Duplication Roots

Peter Leupold

Research Group on Mathematical Linguistics
Rovira i Virgili University
Pça. Imperial Tàrraco 1, 43005 Tarragona, Catalunya, Spain
eMail: klauspeter.leupold@urv.cat

Abstract. Recently the duplication closure of words and languages has received much interest. We investigate a reversal of it: the duplication root reduces a word to a square-free one. After stating a few elementary properties of this type of root, we explore the question whether or not a language has finite duplication root. For regular languages and uniformly bounded duplication root this is decidable.
The main result then concerns the closure of regular and context-free languages under duplication. Regular languages are closed under bounded and uniformly bounded duplication root, while neither regular nor context-free language are closed under general duplication root.

1 Duplication

A mutation, which occurs in DNA strands, is the duplication of a factor inside a strand. The interpretation of this as an operation on a string has inspired much recent work in Formal Languages, most prominently the duplication closure.

The duplication closure of a word was introduced by Dassow et al. [6], who showed that the languages generated are always regular over two letters. Wang then proved that this is not the case over three or more letters [14]. These results had actually been discovered before in the context of copy systems [8], [2]. Later on, length bounds for the duplicated factor were introduced [12], [11], and also the closure of language classes under the duplication operations was investigated [5].

In the work presented here, we investigate the effect of applying the inverse of duplications, i.e., the effect of undoing duplications leaving behind only half of a square. In this way words are reduced to square-free words, which are in some sense primitive under this notion; this is why we call the set of all square-free words reachable from a given word w the duplication root of w in analogy to concepts like the primitive root or the periodicity root of words. Duplication roots were already introduced in earlier work [11].

As the duplication root is a type of generating set with respect to duplication closure, an immediate question is whether a given language has a finite root. We show that even very simple languages generated only by catenation and iteration can have infinite duplication root. For uniformly bounded duplication roots the decidability is deduced from the fact that regular languages are closed under this operation. Also their closure under bounded duplication roots is proven.

Part of the results exposed here were presented at the Theorietag Formale Sprachen [10].

2 Duplication Roots

For applying duplications to words we use string-rewriting systems. In our notation we mostly follow Book and Otto [1] and define such a *string-rewriting system* R on Σ to be a subset of $\Sigma^* \times \Sigma^*$. Its single-step reduction relation is defined as $u \to_R v$ iff there exists $(\ell, r) \in R$ such that for some u_1, u_2 we have $u = u_1 \ell u_2$ and $v = u_1 r u_2$. We also write simpler just $\to$, if it is clear which is the underlying rewriting system. By $\xrightarrow{*}$ we denote the relation's reflexive and transitive closure, which is called the *reduction relation* or *rewrite relation*. The inverse of a single-step reduction relation $\to$ is $\to^{-1} := \{(r, \ell) : (\ell, r) \in R\}$.

The string-rewriting system we use here is the *duplication relation* $u \heartsuit v :\Leftrightarrow \exists z[z \in \Sigma^+ \wedge u = u_1 z u_2 \wedge v = u_1 z z u_2]$. If we have length bounds $|z| \leq k$ or $|z| = k$ on the factors to be duplicated we write $\heartsuit^{\leq k}$ or $\heartsuit k$ respectively; the relations are called *bounded* and *uniformly bounded duplication* respectively. $\heartsuit^*$ is the reflexive and transitive closure of the relation $\heartsuit$ and thus it is our reduction relation. The *duplication closure* of a word w is then $w^{\heartsuit} := \{u : w \heartsuit^* u\}$. The languages $w^{\heartsuit^{\leq k}}$ and $w^{\heartsuit k}$ are defined analogously.

Further notation that will be used is $IRR(R)$ for the set of words irreducible of a string-rewriting system R. For a word w with $w[i]$ we denote its i-th letter, with $w[i \dots j]$ the factor from position i to j. We call a word w *square-free* iff it does not contain any non-empty factor of the form u^2, where exponents of words refer to iterated catenation, and thus u^i is the i-fold catenation of the word u with itself. If also w^2 does not contain such a square shorter than the entire w^2, then w is said to be circular square-free. With this we come to our central definition.

