Duplication in DNA Sequences *

Masami Ito[1], Lila Kari[2], Zachary Kincaid[3,2] 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
[3] Department of Mathematics, University of Western Ontario, London, Ontario, Canada, N6A 5B7, zkincaid@uwo.ca

Abstract. The duplication and repeat-deletion operations are the basis of a formal language theoretic model of errors that can occur during DNA replication. During DNA replication, subsequences of a strand of DNA may be copied several times (resulting in duplications) or skipped (resulting in repeat-deletions). As formal language operations, iterated duplication and repeat-deletion of words and languages have been well-studied in the literature. However, little is known about single-step duplications and repeat-deletions. In this paper, we investigate several properties of these operations, including closure properties of language families in the Chomsky hierarchy and 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 [16], model duplication using string rewriting systems. Several properties of languages generated by duplication grammars were investigated in [16] and [17]. Another prevalent model for duplication is a unary operation on words [2], [3], [9], [11], [12], [13]. The research on duplication is motivated by errors that occur during DNA[4] replication. Duplication and repeat-deletion (also called repeat expansion and repeat contraction, i.e., insertions and deletions of tandem repeating sequence) are biologically significant because they are

* 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.

[4] A DNA single strand is a string over the DNA alphabet of bases $\{A, C, G, T\}$. Due to the Watson-Crick complementarity property of bases, wherein A is complement to T and C is complement to G, two DNA single strands of opposite orientation and exact complementary sequences can bind to each other to form a double DNA strand. This process is called base-pairing.

among the most common errors that occur during DNA replication. In general insertions and deletions have been linked to cancer and more than 15 hereditary diseases [1]. They can also have positive consequences such as a contribution to the genetic functional compensation [5]. Interestingly, the mechanisms that cause insertions and deletions are not all well understood by geneticists [4]. For example, the strand slippages at tandem repeats and interspersed repeats are well understood but the repeat expansion and contraction in tri-nucleotide repeat diseases remain unexplained.

Strand slippage is a prevalent explanation for the occurence of repeat expansions and repeat contractions during DNA replication. DNA replication is the process by which the DNA polymerase enzyme creates a new "nascent DNA strand" that is the complement of a given single strand of DNA referred to as the "template strand". The replication process begins by mixing together the template DNA strand, the DNA polymerase enzyme, a special short DNA single strand called a "primer", and sufficient individual bases that will be used as building blocks. The primer is specially designed to base-pair with the template and thus make it double-stranded for the length of the primer. The DNA polymerase will use the primer-template double-strand subsequence as a toe-hold, and will start adding complementary bases to the template strand, one by one, in one direction only, until the entire template strand becomes double-stranded. It has been observed that errors can happen during this process, the most common of them being insertions and deletions of bases. The current explanation is that these repeat expansions and repeat contractions are caused by misalignments between the template and nascent strand during replication [4]. 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). Repeat expansions and contractions occur most frequently on repeated sequences, so they are appropriately modelled by the rewriting rules $u \to uu$ and $uu \to u$, respectively.

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 [10]. 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.

This paper is organized as follows: in Section 2, we define terminology and notations to be used throughout the paper. Section 3 is dedicated to the closure properties of the 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 constructive solutions of the equation in the

case involving duplication operation and regular languages. In Section 5, we introduce a generalization of duplication, namely controlled duplication. Section 6 investigates a characterization of the regular languages that can be obtained as a duplication of a regular language. When complete, such a characterization would constructively solve the language equation involving repeat-deletion and regular languages, for a certain class of languages. Lastly, in Section 7 we present some results on the relationship between duplication, repeat-deletion, and primitive words.

The conference version of this paper was published in [8].

2 Preliminaries

We now provide definitions for terms and notations to be used throughout the paper. For basic concepts in formal language theory, we refer the reader to [6], [7], [20], and [22]. For a relation R, we denote by R^* the reflexive, transitive closure of R. Σ denotes a finite alphabet, Σ^* denotes the set of words over Σ, and Σ^+ denotes the set of words over Σ excluding the empty word λ. For a non-negative integer $n \geq 0$, Σ^n denotes the set of words of length n over Σ, and let $\Sigma^{\leq n} = \bigcup_{i=0}^{n} \Sigma^i$. The length of a word $w \in \Sigma^*$ is denoted by $|w|$. 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 $\mathrm{F}(L)$ (resp. $\mathrm{Pref}(L)$, $\mathrm{Suff}(L)$). The complement of a language $L \subseteq \Sigma^*$, denoted by L^c, is defined as $L^c = \Sigma^* \setminus L$. We denote by FIN the family of all finite languages, by REG the family of all regular languages, by CFL the family of all context-free languages, and by CSL the family of all context-sensitive languages. We note that FIN $\subsetneq$ REG $\subsetneq$ CFL $\subsetneq$ CSL.

For a finite automaton $A = (Q, \Sigma, \delta, s, F)$ (where Q is a state set, Σ is an alphabet, $\delta : Q \times \Sigma \to 2^Q$ 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 $\delta' = \delta \cup (s_0, \lambda, P_1)$. Thus,

$$\mathcal{L}(A_{(P_1,P_2)}) = \{w \mid \hat{\delta}(p_1, w) \cap P_2 \neq \emptyset \text{ for some } p_1 \in P_1\}$$

If P_i is the singleton set $\{p_i\}$, then we may simply write p_i for $i \in \{1, 2\}$.

