-free. R.A.I.R.O informatique théorique / Theoretical Informatics 16(3), pp.191-199 (1982)
16. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. Springer-Verlag, Berlin Heidelberg (1997)
17. Searls, D.B. The computational linguistics of biological sequences. In Hunter, L. (eds.) Artificial Intelligence and Molecular Biology, pp.47-120. AAAI Press, The MIT Press (1993)
18. Yu, S.S.: Languages and Codes. Lecture Notes, Department of Computer Science, National Chung-Hsing University, Taichung, Taiwan 402 (2005)