Definition 1 The *duplication root* of a non-empty word w is

$$\sqrt[\heartsuit]{w} := IRR(\heartsuit^{-1}) \cap \{u : w \in u^{\heartsuit}\}.$$

As usual, this notion is extended in the canonical way from words to languages such that

$$\sqrt[\heartsuit]{L} := \bigcup_{w \in L} \sqrt[\heartsuit]{w}.$$

The roots $\sqrt[\heartsuit \leq k]{w}$ and $\sqrt[\heartsuit k]{w}$ are defined in completely analogous ways, and also these are extended to entire languages in the canonical way. When we want to contrast the duplication (root) without length bound to the bounded variants we will at times call it *general duplication (root)*. First off, we illustrate this definition with an example that also shows that duplication roots are in general not unique, i.e., the set $\sqrt[\heartsuit]{w}$ can contain more than one element.

Example 2 By undoing duplications, i.e., by applying rules from $\heartsuit^{-1}$, we obtain from the word $w = abcbabcbc$ the words in the set $\{abc, abcbc, abcbabc, abcbabcbc\}$

; in a first step either the prefix $(abcb)^2$ or the suffix $(bc)^2$ can be reduced, only the former case results in a word with another square, which can be reduced to *abc*.

Thus we have the root $\sqrt[\heartsuit]{abcbabcbc} = \{abc, abcbabc\}$. Exhaustive search of all shorter words shows that this is a shortest possible example of a word with more than one root over three letters.

Other examples with different cardinalities of the root are the words $w_3 = babacabacbcabacb$ where

$$\sqrt[\heartsuit]{w_3} = \{bacabacb, bacbcabacb, bacb\},$$

and $w_5 = ababcbabcacbabcabacbabcab$ where

$$\sqrt[\heartsuit]{w_5} = \{abcbabcabacbabcab, abcbabcab, abcacbabcab, abcabacbabcab, abcab\},$$

As the examples have finite length, the bounded duplication root is in general not unique either. The uniformly bounded duplication root, however, is known to be unique over any alphabet [11].

It is not clear at this point, whether there exist words w_k over three letters for every $k > 0$ such that $\sqrt[\heartsuit]{w_k}$ has cardinality k. Further it would be interesting to find a function relating this number k to the length of the shortest such word w_k.

3 Finiteness of the Duplication Root

In some way, duplication roots can be seen as generating sets –via duplication closure– for the given language, though not in a strict sense, because they usually generate larger sets. That is, we have $L \subseteq (\sqrt[\heartsuit]{L})^{\heartsuit}$. One of the main questions about generating sets in algebra seems especially interesting also here: does there exist a finite generating set? Or in our context: is the root finite? Trivially, duplication roots are finite over two letters.

Proposition 3 *Over a two-letter alphabet for every language L its duplication root $\sqrt[\heartsuit]{L}$ is finite.*

Proof. It is well-known that over an alphabet of two letters there exist only six non-empty square-free words. Since $\sqrt[\heartsuit]{L}$ contains only square-free words, it must be finite. □

Things become more difficult over three or more letters. Let us first define the *letter sequence* $\mathsf{seq}(u)$ of a word u as follows: any word u can be uniquely factorized as $u = x_1^{i_1} x_2^{i_2} \cdots x_\ell^{i_\ell}$ for some integers $\ell \geq 0$ and $i_1, i_2, \ldots, i_\ell \geq 1$ and for letters $x_1, x_2, \ldots, x_\ell$ such that always $x_j \neq x_{j+1}$; then $\mathsf{seq}(u) := x_1 x_2 \cdots x_\ell$. Intuitively speaking, every block of several adjacent occurrences of the same letter is reduced to just one occurrence. Notice that $\mathsf{seq}(u) = \sqrt[\heartsuit 1]{u}$. As usual, $\mathsf{seq}(L) := \bigcup_{u \in L} \mathsf{seq}(u)$ for languages L.

We now collect a few elementary properties that connect a word's letter sequence with duplication and duplication roots.

Lemma 4 *If for two words $u, v \in \Sigma^*$ we have $\mathsf{seq}(u) = \mathsf{seq}(v)$, then there exists a word w such that $u(\heartsuit^{-1})^*\ w\ \heartsuit^* v$, i.e. both u and v are reducible to w via unduplications.*

Proof. This is immediate, since every word can be reduced to its letter sequence via rules (xx, x) for $x \in \Sigma$. Thus our statement can be satisfied by setting $w = \mathsf{seq}(u)$. □

Now we state a result that links the letter sequence and the duplication root of a word in a fundamental way.