In this paper, we investigate two operations that are defined on words and extended to languages: *duplication* and *repeat-deletion*. We employ the duplication operation $\heartsuit$ described in [2], which is defined as follows:

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

In the canonical way, the duplication operation is extended to a language $L \subseteq \Sigma^*$:

$$L^{\heartsuit} = \bigcup_{u \in L} u^{\heartsuit}.$$

We also define another unary operation based on the dual of the $\heartsuit$ operation. We term this operation *repeat-deletion* and denote it by $\spadesuit$. Note that while biologists refer to this process simply as deletion, in formal language theory, the term deletion typically refers to removing arbitrary (rather than repeated) factors of word.

Definition 1. *For a word $v \in \Sigma^*$, the language generated by repeat-deletion of v is defined*

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

Again, the repeat-deletion operation is extended to languages: for a given language $L \subseteq \Sigma^*$,

$$L^{\spadesuit} = \bigcup_{v \in L} v^{\spadesuit}$$

We avoid inverse notation because $\heartsuit$ and $\spadesuit$ are not inverses when considered as operations on languages. That is, for a language $L \subseteq \Sigma^*$, $L \subseteq (L^{\heartsuit})^{\spadesuit}$ but it is not always the case that $L = (L^{\heartsuit})^{\spadesuit}$.

Example 1. Let $L = a^*bb$. Then $abb \in L \Rightarrow aabb \in L^{\heartsuit}$. Therefore $aab \in (L^{\heartsuit})^{\spadesuit}$, but $aab \notin L$.

Previous work focussed on the reflexive transitive closure of the duplication operation, which we will refer to as duplication closure. All occurrences of $\heartsuit$, duplication, $\spadesuit$, and repeat-deletion refer to the *single step* variations of the operations.

3 Closure Properties

Much of the work on duplication has been concerned with determining which of the families of languages on the Chomsky hierarchy are closed under duplication closure. It is known that, on a binary alphabet, the family of regular languages is closed under duplication closure. In contrast, on a larger alphabet, REG is still closed under n-bounded duplication closure for $n \leq 2$, but REG is 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 refered to [9] 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 [21] (without proofs):

Proposition 1. *The family of regular languages is not closed under duplication.*

Proof. Let $L = ab^*$ and suppose that $L^\heartsuit$ is regular. Since the family of regular languages is closed under intersection, $L' = L^\heartsuit \cap ab^*ab^*$ is regular. But L' is exactly the language $\{ab^iab^j : i \leq j\}$, which is clearly not regular. So by contradiction, $L^\heartsuit$ is not regular, and the family of regular languages is not closed under duplication. □

Note that the proof of the preceding proposition requires that the alphabet contain at least two letters. As we shall see in Section 6, this bound is tight: the family of regular languages over a unary alphabet is closed under duplication.

Proposition 2. *The family of context-free languages is not closed under duplication.*

Proof. Let $L = \{a^ib^i \mid i \geq 1\}$, a context-free language. Suppose $L^\heartsuit$ is context-free. Since the family of context-free languages is closed under intersection with regular languages, $D = L^\heartsuit \cap \{a^*b^*a^*b^*\}$ is context free.

Let p be the pumping-lemma constant of the language D. Consider the word $z = a^pb^pa^pb^p \in D$. We can decompose z as $z = uvwxy$ such that vx is a pumped part. Let $z_i = uv^iwx^iy$. Firstly, v must not contain both a and b; otherwise pumping v results in a word with more than two repetitions of a^ib^j for some $i, j \geq 1$. This also applies to x. Secondly, vx must be within the central b^pa^p part; otherwise, the pumped vx causes a difference between the number of first as and the number of last bs. Now we know that vwx is within the central b^pa^p part of z, and $v = b^i$ and $x = a^j$ for some $0 \leq i, j \leq p$ (with i, j not both zero). Then $z_2 = a^pb^{p+i}a^{p+j}b^p$, which can not be generated by duplication of a word in L. Thus we conclude that $L^\heartsuit$ is not context-free. □

Proposition 3. *The family of context-sensitive languages is closed under duplication.*

Proof. Let L be a context-sensitive language, and A_L be a linear bounded automaton for L. Now we construct a Turing machine $A_\heartsuit$ for $L^\heartsuit$ and show that $A_\heartsuit$ is a linear bounded automaton. Indeed, for a given input $w \in \Sigma^*$, $A_\heartsuit$ nondeterministically choose $w' \in \mathrm{F}(w)$ (let $w = xw'z$ for some $x, z \in \Sigma^*$) and checks whether $w' = yy$ for some $y \in \Sigma^*$. If not, it turns down this choice. Otherwise, it deletes one of y so that the input tape has xyz. Now $A_\heartsuit$ simulates A_L on this tape, and if A_L accepts the given input, xyz, then $A_\heartsuit$ accepts $w = xyyz$. Therefore, $A_\heartsuit$ accepts w if and only if there exists a nondeterministic choice of the infix with respect to which the simulated A_L accepts the given input. Thus, $\mathcal{L}(A_\heartsuit) = L^\heartsuit$.

