RAIRO-Theor. Inf. Appl. **41** (2007) 411–424
DOI: 10.1051/ita:2007021

UNIFORMLY BOUNDED DUPLICATION CODES*

Peter Leupold[1] and Victor Mitrana[1, 2]

Abstract. Duplication is the replacement of a factor w within a word by ww. This operation can be used iteratively to generate languages starting from words or sets of words. By undoing duplications, one can eventually reach a square-free word, the original word's duplication root. The duplication root is unique, if the length of duplications is fixed. Based on these unique roots we define the concept of duplication code. Elementary properties are stated, then the conditions under which infinite duplication codes exist are fully characterized; the relevant parameters are the duplication length and alphabet size. Finally, some properties of the languages generated by duplication codes are investigated.

Mathematics Subject Classification. 68R15, 68Q45, 94B60.

1. Introduction

A fundamental concept concerning words is primitivity. A word is primitive, if it is not a non-trivial power of another word. For every non-primitive word, there is a unique primitive one the original word is a power of. This is often called the original word's primitive root. Thus, words having the same root can be generated by iterated catenation starting from the same word; in this sense they share a common primitive source.

Keywords and phrases. Duplication, duplication primitive word, duplication root, duplication code.

* *This work was done, while the first author was funded by the Spanish Ministry of Culture, Education and Sport under the Programa Nacional de Formación de Profesorado Universitario (FPU).*

[1] Research Group in Mathematical Linguistics, Rovira i Virgili University, Pça. Imperial Tàrraco 1, 43005 Tarragona, Catalunya, Spain;
klauspeter.leupold@urv.cat, victor.mitrana1@urv.cat

[2] Faculty of Mathematics and Computer Science, Bucharest University, Str. Academiei 14, 70109 Bucureşti, Romania.

Here we will take as underlying operation not catenation but duplication, that is replacement of factors w by ww within a given word. Analogously to the primitive root, we define the duplication root of a word. It is obtained by undoing all possible duplications within the word and thus reaching a word without any non-empty factor of the form ww. However, in contrast to the primitive root, the result of this reduction need not be unique in general; for example, the word *ababcbabc* can be reduced to *abcbabc via* un-duplication of $\underbrace{abab}cbabc$, and to *abc via* un-duplication of $a\underbrace{babcbabc}$ and then $\underbrace{abab}c$. Here, we investigate duplications of a fixed length n only, which makes the duplication root to be unique [6].

The primitive words under this notion are the ones not containing any repetition of a given length. Different variations of the duplication operation have received interest in the last years: the mentioned one of fixed length [6] as well as ones of bounded length and unrestricted length [2,5,11]. In all these investigations the main focus was on the generative power of the iterated duplication operation starting from a given word. A somewhat related topic is that of semigroups generated by the idempotency relation $w = ww$ as investigated in the book of Lothaire [7]; seen as an operation on words, this amounts to admitting besides duplications also their reversals.

As primitivity plays a central role in the investigation of codes [1, 10], it seems natural to see if some special type of code can be based on the notion of duplication. Here the uniqueness of the root is very desirable. This is why we restrict ourselves to only uniformly bounded duplications, meaning that all the duplicated factors are of a fixed, uniform length. We define such a notion of code also with an eye to the original motivation for investigating the duplication operation on words.

This operation was inspired by observing the behavior of DNA strands. There, through the formation of loops in space, parts of a strand can be duplicated in the place of the original strand. In an article from [4] which marks the latest milestone in the historic project of Human Genome Sequencing one claims that "5% of the human genome is involved in segmental duplications, and that the distribution of these regions varies widely across the chromosomes. Knowing the nature and extent of such duplications is important for understanding the evolution of the human genome, and for studying the many medically relevant disorders that are involved in segmental duplications, such as DiGeorge syndrome and Charcot-Marie-Tooth syndrome".

In an environment, where such duplications are possible, code words should satisfy the following property to adhere to the original idea of a code: even if arbitrary duplications within code words are performed, their catenation should allow a unique factorization into the original code words, possibly modified by duplications. This is exactly what our definition of an n-dup code guarantees.

We first provide the definition of uniformly bounded duplication and then compile some basic properties of the related duplication root; these will be used in the proofs of the later sections. Then we proceed to formalize the notion of duplication code as informally described above. Some elementary properties and first examples are provided. In Section 4 we fully characterize the conditions, under

which infinite duplication codes exist; the relevant parameters are the length of duplication and the size of the alphabet. In Section 5 we then state some properties of the languages generated by duplication codes, mainly about their density. In a final section we summarize the perspectives opened up by the work presented here.

2. The duplication operation and n-dup primitive words

We now provide the formal definitions concerning the duplication operation. For this, we take for granted elementary concepts from the theory of formal languages as exposed, for example, by Salomaa [9]. A few notations we use are: $|w|$ for the length of the word w, $w[i]$ for the i-th letter of w, and $w[i \dots j]$ for the factor of w starting at position i and ending at position j. A period of a word w is an integer k such that for all $i \leq |w| - k$ we have $w[i] = w[i + k]$. A word w is called *unbordered*, if it has no factorization uvu with a non-empty factor u. With this we come to the central notion of this article.