Lemma 5 *If for two words $u, v \in \Sigma^*$ we have $\mathsf{seq}(u) = \mathsf{seq}(v)$, then also $\sqrt[\heartsuit]{u} = \sqrt[\heartsuit]{v} = \sqrt[\heartsuit]{\mathsf{seq}(u)}$.*

Proof. Via rules (xx, x) for all $x \in \Sigma$ we can obviously go from u to $\mathsf{seq}(u)$. Therefore we have $\sqrt[\heartsuit]{\mathsf{seq}(u)} \subseteq \sqrt[\heartsuit]{u}$. So it remains to show the converse inclusion, and $\sqrt[\heartsuit]{\mathsf{seq}(u)} = \sqrt[\heartsuit]{u}$ will then imply our statement.

Let us suppose there exists a word $z \in \sqrt[\heartsuit]{u}$, which is not contained in $\sqrt[\heartsuit]{\mathsf{seq}(u)}$. As already stated there exists a reduction from u to $\mathsf{seq}(u)$ using only rules (xx, x) for $x \in \Sigma$. Application of these rules preserves the letter sequence of a word. There is also a reduction from u to z via rules from $\heartsuit^{-1}$. Let us look at one specific reduction of this type. As all possible reductions from u to $\mathsf{seq}(u)$ via rules (xx, x) this reduction starts in u, too. At some point –possibly already in the first step– it uses for the first time a rule (ww, w) with $|w| \geq 2$ and results in a word z'. Here this reduction becomes different from the ones to $\mathsf{seq}(u)$.

If the first and the last letter of w are different, then $\mathsf{seq}(w)^2$ is a subsequence of the letter sequence of the word, to which this rule is applied. Consequently, $\mathsf{seq}(w)^2$ is a factor of $\mathsf{seq}(u)$. Thus we can apply a rule $(\mathsf{seq}(w)^2, \mathsf{seq}(w))$ there and obtain the word $\mathsf{seq}(z')$. If the first and the last letter of w are the same, then $\mathsf{seq}(w^2) = \mathsf{seq}(w) \cdot \mathsf{seq}(w[2 \ldots |w|])$, and we obtain z' by application of the rule $(w[2 \ldots |w|]^2, w[2 \ldots |w|])$.

By Lemma 4 z' is reducible to $\mathsf{seq}(z')$, and it is still reducible to z. So we can repeat our reasoning. Because the reduction from u to z is finite, this process will terminate and show that there is a word v reachable from both z and $\mathsf{seq}(u)$ via rules from $\heartsuit^{-1}$.

But $z \in \sqrt[\heartsuit]{u}$ is irreducible under this relation, and thus we must have $v = z$. Now $\mathsf{seq}(u)(\heartsuit^{-1})^* z$ shows that $z \in \sqrt[\heartsuit]{\mathsf{seq}(u)}$. Since this contradicts our assumption, there can be no word in $\sqrt[\heartsuit]{u} \setminus \sqrt[\heartsuit]{\mathsf{seq}(u)}$, and this concludes our proof. □

In the proof, the word $\mathsf{seq}(z')$ is obtained by rules, whose left sides are not longer than the one of the simulated rule (ww, w). Therefore the same argumentation works for bounded duplication.

Corollary 6 *If for two words $u, v \in \Sigma^*$ and an integer k we have $\mathsf{seq}(u) = \mathsf{seq}(v)$, then also $\sqrt[\heartsuit_{\leq k}]{u} = \sqrt[\heartsuit_{\leq k}]{v} = \sqrt[\heartsuit_{\leq k}]{\mathsf{seq}(u)}$.*

Without further considerations, we also obtain a statement about the finiteness of the root of a language.

Corollary 7 *A language L has finite duplication root, iff $\sqrt[\heartsuit]{\mathsf{seq}(L)}$ is finite.*

If a language does not have a finite duplication root, then this root can not be of any given complexity with respect to the Chomsky Hierarchy. There is a gap between finite and context-free languages, in which no duplication root can be situated.