This construction has four steps; the choice of an infix of an input, check of whether the infix is repetitive, deletion, and the simulation of A_L. The first three steps require the workspace linear-proportional to the length of an input. In the fourth step, A_L receives an input which is shorter than the original input to $A_\heartsuit$ and A_L is a linear bounded automaton. As a result, $A_\heartsuit$ is also a linear bounded automaton. □

In the following, we consider the closure properties of the language families in 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 (Due to Z. Ésik). *The family of regular languages is closed under* $\natural$.

Proof. Let $L_1, L_2 \subseteq \Sigma^+$ be regular languages. Let $\#$ be a new letter (not in Σ) and let h be homomorphism 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 the family of regular languages is closed under insertion, concatenation, intersection, and homomorphism, $L_1 \natural L_2$ is regular. □

Let L be a regular language. We can construct a finite automaton $A = (Q, \Sigma, \delta, s, F)$ such that $\mathcal{L}(A) = L$. Recall that for any state $q \in Q$, $\mathcal{L}(A_{(s,q)}) = \{w : sw \vdash_A^* q\}$ and $\mathcal{L}(A_{(q,F)}) = \{w : \exists f \in F \text{ such that } qw \vdash_A^* f\}$. 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 that 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 $sxyyz \vdash^* qyz$ and $qyz \vdash^* f$ for some $f \in F$. 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 $L' \subseteq L^{\spadesuit}$. We conclude that $L' = L^{\spadesuit}$. □

Proposition 5. *The family of regular languages is closed under repeat-deletion.*

Proof. Since the family of regular languages is closed under finite union and the $\natural$ operation, it is closed under repeat-deletion (due to Lemma 1). □

Proposition 6. *The family of context-free languages is closed under $\natural$ with regular languages.*

Proof. Repeat the argument in the proof for Proposition 4. Since the family of context-free languages is closed under insertion, concatenation with regular languages, intersection with regular languages, and homomorphism, the family of context-free languages is closed under $\natural$ with regular languages. □

Lemma 2. *The family of context-free languages 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. *The family of context-free languages 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 $L^{\spadesuit} \cap a^*\#b^*c^* = \{a^i\#b^j c^j \mid i,j \geq 0, i \leq j\}$, which is not context free. Since the family of context-free languages is closed under intersection with regular languages, and since $L^{\spadesuit} \cap a^*\#b^*c^*$ is not context-free, we may conclude that $L^{\spadesuit}$ is not context free. Thus, the family of context-free languages is not closed under repeat-deletion. □

However, there do exist context-free (and non-regular) languages whose image under repeat deletion remains context-free. An example is shown below.

Example 2. Let $L = \{a^nb^n \mid n \geq 0\}$; this is a context-free language. Then $L^{\spadesuit} = \{a^nb^m \mid 1 \leq m < n \leq 2m\} \cup \{a^nb^m \mid 1 \leq n < m \leq 2n\}$. This $L^{\spadesuit}$ is generated by the following context-free grammar, and hence in CFL. Let $G = (\{a,b\}, \{S, X, Y, X_f, Y_f\}, P, S)$, where the set of production rules P is given by

$$\begin{aligned} S &\to X \mid Y, \\ X &\to aXb \mid aaX_fb, \\ Y &\to aYb \mid aY_fbb, \\ X_f &\to aX_fb \mid aaX_fb \mid \lambda, \\ Y_f &\to aY_fb \mid aY_fbb \mid \lambda, \end{aligned}$$

Proposition 8. *The family of context-sensitive languages is closed under repeat-deletion.*

Proof. Let L and A_L be defined as we did in Proposition 3. As $A_{\heartsuit}$ in the proposition, we construct a linear bounded automaton $A_{\spadesuit}$ for $L^{\spadesuit}$ which simulates A_L. In contrast to $A_{\heartsuit}$, $A_{\spadesuit}$ nondeterministically copies an infix of a given input w. Formally speaking, w is regarded as a catenation of x, y, z and y is duplicated so as to result in $xyyz$ on the input tape. Then $A_{\spadesuit}$ runs A_L on the tape. If A_L accepts $xyyz$, then $A_{\spadesuit}$ accepts $w = xyz$. As shown in Proposition 3, $A_{\spadesuit}$ is a linear bounded automaton. □

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

	♡	♠	♮	♮ 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 the duplication operation: 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 3. $\{aaa, aaaa, aaaaa\}^\heartsuit = \{aaa, aaaaa\}^\heartsuit = \{a^i : 4 \leq i \leq 10\}$

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. *If* $[X] \in 2^{\Sigma^*} / \sim_\heartsuit$ *and if* $\Xi \subseteq [X]$ *(*$\Xi \neq \emptyset$*), then* $\bigcup_{L \in \Xi} L \in [X]$.

Proof. Let $[X] \in 2^{\Sigma^*} / \sim_\heartsuit$ and $\Xi \subseteq [X]$ ($\Xi \neq \emptyset$). Prove that $L_\Xi = \bigcup_{L \in \Xi} L \in [X]$.