Let Σ be an alphabet; for a word $w \in \Sigma^+$ and a positive integer n we define the n-duplication set of w by

$$w^{1\heartsuit n} := \begin{cases} \{ruus \mid w = rus, r, s \in \Sigma^*, u \in \Sigma^+, |u| = n\}, & \text{if } |w| \geq n \\ \{w\}, & \text{otherwise.} \end{cases}$$

Here the heart nicely symbolizes the duplication operation: from one origin at the bottom it goes to two equal halves in the upper part. The number one signals that only one duplication is enacted. If the constant n is omitted, this means that the condition $|u| = n$ is dropped; if n is replaced by $\leq n$, then the condition is changed to $|u| \leq n$. We then speak of (general) and bounded duplication respectively. Now we define recursively the languages

$$w^{0\heartsuit n} := \{w\}, \qquad w^{i\heartsuit n} := \bigcup_{u \in w^{(i-1)\heartsuit n}} u^{1\heartsuit n}, \ i \geq 1, \qquad w^{\heartsuit n} := \bigcup_{i \geq 0} w^{i\heartsuit n}.$$

Thus $w^{\heartsuit n}$ is the language of all words that can be obtained from w by a finite number of duplications of length n. For example, $(abcabc)^{\heartsuit 3} = abc(abc)^+$ and $(aaa)^{\heartsuit 2} = a(aa)^+$. In what follows, we will generally use n without specifying its range; this shall mean that $n \geq 1$. All other cases will be explicitly indicated. Languages $w^{\heartsuit}$ and $w^{\heartsuit \leq n}$ are defined analogously.

In the canonical way, the duplication operation is extended to sets of words, setting for such a set W its language generated by duplication as

$$W^{\heartsuit n} := \bigcup_{w \in W} w^{\heartsuit n}.$$

One can also look at the effects of undoing duplications rather than duplicating factors. By this, one finally arrives at a word with no n-square, that is no factor

vv where $|v| = n$. No more duplications can be undone, and we call the resulting word the *n-duplication root* of the original word w. This root has been shown to be unique [6]. We denote the root by $\sqrt[\heartsuit_n]{w}$. In the sequel, we simply say root, when we speak about the n-duplication root and n is clear. First off, we note that

$$\sqrt[\heartsuit_n]{uv} = \sqrt[\heartsuit_n]{\sqrt[\heartsuit_n]{u} \cdot \sqrt[\heartsuit_n]{v}} = \sqrt[\heartsuit_n]{\sqrt[\heartsuit_n]{uv}} = \sqrt[\heartsuit_n]{u \sqrt[\heartsuit_n]{v}}$$

follows directly from the uniqueness of the root. The simpler equation $\sqrt[\heartsuit_n]{uv} = \sqrt[\heartsuit_n]{u} \cdot \sqrt[\heartsuit_n]{v}$ does not hold true in general. A trivial counterexample is $a = \sqrt[\heartsuit_1]{a \cdot a} \neq \sqrt[\heartsuit_1]{a} \cdot \sqrt[\heartsuit_1]{a} = aa$.

Every word which is its n-duplication root is called *n-dup primitive*. The language of all n-dup primitive words over Σ is denoted here by $\heartsuit_n(\Sigma)$. We recall from [6]:

Theorem 2.1. *For all positive n and all alphabets Σ, $\heartsuit_n(\Sigma)$ is regular.*

It is worth mentioning that the union of these languages for all $n \geq 1$ is known to be not even context-free, see, *e.g.*, [8], provided that the alphabet Σ has at least three letters.

We now compile a few elementary properties of n-dup primitive words and the corresponding roots. Some of these will prove useful in the later sections.

Proposition 2.2.

1. If for an n-dup primitive word u, $\sqrt[\heartsuit_n]{uu} = u$ holds, then also $\sqrt[\heartsuit_n]{u^+} = \{u\}$ holds.

2. If w is a word of length n, then $w^{\heartsuit_n} = w^+$.

Proof.

1. Note that $\sqrt[\heartsuit_n]{uuu} = \sqrt[\heartsuit_n]{u \sqrt[\heartsuit_n]{uu}} = \sqrt[\heartsuit_n]{uu} = u$. Inductively, $\sqrt[\heartsuit_n]{u^+} = \{u\}$ holds.

2. The inclusion $w^* \subseteq w^{\heartsuit_n}$ is obvious. We show the converse inclusion by induction on the number of duplications necessary to obtain a word $u \in w^{\heartsuit_n}$ starting from w. Clearly, by one duplication only ww can be obtained, and it is in w^*. Now suppose that $v \in w^*$ and $u \in v^{1\heartsuit_n}$. Then the factor $v[i \dots i+n-1]$ to be duplicated is a conjugate of w. Therefore there is a j such that $v[j+1 \dots i+n-1]v[i \dots j] = w$ and $v[1 \dots j], v[j+1 \dots |v|] \in w^*$. Now $v[i \dots i+n-1]^2 = v[i \dots j]wv[j+1 \dots i+n-1]$, and thus also $u \in w^*$. □

By a quite similar reasoning, we obtain another related result, which shows that duplications (and just as well unduplications) preserve periods, which divide their length.