Proposition 8 *If a language has a context-free duplication root, then its duplication root is finite.*

Proof. For infinite regular and context-free languages the respective, well-known pumping lemmata hold. Since a duplication root consists only of square-free words, no such language can fulfill these lemmata. □

Already for the bounded case this does not hold any more. For example, for any $k \geq 1$ we can use a circular square-free word w of length greater than k; such words exist for all $k \geq 18$ [4]. Then we have $\sqrt[\heartsuit \leq k]{w^+} = w^+$, and this language is regular.

It is quite clear how the iteration of the union of several singleton sets can generate a regular language with infinite root; for the simplest case of this type consider $\{a, b, c\}^+$. We will now illustrate with an example that there are also regular languages constructed exclusively by concatenation and iteration, which have an infinite duplication root.

Example 9 We have seen in Example 2 that the root of $u = abcbabcbc$ consists of the two words $u_1 = abc$ and $u_2 = abcbabc$. Let ρ be the morphism, which simply renames letters according to the scheme $a \to b \to c \to a$. Then $\rho(u)$ has the two roots $\rho(u_1)$ and $\rho(u_2)$; similarly, $\rho(\rho(u))$ has the two roots $\rho(\rho(u_1))$ and $\rho(\rho(u_2))$.

We will now use this ambiguity to construct a word w such that $\sqrt[\heartsuit]{w^+}$ is infinite. This word over the four-letter alphabet $\{a, b, c, d\}$ is

$$w = ud\rho(u)d\rho(\rho(u))d = abcbabcbc \cdot d \cdot bcacbcaca \cdot d \cdot cabacabab \cdot d.$$

Thus the duplication root of w contains among others the three words

$$\begin{aligned} w_a &= abc \cdot d \cdot bca \cdot d \cdot cabacab \cdot d \\ w_b &= abc \cdot d \cdot bcacbca \cdot d \cdot cab \cdot d \\ w_c &= abcbabc \cdot d \cdot bca \cdot d \cdot cab \cdot d, \end{aligned}$$

which are square-free. We now need to recall that a morphism h is called square-free, iff $h(w)$ is square-free for all square-free words w. Crochemore has shown that a uniform morphism h is square-free iff it is square-free for all square-free words of length 3, [3]. Here uniform means that all images of single letters have the same length, which is given in our case.

The morphism we define now is $\varphi(x) := w_x$ for all $x \in \{a, b, c\}$. Thus to establish the square-freeness of φ, we need to check this property for the images of all square-free words up to length 3. These are

$$\begin{aligned}
\varphi(aba) &= abcdbcadcabacabdabcdbcacbcadcabdabcdbcadcabacabd\\
\varphi(abc) &= abcdbcadcabacabdabcdbcacbcadcabdabcbabcdbcadcabd\\
\varphi(aca) &= abcdbcadcabacabdabcbabcdbcadcabdabcdbcadcabacabd\\
\varphi(acb) &= abcdbcadcabacabdabcbabcdbcadcabdabcdbcacbcadcabd\\
\varphi(bab) &= abcdbcacbcadcabdabcdbcadcabacabdabcdbcacbcadcabd\\
\varphi(bac) &= abcdbcacbcadcabdabcdbcadcabacabdabcbabcdbcadcabd\\
\varphi(bca) &= abcdbcacbcadcabdabcbabcdbcadcabdabcdbcadcabacabd\\
\varphi(bcb) &= abcdbcacbcadcabdabcbabcdbcadcabdabcdbcacbcadcabd\\
\varphi(cac) &= abcbabcdbcadcabdabcdbcadcabacabdabcbabcdbcadcabd\\
\varphi(cab) &= abcbabcdbcadcabdabcdbcadcabacabdabcdbcacbcadcabd\\
\varphi(cba) &= abcbabcdbcadcabdabcdbcacbcadcabdabcdbcadcabacabd\\
\varphi(cbc) &= abcbabcdbcadcabdabcdbcacbcadcabdabcbabcdbcadcabd,
\end{aligned}$$

where, of course, the images of all words shorter than three are contained in them. All the twelve words listed here are indeed square-free as an eager reader can check, and thus φ is square-free.

Now let t be an infinite square-free word over the letters a, b and c. Then $\varphi(\mathsf{pref}(t))$ is an infinite set of square-free words. From the construction of φ we know that for any word z of length i we can reach $\varphi(z)$ from w^i by undoing duplications. Therefore $\varphi(\mathsf{pref}(t)) \subseteq \sqrt[\heartsuit]{w^+}$, whence also the latter set is infinite.