Let $Y \in \Xi$. Clearly, $Y \subseteq L_\Xi$ and so $Y^\heartsuit \subseteq {L_\Xi}^\heartsuit$. Now let $w \in {L_\Xi}^\heartsuit$. Then $\exists x, z \in \Sigma^*, y \in \Sigma^+, v \in L_\Xi$ such that $w = xyyz$ and $v = xyz$. Then there exists $Z \in \Xi$ such that $v \in Z$. Since $Y, Z \in \Xi$, $v^\heartsuit \subseteq Z^\heartsuit = Y^\heartsuit$. Then $w \in v^\heartsuit$ implies $w \in Y^\heartsuit$. Thus, ${L_\Xi}^\heartsuit \subseteq Y^\heartsuit$. We conclude that $Y^\heartsuit = {L_\Xi}^\heartsuit$ and $L_\Xi \in [X]$. □

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 L is regular and 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 REG is not closed under duplication, we cannot obtain a similar decidability result for the $X^{\spadesuit} = L$ language equation. This motivates our investigation in the next two sections of necessary and sufficient conditions for the duplication of a regular language to be regular. Indeed, in the cases when the duplication language $(L^c)^{\heartsuit}$ is regular, the solution to language equations $X^{\spadesuit} = L$, $L \in$ REG, can be constructed as described in Corollary 2.

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.

Let L, C be languages over Σ. 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 generalization of the duplication operation can express two variants of duplication that appear in previous literature, namely uniform and length-bounded duplication ([12], [13]). Indeed, using the notation in [13], we have

$$D^1_{\{n\}}(L) = L^{\heartsuit(\Sigma^n)} \text{ and } D^1_{\{0,1,\ldots,n\}}(L) = L^{\heartsuit(\Sigma^{\leq n})}.$$

This section presents basic properties of controlled duplications, some of which will turn out to be useful in Section 6 . For symmetry, we also investigate properties of controlled repeat-deletion.

Lemma 5. *Let $L \subseteq \Sigma^*$ be a language and $C_1, C_2 \subseteq \Sigma^*$ be control sets. If $C_1 \subseteq C_2$, then $L^{\heartsuit(C_1)} \subseteq L^{\heartsuit(C_2)}$.*

Lemma 6. *Let $L \subseteq \Sigma^*$ be a language and $C_1, C_2 \subseteq \Sigma^*$ be control sets. Then $L^{\heartsuit(C_1 \cup C_2)} = L^{\heartsuit(C_1)} \cup L^{\heartsuit(C_2)}$.*

Let $L \subseteq \Sigma^*$ be a language, $C \subseteq \Sigma^*$ be a control set, and $w \in C$. Then w is said to be *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 and only 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 $L^{\heartsuit(C)} = L^{\heartsuit(C \setminus C')}$.*

Proof. Lemma 6 implies $L^{\heartsuit(C)} = L^{\heartsuit(C \setminus C')} \cup L^{\heartsuit(C')}$. Since $L^{\heartsuit(C')} = \emptyset$, $L^{\heartsuit(C)} = L^{\heartsuit(C \setminus C')}$ □

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.*

Proof. Since L and C are regular, $\mathrm{F}(L)$ and hence $\mathrm{F}(L) \cap C$ are also regular. Since finiteness of a regular language is decidable, it is decidable whether or not a regular control set C contains an infinite number of useful words with respect to a language L. □

Note that if $L \subseteq \Sigma^*$, $C \subseteq \Sigma^*$ is a control set, and C contains at most a finite number of useful words with respect to L, then $C' = C \cap \mathrm{F}(L)$ is a finite language and satisfies $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. In particular, for any finite language L and any control set C, there exists a finite control set $C' \subseteq C$ satisfying $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.

We now extend our previous results on the closure properties of language families so as to accommodate the controlled duplication. Since $\heartsuit = \heartsuit_{\Sigma^*}$, we trivially have the following:

– The family of regular languages is not closed under controlled duplication.

– The family of context-free languages is not closed under controlled duplication, repeat-deletion, or $\natural$.

We conclude this section with definions of repeat-deletion and the $\natural$ operation using control sets, and by providing a few results of them.

Let $L, L_1, L_2, C \subseteq \Sigma^*$. Then

$$L^{\spadesuit(C)} = \{xyz \mid xyyz \in L, y \in C\},$$
$$L_1 \natural_C L_2 = \{xyz \mid xy \in L_1, yz \in L_2, y \in C\}.$$

It is straightforward to prove that the family of regular languages is closed under $\natural_C$ for any regular language C. Let L_1, L_2 be regular languages and form $\overline{L_1}$ and $\overline{L_2}$ as defined in the proof of Proposition 4. We see that $L_1 \natural_C L_2 = h(\overline{L_1} \cap \overline{L_2} \cap \Sigma^* \# C \# \Sigma^*)$. Furthermore, by repeating the argument in the proof of Proposition 5, we have that the family of regular languages is closed under $\spadesuit_C$ for any regular control set C.