Lemma 2.3. *If a word w has a period k, which divides n, then all words in $w^{\heartsuit_n}$ and $\sqrt[\heartsuit_n]{w}$ have period k, too.*

Proof. For $\sqrt[\heartsuit_n]{w}$ the statement is trivial, because removing factors of length n from a word with period k maintains this period. For $w^{\heartsuit_n}$ we prove the claim by induction. Of course, w has period k by assumption. Now, we suppose some

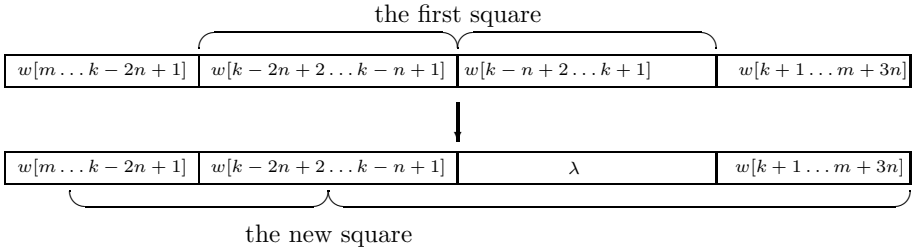

FIGURE 1. Undoing the first n-square starting at position $k - 2n + 1$.

word $u \in w^{\circlearrowleft n}$ has period k, and a factor u of length n is duplicated starting from position i. The resulting word is $w[1 \dots i + n - 1]uw[i + n \dots |w|] = w[1 \dots i - 1]uuw[i + n \dots |w|]$. Now $w[1 \dots i - 1]u$ and $uw[i + n \dots |w|]$ are a prefix and suffix of w, therefore have period n. Since at the point of catenation they agree on the n letters of u to both sides, also the catenation has period n. This, together with the fact that $w^{\circlearrowleft n} = \{w\}$ for words shorter than n, suffices to prove the claim. □

Now we turn our attention to cases, where a word and some of its powers have the same root. This is not always the case and thus some implications regarding the structure of the words satisfying this condition can be inferred.

Proposition 2.4. *If* $\sqrt[\circlearrowleft n]{ww} = \sqrt[\circlearrowleft n]{w}$ *for some word* w*, then* $|w|$ *is a multiple of* n*.*

Proof. $\sqrt[\circlearrowleft n]{ww} = \sqrt[\circlearrowleft n]{w}$ implies $\sqrt[\circlearrowleft n]{\sqrt[\circlearrowleft n]{w}\,\sqrt[\circlearrowleft n]{w}} = \sqrt[\circlearrowleft n]{w}$. This means that one can get from $\sqrt[\circlearrowleft n]{w}$ to $\sqrt[\circlearrowleft n]{w}\,\sqrt[\circlearrowleft n]{w}$ *via* duplications of length n. Because every such duplication increases the word's length by n, $|\sqrt[\circlearrowleft n]{w}|$ must be a multiple of n. As also w can be reached from $\sqrt[\circlearrowleft n]{w}$ *via* duplications of length n, its length must be a multiple of n, too. □

For general powers w^k with $k > 2$ (instead of ww) this statement is not true anymore; for example, whenever k is a multiple of n there are trivial counterexamples over the one-letter alphabet. Before we can make a more general statement, we prove an auxiliary lemma and recall the well-known periodicity lemma of Fine and Wilf.

Lemma 2.5. *If* $w[1 \dots k]$ *is an* n*-dup primitive prefix of* w*, then* $w[1 \dots k - n + 1]$ *is a prefix of* $\sqrt[\circlearrowleft n]{w}$*.*

Proof. If $w[1 \dots k]$ is an n-dup primitive prefix of w, then the first n-square in w can start at position $k - 2n + 2$. Suppose that unduplicating this square creates a new one starting at a position closer to the beginning, say m. Then the $2n$ letters $w[m \dots k - 2n + 1]w[k - 2n + 2 \dots k - n + 1]w[k + 2 \dots m + 3n]$ form this new n-square as can be seen in Figure 1.

This in turn implies that $w[m \dots k-2n+1]$ is a suffix of $w[k-2n+2 \dots k-n+1]$ and $w[k+2 \dots m+3n]$ is a prefix of it, in fact, because $w[k-2n+2 \dots k-n+1]$ has

length n and the square has length $2n$, we must have $w[k-2n+2\ldots k-n+1] = w[m\ldots k-2n+1]w[k+2\ldots m+3n] = w[k-n+2\ldots k+1]$. But this shows that $w[k+2\ldots m+3n]$ is also a suffix of $w[k-n+2\ldots k+1]$, and thus a square starting at position m was already present in the original word, which contradicts our assumption. □

This bound is tight as shown by the example of $\sqrt[\heartsuit 3]{babaaba} = baba$, where the longest 3-dup primitive prefix has length 6, and the root has length $4 = 6-3+1$. Of course, the same reasoning applies from the end of the word.