Thus even very simple languages can have rather complicated roots. Therefore it seems to be a challenging problem to decide the finiteness for regular languages in general.

4 Closure under Duplication Root

Now we investigate the closure of regular languages under the three variants of duplication root. We start with the most restricted variant, uniformly bounded duplication.

Proposition 10 *If* $L \in REG$*, then also* $\sqrt[\heartsuit k]{L} \in REG$ *for all* $k \geq 1$*.*

Proof. If a language L is regular, then it can be generated by a regular grammar $G = (N, \Sigma, S, P)$, which has only rules of the forms (A, xB) and (A, x) for non-terminals A and B and $x \in \Sigma$; for simplicity we ignore the possible rule (S, λ) to generate the empty word. From this grammar we construct another one that generates $\sqrt[\heartsuit k]{L}$.

The new grammar's set of non-terminals is $N' = \{A_w : A \in N \wedge w \in \Sigma^{\leq 2k}\}$. The rule set is derived from P in the following way. The rules from $\{(A_w, B_{wx}) : (A, B) \in P \wedge |w| < 2k - 1\}$ go in parallel to those of P, but store the letters generated in the non-terminals' index instead of actually generating them. When the index reaches length $2k$, the first letters stored in the index are finally put out, when new ones come in. This is done by rules from the set

$$\{(A_w, w[1]B_{w[2\ldots 2k]x}) : (A, xB) \in P \wedge |w| = 2k \wedge w[2\ldots 2k]x \in IRR((\heartsuit^k)^{-1})\}.$$

Only if the index would become a square u^2 of length k, then this square is reduced to u, instead of putting anything out. The set

$$\{(A_w, B_{w[1\ldots k+1]}) : (A, xB) \in P \wedge |w| = 2k \wedge w[2\ldots 2k]x \notin IRR((\heartsuit^k)^{-1})\}$$

provides the rules implementing this. The rules from

$$\{(A_w, B_{w[1\ldots k]}) : (A, xB) \in P \wedge |w| = 2k-1 \wedge w[1\ldots k] = w[k+1\ldots 2k-1]x\}$$

take care of the case that upon filling the index already a k-square is produced. From the terminating rules of P we derive the sets

$$\{(A_w, wx) : (A, x) \in P \wedge wx \text{ is not a } k-\text{square}\}$$

and

$$\{(A_w, w[1\ldots k]) : (A, x) \in P \wedge wx \text{ is a } k-\text{square}\}.$$

This new grammar obviously generates the words that also G generates, only leaving out all squares of length $2k$ that occur when going from left to right. If there are two squares of length $2k$ overlapping in more than k symbols, then the entire sequence has period k, and thus the reduction of either one results in the same word. Therefore this proceeding from left to right will indeed reduce all the squares, and in this way all the words in $\sqrt[\heartsuit k]{L}$ are reached. □

The grammar constructed for $\sqrt[\heartsuit k]{L}$ uses a similar idea as the algorithm for deciding the question "$u \in v^{\heartsuit k}$?," which we gave in an earlier article [11]. The effective closure of regular languages under uniformly bounded duplication can be used to decide the problem of the finiteness of the root for the uniformly bounded case.

Corollary 11 *For regular languages it is decidable, whether their uniformly bounded duplication root is finite.*

Proof. From the proof of Proposition 10 we see that from a regular grammar for a language L a regular grammar for the language $\sqrt[\heartsuit k]{L}$ can be constructed. This construction method is effective. Since the finiteness problem is decidable for regular languages, it can then also be decided for $\sqrt[\heartsuit k]{L}$. □

Now we turn to bounded duplication. As the bounded duplication closure of even single words is in general not regular, it is not clear whether its inverse will preserve regularity. We will show that this is the case by essentially reducing bounded duplication root to an inverse monadic string-rewriting system. A string-rewriting system is *monadic*, iff for all of its rules (ℓ, r) we have $|\ell| > |r|$ and $|r| \leq 1$. Such a system R is said to preserve regularity (context-freeness) iff for all regular languages L also the language $\bigcup_{w\in L}\{u : w \xrightarrow{*}_R u\}$ is regular (context-free).