It is simple to check that if each word in L contains a subword that is in C, $\heartsuit_C$ and $\spadesuit_C$ satisfy the requirements of Proposition 9, so that we have a procedure to find X such that $X^{\heartsuit(C)} = L$ if such an X exists.

Proposition 12. *Let $L \subseteq \Sigma^*$ be a context-free language and let $C \subseteq \Sigma^+$ be a finite control set. Then $L^{\spadesuit(C)}$ is context-free.*

Proof. Let h be the homomorphism defined by $h(a) = h(\overline{a}) = a$ for $a \in \Sigma, \overline{a} \in \overline{\Sigma}$. Then $L' = h^{-1}(L)$ is context-free. Consider $L'' = L' \cap (\Sigma^* \{u\overline{u} \mid u \in C\} \Sigma^*)$. Then L'' is context-free. Now let θ be the homomorphism defined by $\theta(a) = a$ and $\theta(\overline{a}) = \lambda$ for $a \in \Sigma$. Then $\theta(L'') = L^{\spadesuit(C)}$ and hence $L^{\spadesuit(C)}$ is context-free. □

6 Conditions for $L^{\heartsuit(C)}$ to be Regular

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. As suggested in the following example, even for a "simple" language L and a control set C, $L^{\heartsuit(C)}$ can be non-regular.

Example 4. Let $\Sigma = \{a, b\}$ and $L = \{w \in \Sigma^* \mid |w| = 0 \pmod 3\}$ and $C = \Sigma^*$. Then $L^{\heartsuit(C)} \notin$ REG.

Given a regular language L, a sufficient condition for $L^{\heartsuit(C)}$ to be regular is a corollary of the following result in [3]. A family of languages is called a *trio* if it is closed under λ-free homomorphism, inverse homomorphism, and intersection with regular languages. Note that both the families of regular languages and of context-free languages are trio.

Theorem 1 ([3]). *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.*

Given a regular language L, we now investigate necessary conditions for $L^{\heartsuit(C)}$ to be regular. Results in [19] stating 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 5. 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 $C \subseteq a(\Sigma \setminus \{a\})^*a$. Moreover, it turned out that the proof technique we employ for this fourth case can be utilized to prove that the claim holds for the case where C is nonoverlapping and an infix code, which is more general than the fourth case. In the following, we prove the direct implication of the claim for these cases (the reverse one is clear from Corollary 4).

In the case where L is finite, $L^{\heartsuit(C)}$ is finite and hence regular. Since $\mathrm{F}(L)$ is finite, by letting $C' = C \cap \mathrm{F}(L)$, we have $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. Thus the claim holds for this case. Moreover, even for an infinite L, we can say that if C contains at most a finite number of useful words with respect to L, then the claim holds because C', defined in the same manner as above, is finite. Therefore in the following we assume that L is infinite and C contains an infinite number of useful words with respect to L.

Next, we show that the claim holds in the case of a unary alphabet. We employ the following known result for this purpose.

Proposition 13 ([6]). *Let $\Sigma = \{a\}$ be a unary alphabet, and L be a language over Σ. L is regular if and only if there exists a finite set $\mathcal{N}$ of pairs of integers such that $L = \bigcup_{k \geq 0, (n,m) \in \mathcal{N}} a^{kn+m}$.*

Proposition 14. *Let Σ be a unary alphabet, say $\Sigma = \{a\}$, $L \subseteq \Sigma^*$ be a regular language, and $C \subseteq \Sigma^*$ be an arbitrary language. Then $L^{\heartsuit(C)}$ is regular, and there exists a finite context $C' \in \mathrm{FIN}$ such that $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

Proof. L being regular, there exists a finite set of pairs of integers $\mathcal{N} = \{(p_i, q_i) \mid p_i, q_i \in \mathbb{N}_0, 1 \leq i \leq n\}$ for some $n \in \mathbb{N}$ such that $L = \bigcup_{x \geq 0, (p_i,q_i) \in \mathcal{N}} a^{p_i x + q_i}$.

Let $L_i = \bigcup_{x \geq 0} a^{p_i x + q_i}$, and consider a word $a^k \in C$, where $k \in \mathbb{N}$. For some $x \geq 0$, we can apply the duplication with respect to a^k to $a^{p_i x + q_i}$ if and only if $p_i x + q_i \geq k$. The application generates $a^{p_i x + q_i + k} \in L^{\heartsuit(C)}$. Note that for $x_1, x_2 \in \mathbb{N}_0$, $p_i x_1 + q_i + k = p_i x_2 + q_i + k \pmod{p_i}$. We define a function $\psi_i : C \mapsto \{0, 1, \ldots, p_i - 1\}$ such that for $a^k \in C$, $\psi_i(a^k) = q_i + k \pmod{p_i}$. Hence, we can partition C into p_i disjoint sets depending on ψ_i. Formally speaking, $C = \bigcup_{0 \leq m < p_i} C_{i,m}$, where $C_{i,m} = \{w \in C \mid \psi_i(w) = m\}$. Now the necessary and sufficient condition mentioned above as to the applicability implies that for $a^j, a^k \in C_{i,m}$, if $j \leq k$, then $L_i^{\heartsuit(\{a^j\})} \supseteq L_i^{\heartsuit(\{a^k\})}$. Let $w_{i,m}$ be the shortest word in $C_{i,m}$. Then $L_i^{\heartsuit(\{w_{i,m}\})} = L_i^{\heartsuit(C_{i,m})}$ holds. Thus, by letting $C' = \{w_{i,m} \mid 1 \leq i \leq n, 0 \leq m < p_i\}$, we have $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. Clearly C' is finite, and hence $L^{\heartsuit(C')}$ is regular. □

By letting $C = \Sigma^*$, Proposition 14 implies that the family of regular languages is closed under duplication when Σ is unary.