Corollary 2.6. *If $w[k\ldots|w|]$ is a square-free suffix of w, then $w[k+n-1\ldots|w|]$ is a suffix of $\sqrt[\heartsuit n]{w}$.*

In what follows we will use a result originally due to Fine and Wilf; the formulation for words can for example be found in the book by Lothaire.

Lemma 2.7. [3,7]. *If a word w has periods k and l, then also $k+l-gcd(k,l)$ is a period of w.*

Now we are ready to make a statement about the case where general powers of a word have the same root as the word itself. This result will be used later in the proof of Proposition 5.2.

Lemma 2.8. *If $\sqrt[\heartsuit n]{w^k} = \sqrt[\heartsuit n]{w}$ for some word w and some integer $k \geq 2$, then n has a period of w as divisor.*

Proof. First notice that due to the uniqueness of the root, one can undo first all duplication within the different factors w of w^k. By Lemma 2.3 this would not change the fact that a period of w divides n. Thus, without restriction of generality we can suppose that w is n-dup primitive. For words shorter than n, $\sqrt[\heartsuit n]{w^k} = \sqrt[\heartsuit n]{w}$ can never hold; for $|w| = n$, obviously always $\sqrt[\heartsuit n]{w^k} = \sqrt[\heartsuit n]{w}$ and also n trivially is a period of w. Therefore we can suppose $|w| > n$ in the following.

Because w is n-dup primitive any duplication to be undone in w^k must cross a border in between two of the factors w. Further, we suppose that the first n-square in w^k involves at most the last n letters of the first factor w. This means that the entire word w remains unchanged by unduplicating this square, and thus by Lemma 2.5 it remains unchanged in the whole process of arriving at $\sqrt[\heartsuit n]{w}$. If the first square starts earlier, the same reasoning will work from the end of the word, and the last factor w will remain unchanged.

The only case, where neither is true is the occurrence of a third power uuu such that the central u includes the border between the ws. If w has length at least $2n$, then these blocks do not overlap each other, we can just delete the initial n letters of each w and proceed with the resulting word; this preserves squares uu and also preserves any period not longer than n. For shorter w, since $|w| > n$ also $k > 2$, and the factors uuu overlap. This implies that the entire word w has period $|u|$. Since $|u| = n$, by Lemma 2.3 the initial claim is proved in this case.

Summarizing the reasoning to this point, we can now assume that w is n-dup primitive, $|w| > n$, and that the first n-square in w^k involves at most the last n letters of the first factor w. This means that we can cancel the last copy of each

occurrence of this n-square within each of the second to k-th factor w, and arrive at a new word $w(w')^{k-1}$. Since we started at the left-most n-square, according to Lemma 2.5 this process can be continued, until we arrive at a word of this form, where w' is shorter than n – under the assumption $\sqrt[\heartsuit n]{w^k} = \sqrt[\heartsuit n]{w}$ this has to be possible, because we must be able to arrive at a word of length only $|w|$. If the length of w' is a divisor of n, then $(w')^{k-1}$ has a period dividing n, and by Lemma 2.3 this holds also for w^{k-1}, which proves our initial claim.

If the length of w' is not a divisor of n, there still must be an n-square in $(w')^{k-1}$. Since it has period n and also period $|w'| < n$, by Lemma 2.7 w' has period $gcd(n, |w'|)$, which by definition divides n, and again our initial claim follows with Lemma 2.3. □

3. n-DUP CODES

We now proceed to define the central notion of this article, the *n-duplication code*, or shortly n-dup code. It is closely oriented after the definition of a conventional code, only instead of the catenation of words we consider the catenation of their n-duplication sets. Recall that a set of words W is a conventional code, if for two integers k, l and words $u_0, \ldots, u_k, v_0, \ldots, v_l \in W$ the equation

$$u_0 u_1 \ldots u_k = v_0 v_1 \ldots v_l$$

implies that $k = l$ and all $u_i = v_i$ hold for $0 \leq i \leq k$. Analogously, in the sense described above, we say that W is an *n-dup code*, if

$$u_0^{\heartsuit n} u_1^{\heartsuit n} \ldots u_k^{\heartsuit n} \cap v_0^{\heartsuit n} v_1^{\heartsuit n} \ldots v_l^{\heartsuit n} \neq \emptyset$$

implies that $k = l$ and $u_i = v_i$ for $0 \leq i \leq k$. From the definition it is clear that every n-dup code is also a code in the conventional sense, because always $w \in w^{\heartsuit n}$. The converse is trivially not true. According to the definition, the sequence of words $u_0 u_1 \ldots u_k$ such that $w \in u_0^{\heartsuit n} u_1^{\heartsuit n} \ldots u_k^{\heartsuit n}$ must be unique for any word w; this, however, still might admit some ambiguity as to the actual factorization of w. Different combinations of words from the sets $u_i^{\heartsuit n}$ might provide factorizations of w. We would like to point out here a few more considerations. Let $W = \{abb, aab\}$; clearly, $W^{\heartsuit 1} = a^+ b^+$ is a code, but W is not an 1-dup code. On the other hand, $W = \{ab, bc\}$ is a 1-dup code while $W^{\heartsuit 1} = a^+ b^+ \cup b^+ c^+$ is not a code.