Proposition 12 *The class of regular languages is closed under bounded duplication root.*

Proof. For a given regular language L the defining condition for a word w to be in $\sqrt[\heartsuit\leq k]{L}$ can be formulated in the following way: $w^{\heartsuit\leq k} \cap L \neq \emptyset$. This is the way we will decide the word problem for $\sqrt[\heartsuit\leq k]{L}$.

Thus we start from the input word and have to determine, whether from it one can reach another word in a given regular language L. Let the length bound for duplication be k. We will transform words from Σ^+ into a redundant representation, where every letter contains also the information about the $k-1$ following ones. This way rewrite rules from $\heartsuit^{\leq k}$ can be simulated by ones with a left side of length only one, i.e. by context-free ones.

First off we define the mapping $\phi : \Sigma^+ \mapsto ((\Sigma \cup \{\Box\})^k)^+$ as follows. We delimit with $(\ldots)$ letters from $(\Sigma \cup \{\Box\})^k$ and with $[\ldots]$ factors of a word as usual. The image of a word u is

$$u \mapsto (u[1 \ldots k])\,(u[2 \ldots k+1]) \cdots (u[|u|-k+1 \ldots |u|]) \cdot \\ (u[|u|-k+2 \ldots |u|]\Box) \cdots (u[|u|]\Box^{k-1}).$$

Thus every letter contains also the information about the k following original ones from the original word u, at the end of the word letters are filled up with the space symbol $\Box$. This encoding can be reversed by a letter-to-letter homomorphism h defined as $h(x) := x[1]$ if $x[1] \in \Sigma$, for the other case we select for the sake of completeness some arbitrary letter a and set $h(x) := a$ if $x[1] = \Box$; the latter case will never occur in our context. It is clear that $h(\phi(u)) = u$ for words from Σ^*. Both mappings are extended to languages in the canonical way such that $\phi(L) := \{\phi(u) : u \in L\}$ and $h(L) := \{h(u) : u \in L\}$.

Now we define the string-rewriting system R over the alphabet $(\Sigma \cup \{\Box\})^k$ as follows:

$$R := \{((uv), \phi(u^2v')[1 \ldots |\phi(u^2v')| - k + 1]) : uv \in \Sigma^k \wedge v' \in \Sigma^* \wedge \\ v \in v' \cdot \{\Box^*\}\}.$$

A letter $[uv]$ is replaced by the image of u^2v under ϕ minus the suffix of letters that contain $\Box$. This way application of rules from R keeps this space symbol only in the last letters of our words. It should be rather clear that $\phi(w^{\heartsuit\leq k}) = \{u : \phi(w)R^*u\}$ or, in other words $w^{\heartsuit\leq k} = h(\{u : \phi(w)R^*u\})$.

As all the rules of R have left sides of length one, the application of this system amounts to substituting each letter of w with a word from a context-free language. Since context-free languages are closed under this type of substitution, the language of all the resulting words is context-free if the original one is.

As all the rules of R have left sides of length one and right sides of length greater than one, their inverses are all monadic, i.e. the system R^{-1} is monadic. Monadic string-rewriting systems preserve regularity, see for example the work of Hofbauer and Waldmann [9].

The language $\sqrt[\heartsuit\leq k]{\Sigma^*}$ is regular as it is the complement of the regular $\Sigma^*\{uu : |u| \leq k\}\Sigma^*$.

Summarizing, we can obtain $\sqrt[\heartsuit\leq k]{L}$ by a series of regularity-preserving operations in the following way:

$$\sqrt[\heartsuit\leq k]{L} = (R^{-1})^*(\phi(L)) \cap \sqrt[\heartsuit\leq k]{\Sigma^*}.$$

Though ϕ is not a morphism, it is a gsm mapping and as such certainly preserves regularity. Thus bounded duplication root preserves regularity. □

Reducing the set R does not affect the context-freeness of the language. Thus we can take away all rules effecting duplications shorter than the length bound k and obain another proof for Proposition 10, however not a constructive one that lets us conclude the decidability. Unfortunately, the proof can not be generalized to context-free languages, because inverse monadic rewriting systems do not preserve context-freeness.

Under the unbounded root neither the regular nor the context-free languages are not closed, as for example $\sqrt[\heartsuit]{\{a,b,c\}^*}$, the language of all square-free words, is not context-free, see [7] and [13].

Summarizing we state the closure properties determined in this article in a comprehensive form.