Next we show that the claim holds for the case when $L = \Sigma^*$ (Corollary 5). This requires the following known two lemmata. 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 $x, y \in \Sigma^*$.

Lemma 8 ([14]). *For a primitive word p, any conjugate of p is primitive.*

Lemma 9 ([15]). *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 15. *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 with 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 $yy \in \mathrm{Dup}(C)$ (*i.e.*, $y \in C$). The duplication of y in $xyz \in \Sigma^*$ results in $xyyz = v$, and hence $v \in (\Sigma^*)^{\heartsuit(C)}$. □

The following corollary derives from Lemma 10 and Proposition 15. 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 $a \in \Sigma$.

It turned out that we could employ the proof of the claim in the case of the marked duplication for the more general case when C is a nonoverlapping and an infix code. A language L is called *non-overlapping* if $vx, yv \in L$ implies $x = y = \lambda$, and L is called *infix-code* if $L \cap (\Sigma^*L\Sigma^+ \cup \Sigma^+L\Sigma^*) = \emptyset$. That is, any elements of the language which is non-overlapping and an infix-code do not overlap each other. In the following, we prove the claim for this case.

We introduce several notions and notations used in the proof. 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. Let $\delta(w)$ be the number of dup-factorizations of w with respect to L and C. 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. Let $F_d(w)$ be the set of all dup-factors of w. Note that $|F_d(w)| \leq \delta(w)$ but the inequality may be strict.

Proposition 16. *Let L be a regular language and C be a control set which 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 $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

Proof. Let $\equiv_L$ and $\equiv_\heartsuit$ be the syntactic congruences of L and $L^{\heartsuit(C)}$, respectively, and we define $\equiv \; = \; \equiv_L \cap \equiv_\heartsuit$. Since both L and $L^{\heartsuit(C)}$ are regular, $C/\equiv$ is

finite. Let $\Gamma_2 = \{[c] \in C/\equiv \text{ s.t. } |[c]| \leq 2\}$. 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, y, z) be a dup-factorization of w. Then there exists $y' \in C$ such that $y' \equiv y$, $y' \neq y$, and $y' \notin \mathrm{Suff}(x)$. Let $w' = xy'yz$. This is in $L^{\heartsuit(C)}$, and hence w' must have a dup-factorization, say (α, β, γ) for some $\alpha, \beta, \gamma \in \Sigma^*$. Due to the non-overlapping and infix-code properties of C, β^2 is an infix of either x or yz. Here we assume that it is in x, and let $x = \alpha\beta^2 v$, $\gamma = vy'yz$ for some $v \in \Sigma^*$. Then

$$\begin{aligned} w' = \alpha\beta^2\gamma \in L^{\heartsuit(C)} &\Rightarrow \alpha\beta v y' y z \in L \\ &\Rightarrow \alpha\beta v y y z \in L \\ &\Rightarrow \alpha\beta^2 v y y z = w \in L^{\heartsuit(C)}. \end{aligned}$$

Thus, $(\alpha, \beta, vyyz)$ is a dup-factorization of w. Generally speaking, for a dup-factorization (α, β, γ) of w', w has a corresponding dup-factorization (α', β, γ) if y' is an infix of α, or (α, β, γ') otherwise (*i.e.*, y' is an infix of γ). Indeed, this means that $\delta(w') < \delta(w)$ and $F_d(w') \subseteq F_d(w)$. The first consequence is a contradiction while the second one is of importance in the induction step. The second is clear from the above discussion. In order to show the first, it is enough to prove that there do not exist two distinct dup-factorizations of w' which correspond to the same dup-factorization of w, and there exists no dup-factorization of w' which corresponds to (x, y, z).

Let $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$ be two distinct dup-factorizations of w', and consider dup-factorizations of w which correspond to them respectively (either $(\alpha_i', \beta_i, \gamma_i)$ or $(\alpha_i, \beta_i, \gamma_i')$ for each $i = 1, 2$). Firstly we prove that $(\alpha_1, \beta_1, \gamma_1') \neq (\alpha_2, \beta_2, \gamma_2')$. Suppose not, then since $w' = \alpha_1\beta_1^2\gamma_1 = \alpha_2\beta_2^2\gamma_2$, we have $\gamma_1 = \gamma_2$, a contradiction. Next we compare $(\alpha_1, \beta_1, \gamma_1')$ and $(\alpha_2', \beta_2, \gamma_2)$ (see Fig. 1). Their construction shown above implies that γ_1 and α_2 must contain y' as their infix. Hence $|\alpha_1\beta_1^2| + |y'| \leq |\alpha_2|$. Since α_2' is generated by replacing y' in α_2 with y and $\beta \neq \lambda$, we have $|\alpha_1| < |\alpha_2|$. Thus, $(\alpha_1, \beta_1, \gamma_1') \neq (\alpha_2', \beta_2, \gamma_2)$. Using the same way, we can easily check that $(\alpha_i', \beta_i, \gamma_i), (\alpha_i, \beta_i, \gamma_i') \neq (x, y, z)$.