Example 3.1. As a first example of an n-dup code, $n \geq 2$, we look at the set $\{aba\}$, which is a 2-duplication code. Clearly $(aba)^{\heartsuit 2} = a(ba)^*$. Thus any catenation of words from $(aba)^{\heartsuit 2}$ has two consecutive a exactly at the borders between the catenated words; this provides the unique factorization of these catenations.

We now compile some simple properties of words contained in n-dup codes. The first one is obvious, because for words shorter than n the duplication language generated contains only the original words itself.

Proposition 3.2. *A set of words all shorter than n is an n-dup code if and only if it is a code.*

Further, for every word w of length n, we have $w \cdot w \in w^{\heartsuit_n}$. Therefore containment of w in an n-dup code would result in two distinct factorizations of ww. Thus we can state two conditions that necessarily makes a set of words not an n-dup code.

Proposition 3.3.
1. *An n-dup code cannot contain any word of length n.*
2. *An n-dup code cannot contain two words with the same n-duplication root.*

Proof. The second statement follows immediately from the next result that appears in [6]:

Lemma 3.4. [6]. *Let $k \geq 1$, if $u, v \in w^{\heartsuit_n}$ for some w, then $u^{\heartsuit_n} \cap v^{\heartsuit_n} \neq \emptyset$.* □

Therefore, the only words really interesting which belong to an n-dup code are the ones of length greater than n with at least two letters. Having stated several conditions for a set of words not to be a code, we now state a property that makes a set of two words an n-dup code in a non-trivial way.

Proposition 3.5. *If uu, vv, uv and vu are n-dup primitive words longer than n, then $\{u, v\}$ is an n-dup code.*

Proof. If uu, vv, uv and vu are all n-dup primitive and longer than n, then all words in $\{u, v\}^*$ are n-dup primitive. Thus every such word $w_0 w_1 \dots w_n$ is the unique root of any word in $w_0^{\heartsuit_n} w_1^{\heartsuit_n} \dots w_n^{\heartsuit_n}$, if all w_i are from the set $\{u, v\}$. Suppose now that some word has two such factorizations into words from $\{u, v\}^{\heartsuit_n}$. If they are distinct, then they result in two distinct duplication roots as just exposed. This is in contradiction to the uniqueness of the root. Therefore no word can have two distinct factorizations of this type, and $\{u, v\}$ is an n-dup code. □

Of course, the argumentation from the proof of Proposition 3.5 can be generalized to any number of words. Hence, we state:

Corollary 3.6. *Let W be a set of words all longer than n such that all words in W^2 are n-dup primitive. Then W is an n-dup code.*

4. Infinite n-dup codes

Of course, there are infinite conventional codes, however, it is not self-evident that also infinite duplication codes exist. As we will see, this depends on the size of the alphabet and on the length of the duplications. We start with a negative result, i.e. with a case where no infinite dup code exists.

Proposition 4.1. *There is no infinite* 1*-dup code over a two-letter alphabet.*

Proof. Let W be a 1-dup code over the alphabet $\{a, b\}$. Suppose that W contains a word w that starts with a and ends with b. If there is another word u from W with the same properties, then let k be the number of changes from a to b; that is reading u from left to right, we read k non-empty blocks of consecutive a, which are followed by non-empty blocks of consecutive b. Let l be the corresponding number for w.

We now start from the word $(ab)^{(k \cdot l)}$ and duplicate the initial a so often, that the initial block of a is as long as the longer one from u^l and w^k. Then the same is done for the first block of b and so on for all blocks. Clearly the resulting word is in both $(w^{\triangledown 1})^k$ and $(u^{\triangledown 1})^l$. Thus W is a code only if $u = w$, and any 1-dup code can contain at most one word starting with a and ending with b.

For words starting with b and ending with a the argumentation is the same, for words starting and ending with the same letter (a or b), a very similar line of thought works. As there are only four possibilities of different first/last letter combinations, and for every one at most one word can be in W, no 1-dup code can be infinite. □

From the proof we immediately see an even tighter bound for the size of a 1-dup code. Namely that over a two-letter alphabet there is no 1-dup code consisting of more than four words, because there are only four possible combinations of first and last letter. This, however, is not yet optimal. In fact, the maximum number of words in a 1-dup code is only one – considerations only slightly more intricate than above show this.

Proposition 4.2. *Over a two-letter alphabet there is no* 1*-dup code consisting of more than one word.*

Proof. An argumentation analogous to that of the proof of Proposition 4.1 works for any two words having a change of letter inside. Only, if they do not start and end with the same letters the construction gets slightly more intricate, some "padding" at the start and end may be necessary. □

The situation changes, when we increase the size of the alphabet. Already three letters suffice to construct an infinite code.