Theorem 13 *The closure properties of the class of regular languages under the three duplication roots are as follows:*

	$\sqrt[\heartsuit k]{L}$	$\sqrt[\heartsuit \leq k]{L}$	$\sqrt[\heartsuit]{L}$
REG	*Y*	*Y*	*N*
CF	*?*	*?*	*N*

Here Y stands for closure, N stands for non-closure, and ? means that the problem is open.

5 Outlook

Theorem 13 leaves open the closure of context-free languages under bounded duplication root. The answer might actually depend on the length bound as is the case for bounded duplication. For $k \in \{1,2\}$ regular languages are closed under bounded duplication, for $k \geq 4$ they are not, while for $k = 3$ the problem remains open, see [5]. Also for context-sensitive languages the only trivial case seems to be uniformly bounded duplication root. For all others the straightforward approach of the proof of Theorem 12 cannot be used with a LBA, because it would exceed the linear length bound.

The other complex of interesting questions is the investigation of the boundary between decidability and undecidability for the question of the finiteness of duplication roots. For the simplest case, i.e. regular languages and uniformly bounded duplication, Corollary 11 establishes decidability; on the other hand, for context-free languages or even more complicated classes, one would expect undecidability for all variants of duplication. Thus sómewhere inbetween there must be the crucial point, probably at different complexities for the different variants of duplication root.

Finally, the investigation of the duplication root of single words promises to pose some interesting questions, as hinted at the end of Section 2. Especially the possible degrees of ambiguity might follow interesting rules.

Acknowledgements. The words w_3 and w_5 from Example 2 were found by Szilárd Zsolt Fazekas, the fact that w is the shortest example of a word with ambiguous root was established by Artiom Alhazov with a computer.

References

1. R. Book and F. Otto: *String-Rewriting Systems.* Springer, Berlin, 1988.
2. D.P. Bovet and S. Varricchio: *On the Regularity of Languages on a Binary Alphabet Generated by Copying Systems.* In: Information Processing Letters 44, 1992, pp. 119–123.
3. M. Crochemore: *Sharp Caracterizations of Squarefree Morphisms.* In: Theoretical Computer Science 18, 1982, pp. 221–226.
4. J.D. Currie: *There are Ternary Circular Square-free Words of Length n for $n \geq$* 18. In: Electric Journal of Combinatorics, 9(1) N10, 2002.
5. M. Ito, P. Leupold, and K. Shikishima-Tsuji: *Closure of Language Classes under Bounded Duplication.* In: Lecture Notes in Computer Science 4036, DLT 2006, Springer, Berlin, pp. 238–247.
6. J. Dassow, V. Mitrana and Gh. Păun: *On the Regularity of Duplication Closure.* In: Bull. EATCS 69, 1999, pp.133–136.
7. A. Ehrenfeucht and G. Rozenberg : *On the Separating Power of EOL Systems.* In: RAIRO Informatique Théorique, Vol. 17, No. 1, 1983, pp. 13–22.
8. A. Ehrenfeucht and G. Rozenberg: *On Regularity of Languages Generated by Copying Systems.* In: Discrete Applied Mathematics 8, 1984, pp. 313–317.
9. D. Hofbauer and J. Waldmann: *Deleting String-Rewriting Systems preserve regularity.* In: Theoretical Computer Science 327, 2004, pp. 301-317.
10. P. Leupold: *Duplication Roots – Extended Abstract.* In: Proceedings Theorietag Automaten und Formale Sprachen, TU Wien, Wien, 2006, pp. 91–93.
11. P. Leupold, C. Martín Vide and V. Mitrana: *Uniformly Bounded Duplication Languages.* In: Discrete Applied Mathematics 146 (3), 2005, pp. 301–310.
12. P. Leupold, V. Mitrana and J. Sempere: *Formal Languages Arising from Gene Repeated Duplication.* In: Aspects of Molecular Computing. Essays in Honour of Tom Head on his 70th Birthday. LNCS 2950, Springer Verlag, Berlin, 2004, pp. 297–308.
13. R. Ross and K. Winklmann: *Repetitive Strings are not Context-Free.* In: RAIRO Informatique Théorique, Vol 16, Nr.3, 1982, pp. 191–199.
14. M.-W. Wang: *On the Irregularity of the Duplication Closure.* Bull. EATCS 70, 2000, pp. 162–163.