Now we assume that for all words in $L^{\heartsuit(C)}$ which have at most n dup-factorizations have a dup-factor which is in the 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 satisfies $\delta(v') \leq n$ and $F_d(v') \subseteq F_d(v)$, which contradict with the induction assumption. □

Corollary 6. *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 $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.*

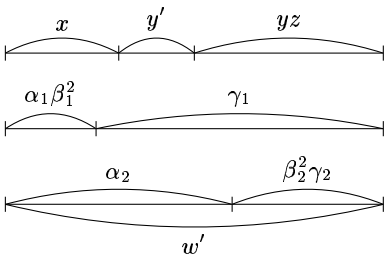

Fig. 1. The comparison between two dup-factorizations, $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$, of w'.

Proof. Note that $L^{\heartsuit(C)} = L^{\heartsuit(C_1)} \cup L^{\heartsuit(C \setminus C_1)}$. Proposition 16 implies the existence of a finite control set C_2 such that $L^{\heartsuit(C \setminus C_1)} = L^{\heartsuit(C_2)}$. Then by letting $C' = C_1 \cup C_2$, which is finite, we have $L^{\heartsuit(C)} = L^{\heartsuit(C')}$. □

Indeed, we can prove that $\Gamma_1 = \{[c] \in C/\equiv \text{ s.t. } |[c]| = 1\}$ is enough to generate $L^{\heartsuit(C)}$, that is, for a finite control set $C' = \{c \mid [c] \in \Gamma_1\}$, $L^{\heartsuit(C)} = L^{\heartsuit(C')}$.

Proposition 17. *Let L be a regular language and $C \subseteq \Sigma^*$ be a nonoverlapping and an infix code. If $L^{\heartsuit(C)}$ is regular, then $L^{\heartsuit(C)} = L^{\heartsuit(C')}$, where $C' = \{c \mid [c] \in \Gamma_1\}$.*

Proof. All we have to prove is that for $w \in L^{\heartsuit(C)}$, unless w has a dup-factor which is in C', there exists $w' \in L^{\heartsuit(C)}$ such that $\delta(w') < \delta(w)$ and $F_d(w') \subseteq F_d(w)$.

Let (x, y, z) be a dup-factorization of w, and let $y' \in C$ such that $y \neq y'$ but $y \equiv y'$. Then let $w_0 = xy'yz$, which is in $L^{\heartsuit(C)}$. The proof of Proposition 16 implies that if either (1) $y' \notin \mathrm{Suff}(x)$ or (2) $x = x_1 y'$ for some $x_1 \in \Sigma^*$ but (x_1, y', yz) is not a dup-factorization of w_0, then $\delta(w_0) < \delta(w)$. Even otherwise ($w_0 = x_1 y' y' yz$), $\delta(w_0) \leq \delta(w)$. If this holds with equality, consider $w_1 = x_1 yy' yz \in L^{\heartsuit(C)}$. If either (1) $y \notin \mathrm{Suff}(x_1)$ or (2) $x_1 = x_2 y$ for some $x_2 \in \Sigma^*$ but $(x_2, y, y'yz)$ is not a dup-factorization of w_1, then $\delta(w_1) < \delta(w_0) = \delta(w)$. Otherwise, let $w_2 = x_2 y' yy' yz$. Note that x_k is getting strictly shorter. Hence repeating this process, we eventually reach an integer $i \geq 0$ such that either (1) or (2) holds for w_i. We can check that $\delta(w_i) < \delta(w_{i-1}) \leq \cdots \leq \delta(w_0) \leq \delta(w)$ and $F_d(w_i) \subseteq F_d(w)$ as follows: Let $w_i = x_i (y'y)^{i/2+1} z \in L^{\heartsuit(C)}$ (for even i; the odd case is essentially same and hence omitted). Let $w_i = \alpha\beta^2\gamma$, where (α, β, γ) is a dup-factorization of w_i. Since either (1) or (2) holds, β^2 is an infix of x_i or that of yz. Assume the former and let $x_i = \alpha\beta^2\gamma'$ and $\gamma = \gamma'(y'y)^{i/2+1} z$. Then $\alpha\beta\gamma'(y'y)^{i/2+1} z \in L$. Using $y \equiv y'$, we can say that $\alpha\beta\gamma'(yy')^{i/2} yyz \in L$, and hence $\alpha\beta^2\gamma'(yy')^{i/2} yyz \in L^{\heartsuit(C)}$. The lefthand side is $x_i (yy')^{i/2} yyz = x_{i-1} y' (yy')^{i/2-1} yyz = \cdots = xyyz = w$. □

Consequently, we can say that if we let $m = |C/\equiv|$, then the size of finite control set C' is at most $m-1$ because at least one equivalence class in $C/\equiv$ must have infinite cardinality.