Proposition 4.3. *There exist infinite* 1*-dup codes over a three-letter alphabet.*

Proof. We prove this by providing an example for such a code. The language $W = (ab)^+c$ is an infinite 1-dup code. First off, we note the fact that the duplication of a single letter can never change the number of letter-changes in a given word. From this, a 1-dup code factorization for every word w from $W^{\triangledown n}$ can be found by splitting it after every block of c. Further, the number of changes from a to b uniquely determines the word from W, from which the respective factor originated. □

Proposition 4.4. *There exist infinite* n*-dup codes over a two-letter alphabet for any* $n \geq 2$ *.*

Proof. The language $W = a(abb)^+$ is a 2-dup code. To see this consider the effects of possible 2-duplications on a word from W: $aa \to aaaa$, $ab \to abab$, and $bb \to bbbb$. All of them preserve the number of blocks of the same letter of length greater than one in the original word – for this one needs to look also at the letters immediately preceding and following the duplicated factor.

Because in W^+ all W-factors of a word start with aa and this is the only occurrence of aa, the W-factorization is unique. Further, for every positive integer the word from W having this number of bb-blocks unique. Thus it is easy to reconstruct from any word in $W^{[\heartsuit 2]}$ its unique 2-dup factorization by separating the word at the beginning of every (maximal) block of a, which is longer than one.

With slight modifications we can now construct analogous examples for all $n \geq 3$: the language $W = a(a^{n-1}b^{n-1})^+b$ is an n-dup code. We first note that any n-duplication applied to a word in W can produce neither a factor a^n to the right of any occurrence of b nor a factor b^n to the left of any occurrence of a. Therefore, all $W^{\heartsuit n}$-factors of a word in $(W^{\heartsuit n})^+$ start with a^n and end with b^n, hence the $W^{\heartsuit n}$-factorization of every word in $(W^{\heartsuit n})^+$ is unique. Checking the possible results of all duplications of length n, we see that the number of blocks of $a^{n-1}b^{n-1}$ is preserved within each code word. Therefore they are uniquely identifiable in the factorization, which proves the property of being a duplication code. □

Summarizing the results of this section and adding a few trivial considerations for one-letter alphabets, we obtain the following theorem, which fully characterizes the conditions under which infinite duplication codes exist.

Theorem 4.5. *There exist infinite n-dup codes over a k-letter alphabet, if and only if $k, n \geq 2$ or if $n = 1$ and $k \geq 3$.*

All the examples for infinite n-dup codes provided in this section have been regular, however there are arbitrarily complex n-dup codes with respect to the Chomsky hierarchy, over a two letter alphabet, for any $n \geq 2$. It suffices to take arbitrarily complex subsets of the n-dup codes W from the proof of Proposition 4.4.

5. Languages generated by n-dup codes

An interesting concept in relation with codes is the density of the languages they generate. Informally speaking, density means that any word appears as a factor of some word in the generated language. Formally, a language $L \subset \Sigma^*$ is called *dense*, if for every word $w \in \Sigma^*$ we have $\Sigma^* w \Sigma^* \cap L \neq \emptyset$.

The constructions used to prove that the languages generated from one word by general duplication [11] and by (non-uniformly) bounded duplication [6] show that in most cases those languages are also dense; for example the occurrence of a factor abc suffices to guarantee this over the corresponding three-letter alphabet. For uniformly bounded duplications, however, these construction techniques cannot be applied.

Proposition 5.1. *There exists an infinite n-dup code W, such that the language generated by W, $(W^{\heartsuit_n})^*$, is not dense for all $n \geq 1$.*

Proof. The language $W = \{d\}T_3\{d\} \subseteq \{a, b, c, d\}^*$ is an infinite n-dup code, where T_3 is the infinite set of n-dup primitive words over $\{a, b, c\}$. Following the argumentation showing that this is so, we also see that words from $(W^{\heartsuit_n})^*$ do not contain any factor *caac*. Thus $(W^{\heartsuit_n})^*$ is not dense. □

On the other hand, density of W, or even of W^* guarantees the density of $(W^{\heartsuit_n})^*$. These observations raise the question, whether there is an n-dup code W, such that W^* is not dense, but its generated language $(W^{\heartsuit_n})^*$ is dense. If we require only W not to be dense, then there are trivial solutions like Σ itself, which is an n-dup code for any $n > 1$ and generates entire Σ^*.

The most prominent result concerning conventional codes in this respect is that density is given if and only if a code is maximal [1]. We now present two somewhat contrasting results, the first showing that there are always infinitely many n-dup primitive words not in the root of the language; then we will see that this still allows the languages generated to be dense.

Proposition 5.2. *For every n-dup code W over the alphabet Σ, the set $\heartsuit_n(\Sigma) \setminus \sqrt[\heartsuit_n]{(W^{\heartsuit_n})^*}$ is infinite.*

Proof. First we notice that $\heartsuit_n(\Sigma)$ is always infinite. Thus, if $\sqrt[\heartsuit_n]{(W^{\heartsuit_n})^*}$ is not infinite, the proposition is true. In the contrary case, $\sqrt[\heartsuit_n]{(W^{\heartsuit_n})^*}$ contains an infinite set U, which consists of words longer than $2n + 2$. For such a word u we now look at the words $v = u[1 \dots \left\lfloor \frac{|u|}{2} \right\rfloor]$ and $w = u[\left\lfloor \frac{|u|}{2} \right\rfloor + 1 \dots |u|]$, which are both n-dup primitive, just as u.