7 Duplication and Primitivity

Recall that a word $w \in \Sigma^*$ is primitive if there exists no $u \in \Sigma^*$ such that $w = u^k$ for some $k \geq 2$. We denote by Q the set of all primitive words over the alphabet Σ. There is evidently a connection between duplication, repeat-deletion, and primitive words, but the nature of this relationship is unclear. The following section elucidates some of the properties of this relationship.

Proposition 18 (see, for instance, [18]). *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 19. *Let $w \in \Sigma^*$ be a non-primitive word. If we duplicate an infix of w which is strictly shorter than the primitive root of w, then the resulting word is primitive.*

Proof. Let $w = f^n$ for $f \in Q$ and $n \geq 2$. We also denote $w = xyz$ for $x, y, z \in \Sigma^*$, where y is the infix we duplicate so that the resulting word is $xyyz$. Since $w = f^n = xyz$, there exist $f_s \in \mathrm{Suff}(f)$ and $f'_p \in \mathrm{Pref}(f)$ satisfying $y = f_s f'_p$. Then yzx, a conjugate of xyz, is written as $yzx = (f_s f_p)^n$, where $f_p \in \mathrm{Pref}(f)$ satisfying $f = f_p f_s$. Let $g = f_s f_p$. Clearly $g \in Q$. Now we prove that $yyzx$ is primitive, and hence $xyyz$ is also primitive.

We have $yyzx = f_s f'_p yzx = f_s f'_p g^n$. Since $|y| < |f|$, there exists a word $v \in \Sigma^+$ such that $f_p = f'_p v$. Then $yyzx = y(yv)^n$ and Proposition 18 implies that $yyzx$ is primitive. □

Proposition 20. *Let $x, y, z \in \Sigma^*$. If xyz is primitive and $xyyz$ is not primitive, then xz is primitive.*

Proof. Let f be the primitive root of y, *i.e.*, $y = f^m$ for some $m \geq 1$. Since $xyyz \notin Q$, its conjugate $zxyy$ is also not primitive. Suppose zx were not primitive, *i.e.*, $zx = g^n$ for some $n \geq 2$ and $g \in Q$. If $g \neq f$, then $zxyy = g^n f^{2m}$. Lemma 9 implies that $zxyy \in Q$, a contradiction. If $g = f$, then $y = g^m$ and hence $zxy = g^{n+m} \notin Q$. Thus, $xyz \notin Q$, a contradiction. As a result, $zx \in Q$, that is, $xz \in Q$. □

8 Discussion

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 our discussion on the claim and Dr. Kathleen Hill for extended discussions on the biological motivation for duplication and repeat-deletion.

References

1. Bichara, M., Wagner, J. Lambert, I.B.: Mechanisms of tandem repeat instability in bacteria. Mutation Research, 598(1-2), pp. 144-163 (2006)
2. Dassow, J., Mitrana, V., Paun, Gh.: On the regularity of duplication closure. Bull. EATCS 69, pp. 133-136 (1999)
3. Dassow, J., Mitrana, V., Salomaa, A.: Operations and language generating devices suggested by the genome evolution. Theoretical Computer Science 270, pp. 701-738 (2002)
4. 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)
5. Gu, Z., Steinmetz, L.M., Gu, X., Scharfe, G., Davis, R.W., Li, W-H.: Role of duplicate genes in genetic robustness against null mutations. Nature 421, pp.63-66 (2003)
6. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley, 1978.
7. Ito, M.: Algebraic Theory of Automata and Languages. World Scientific Pub. Co. Inc. (2004)
8. Ito, M., Kari, L., Kincaid, Z., Seki, S.: *Duplication in DNA sequences.* In Proc. DLT 2008, to appear (2008)
9. 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)
10. Leupold, P.: Duplication roots. In Harju, T. Karhumäki, J., and Lepistö, A. (eds.): DLT 2007, LNCS4588, pp.290-299 (2007)
11. 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)
12. Leupold, P., M-Vide, C., Mitrana, V.: Uniformly bounded duplication languages. Discrete Applied Mathematics 146(3), pp.301-310 (2005)
13. 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)
14. Lothaire, M.: Combinatorics on Words, Encyclopedia of Mathematics and its Applications 17, Addison-Wesley Publishing Co. (1983)

15. 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)
16. M-Vide, C., Păun, Gh.: Duplication grammars. Acta Cybernetica 14, pp.151-164 (1999)
17. Mitrana, V., Rozenberg, G.: Some properties of duplication grammars. Acta Cybernetica 14, pp.165-177 (1999)
18. Reis, C.M., Shyr, H.J.: Some properties of disjunctive languages on a free monoid. Information and Control 37, pp.334-344 (1978)
19. 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)
20. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. Springer-Verlag, Berlin Heidelberg (1997)
21. 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)
22. Yu, S.S.: Languages and Codes. Lecture Notes, Department of Computer Science, National Chung-Hsing University, Taichung, Taiwan 402 (2005)