If there existed words $v_1, w_1 \in (W^{\heartsuit_n})^*$ such that $\sqrt[\heartsuit_n]{v_1} = v$ and $\sqrt[\heartsuit_n]{w_1} = w$, then also $v_1 w_1$ would be in $(W^{\heartsuit_n})^*$. Then by Lemma 3.4 there would also exist words $v_2, w_2 \in \Sigma^*$ such that $v_2 \in v^{\heartsuit_n} \cap v_1^{\heartsuit_n}$ and $w_2 \in w^{\heartsuit_n} \cap w_1^{\heartsuit_n}$. But this implies that $v^{\heartsuit_n} w^{\heartsuit_n} \cap u^{\heartsuit_n} \neq \emptyset$. Therefore for at least one of v and w no word can be in W that has this root, otherwise W would not be an n-dup code. Neither can this word be composed by shorter ones, the same argumentation would apply. This provides us with one word in the set $\heartsuit_n(\Sigma) \setminus \sqrt[\heartsuit_n]{(W^{\heartsuit_n})^*}$.

Thus it remains to construct an infinite sequence of such words providing us with pairwise different words from $\heartsuit_n(\Sigma) \setminus \sqrt[\heartsuit_n]{(W^{\heartsuit_n})^*}$. For an infinite n-dup code W, already $\sqrt[\heartsuit_n]{W}$ is infinite by Proposition 3.3. Thus we can find an infinite sequence $(u_i)_{i \in \mathbb{N}}$ of words in $\sqrt[\heartsuit_n]{W}$ such that always $|u_i| > |u_{i-1}| + 2$, which satisfies the requirements stated.

For a finite set W, we pick a word $w \in W$, which has no period that divides n. Then by Lemma 2.8 the sequence of $u_i := w^{2n}$ works. If all words in W have periods dividing n, then we take $u_i := (vv)^{2n}$ for such a word $v \in W$. Now, if vv still had a period dividing n, then v^{n+1} could be reduced *via* n-unduplications to v, and consequently v cannot be in an n-dup code. Therefore vv has no period dividing n, and can be used just as w above. □

Proposition 5.3. *Over an alphabet Σ with three or more letters, there exists an infinite 1-dup code W, such that $(W^{\heartsuit 1})^*$ is dense.*

Proof. Recall that $\heartsuit_1(\Sigma)$ is the language of all 1-dup primitive words. Now we choose an arbitrary non-empty, unbordered word w from $\heartsuit_1(\Sigma)$ with $w[1] = b$, $w[|w|] = a$ and $|w| > 1$. We set $\heartsuit'_1(\Sigma) := \heartsuit_1(\Sigma) \setminus (\Sigma^* w \Sigma^* \cup a\Sigma^* b)$. Note that $\heartsuit'_1(\Sigma)$ is infinite over an alphabet Σ with three or more letters.

Then $W := \heartsuit'_1(\Sigma) \cdot w$ is a 1-dup code, because $\heartsuit'_1(\Sigma)^{\heartsuit 1}$ and $w^{\heartsuit 1}$ are disjoint. Thus any word from $(W^{\heartsuit 1})^*$ is uniquely factorized into words from $W^{\heartsuit 1}$ by separating them after any occurrence of a factor from $w^{\heartsuit 1}$. Note that different occurrences of w cannot overlap, because the word is unbordered.

It remains to show that $(W^{\heartsuit 1})^*$ is dense. Intuitively speaking, $\heartsuit_1(\Sigma)$ is the language one obtains from Σ^* by condensing all blocks of the same letter within a word to length one. From these words any other word can be obtained by doing the appropriate 1-duplications. Therefore it suffices to show that for all $u \in \heartsuit_1(\Sigma)$ we have $\Sigma^* u \Sigma^* \cap W^* \neq \emptyset$. For words u not containing a factor w this is true, because they are already contained in W.

We first look at words not starting with a. For such a word u not containing one factor w, there is a factorization $u = u_1 w u_2$. Here it is crucial to note that $u_2[1] \neq a$, otherwise u would contain a 1-square, the same for $u_1[|u_1|] \neq b$. But now we have $u_1 w, u_2 w \in W$, and thus $u_1 w, u_2 w \in W^*$ with a factor u. For words with more occurrences of factors w analogous factorizations can be found.

Thus, all 1-dup primitive words not starting with a are prefixes of words in W^*. With the observation that words av are factors of the corresponding wvw we conclude the proof. □

Another interesting questions is, whether the step from W to $(W^{\heartsuit n})^*$ increases the complexity of the language with respect to the Chomsky Hierarchy. For regular languages no increase in complexity can be observed.

Theorem 5.4. *The language $W^{\heartsuit n}$ is regular for every regular language W and $n \geq 1$.*

Proof. Let $W \subseteq \Sigma^*$ be a regular language and let $W = W_1 \cup W_2$, where $W_1 = \{x \in W \mid |x| \leq n\}$, and $W_2 = W \setminus W_1$. Obviously, $W^{\heartsuit n} = W_2^{\heartsuit n} \cup W_1$. Assume that W_2 is recognized by the DFA (deterministic finite automaton) $A = (Q, \Sigma, \delta, q_0, F)$. We construct the DFA

$$A' = (Q', \Sigma, \delta', \langle q_0, \lambda \rangle, F'),$$

where

$$\begin{aligned} Q' &= \{\langle q, x \rangle \mid q \in Q, x \in \Sigma^*, |x| \leq n\} \\ F' &= \{\langle q, x \rangle \mid q \in F, x \in \Sigma^*, |x| = n\} \end{aligned}$$

and the transition mapping δ' is defined as follows:

$$\delta'(\langle q, x \rangle, a) = \begin{cases} \langle \delta(q, a), xa \rangle, & \text{if } |x| < n \\ \langle \delta(q, a), Suf_n(xa) \rangle, & \text{if } |x| = n. \end{cases}$$

Here $Suf_n(z)$ denotes the suffix of z of length n. Clearly, the automaton A' recognizes the same language as A does, namely W_2.

We now recall a result from [6], which is very useful for the last part of our proof. For two words x, y over an alphabet Σ such that $y \in x^{\heartsuit n}$ and p a positive integer, we write $x \bowtie_{(p,k)} y$ if $x = tuv$, $|t| = p - 1$, $|u| = k$ and $y = tuuv$. The following result appears in [6]:

Proposition 5.5. *If* $x = x_1 \bowtie_{(p_1,k)} x_2 \bowtie_{(p_2,k)} x_3 \bowtie_{(p_3,k)} \dots x_r \bowtie_{(p_r,k)} w$ *for some* p_i, $1 \leq i \leq r$, *then* $x = y_1 \bowtie_{(q_1,k)} y_2 \bowtie_{(q_2,k)} y_3 \bowtie_{(q_3,k)} \dots y_r \bowtie_{(q_r,k)} w$ *holds for some* $q_1 \leq q_2 \leq \dots \leq q_r$. *Furthermore, for each* $i \in [r-1]$, *either* $q_i = q_{i+1}$ *or* $q_{i+1} - q_i > k$ *holds.*

By this, if one adds a loop labeled by x to any state $\langle q, x\rangle \in Q'$ with $|x| = n$, one gets an automaton (not necessarily deterministic) which accepts the language $W_2^{\heartsuit n}$ and we are done. □

Since the family of regular language is closed under Kleene closure, we obtain a statement about the languages generated by regular n-dup codes.

Corollary 5.6. *The language generated by a regular* n*-dup code* W, $n \geq 1$, *is still regular.*

6. Perspectives

We have seen in Section 3 that in the case of uniformly bounded duplication infinite duplication codes exist in most cases. From earlier research it is known that only an upper bound on the length of duplicated factors results in much more complicated structures. Thus $w^{\heartsuit n}$ is always regular, while $w^{\heartsuit \leq n}$ is in general not. This leads us to suspect that for bounded or even general duplication infinite codes complying with our definition will be much harder to find and should constitute an interesting problem. The questions of Section 5, however, would become obsolete, because for $n \geq 4$ already the languages $w^{\heartsuit \leq n}$ for a single word w are dense.

References

[1] J. Berstel and D. Perrin, *Theory of Codes*. Academic Press, Orlando (1985).
[2] J. Dassow, V. Mitrana and Gh. Păun, On the Regularity of Duplication Closure. *Bull. EATCS* **69** (1999) 133–136.
[3] N. Fine and H. Wilf, Uniqueness Theorems for Periodic Functions. *Proc. Amer. Math. Soc.* **16** (1965) 109–114.
[4] International Human Genome Sequencing Consortium, Finishing the Euchromatic Sequence of the Human Genome. *Nature* **431** (2004) 931–945.
[5] P. Leupold, V. Mitrana and J. Sempere, *Languages Arising from Gene Repeated Duplication*, in *Aspects of Molecular Computing*. Essays Dedicated to Tom Head on the Occasion of his 70th Birthday. *Lect. Notes Comput. Sci.* **2950** (2004) 297–308.
[6] P. Leupold, C. Martín Vide and V. Mitrana, Uniformly Bounded Duplication Languages. *Discrete Appl. Math.* **146** (2005) 301–310.

[7] M. Lothaire, *Combinatorics on Words*. Addison-Wesley, Reading, MA (1983).
[8] R. Ross and K. Winklmann, Repetitive Strings are not Context-Free. *RAIRO-Theor. Inf. Appl.* **16** (1982) 191–199.
[9] A. Salomaa, *Formal Languages*. Academic Press, Orlando (1973).
[10] H.J. Shyr, *Free Monoids and Languages*. Hon Min Book Company, Taichung (1991).
[11] M.-W. Wang, On the Irregularity of the Duplication Closure. *Bull. EATCS* **70** (2000) 162–163.

Communicated by H.J. Hoogeboom.
Received September 8, 2005. Accepted November 14, 2006.