On Infinite Words Determined by L Systems☆

Tim Smith

Northeastern University
Boston, MA, USA

Abstract

A deterministic L system generates an infinite word α if each word in its derivation sequence is a prefix of the next, yielding α as a limit. We generalize this notion to arbitrary L systems via the concept of prefix languages. A prefix language is a language L such that for all $x, y \in L$, x is a prefix of y or y is a prefix of x. Every infinite prefix language determines a single infinite word. Where C is a class of L systems (e.g. 0L, DT0L), we denote by $\omega(C)$ the class of infinite words determined by the prefix languages in C. This allows us to speak of infinite 0L words, infinite DT0L words, etc. We categorize the infinite words determined by a variety of L systems, showing that the whole hierarchy collapses to just three distinct classes of infinite words: ω(PD0L), ω(D0L), and ω(CD0L). Our results are obtained with the help of a pumping lemma which we prove for T0L systems.

Keywords: infinite word, L system, prefix language, pumping lemma, morphic word, D0L

1. Introduction

L systems are parallel rewriting systems which were originally introduced to model growth in simple multicellular organisms. With applications in biological modelling, fractal generation, and artificial life, L systems have given rise to a rich body of research [14, 11]. L systems can be restricted and generalized in various ways, yielding a hierarchy of language classes.

The simplest L systems are D0L systems (deterministic Lindenmayer systems with 0 symbols of context), in which a morphism is successively applied to a start string or "axiom". The resulting sequence of words comprises the language of the system. If the morphism is prolongable on the start string, then each word in the derivation sequence will be a prefix of the next, yielding an infinite word as a limit. An infinite word obtained in this way is called an infinite D0L word.

Two well-studied generalizations of D0L systems are 0L systems, which introduce nondeterminism by changing the morphism to a finite substitution, and DT0L systems, in which the morphism is replaced by a set of morphisms or "tables". In each case, there is no longer just one possible derivation sequence; rather, there are many possible derivations, depending on which letter substitutions or tables are chosen at each step. This raises the question of under what conditions such a system can be said to determine an infinite word.

We answer this question with the concept of a prefix language. A prefix language is a language L such that for all $x, y \in L$, x is a prefix of y or y is a prefix of x. Every infinite prefix language determines a single infinite word. Where C is a class of L systems (e.g. 0L, DT0L), we denote by $\omega(C)$ the class of infinite words determined by the prefix languages in C. This allows us to speak of infinite 0L words, infinite DT0L words, etc.

With this notion in place, we categorize the infinite words determined by a variety of L systems. We consider four production features (D,P,F,T) and five filtering features (E,C,N,W,H). Each production feature

☆This journal paper comprises the full version of the author's conference paper [16] and also includes material from the author's conference paper [15].

Email address: `smithtim@ccs.neu.edu` (Tim Smith)

may be present or absent, and at most one filtering feature may be present, giving a total of $2^4 \cdot 6 = 96$ classes of L systems. We show that this whole hierarchy collapses to just three classes of infinite words: ω(PD0L), ω(D0L), and ω(CD0L). Our results appear in Figure 1.

The inclusions among these three classes are proper, giving $\omega(\text{PD0L}) \subset \omega(\text{D0L}) \subset \omega(\text{CD0L})$. The class ω(CD0L) contains exactly the morphic words, while ω(D0L) properly contains the pure morphic words.

1.1. Proof techniques

We obtain our categorization results by showing that all infinite languages in certain classes of L systems have infinite subsets in certain smaller classes of L systems. This limits the infinite words of the larger class to the infinite words of the smaller class. First, we provide a necessary and sufficient condition under which a T0L system is infinite, in the form of a pumping lemma. In getting this result, we adapt a proof technique used in [13] to obtain a pumping lemma for ET0L systems. It follows from our pumping lemma that every infinite T0L language has an infinite D0L subset. With this result, we show that every infinite ET0L language has an infinite CD0L subset, and we make further use of the pumping lemma to show that every infinite PT0L language has an infinite PD0L subset. A separate argument shows that every infinite ED0L (EPD0L) language has an infinite D0L (PD0L) subset.

1.2. Related work

Prefix languages were investigated by Book [3], who formulated a "prefix property" intended to allow languages to "approximate" infinite sequences, and showed that for certain classes of languages, if a language in the class has the prefix property, then it is regular. Languages whose complement is a prefix language, called "coprefix languages", have also been studied [2]. The present journal paper comprises the full version of the author's conference paper [16] and also includes material from the author's conference paper [15], in particular the pumping lemma for T0L systems. In Smith [17], prefix languages are used to characterize the infinite words determined by several classes of one-way stack automata, and also studied in connection with multihead deterministic finite automata. In Smith [18], prefix languages are used to characterize the infinite words determined by the indexed languages of Alfred Aho.

The iterative processes underlying L systems have been investigated in connection with infinite words. Pansiot [12] considered various classes of infinite words obtained by iterated mappings. Culik & Karhumäki [5] considered iterative devices generating infinite words. Culik & Salomaa [6] investigated infinite words associated with D0L and DT0L systems, studying two ways of associating a language L with a set of infinite words: the adherence of L and the limit of L. Their notion of "strong uniform convergence" is equivalent to our notion of a language "determining" an infinite word.

Our results on infinite subsets can be restated in the framework of set immunity [19]. For a language class C, a language L is C-immune iff L is infinite and no infinite subset of L is in C. For example, our result that every infinite ET0L language has an infinite CD0L subset could be stated: no ET0L language is CD0L-immune. In addition to categorizing the infinite words determined by L systems, our results characterize the immunity relationships among these systems.

Corollary 2, which states our pumping lemma for DT0L systems, can also be proved via a connection with non-negative integer matrices. Each table in a DT0L system can be associated with a "growth matrix" indicating for each production, how many times each symbol appears on the righthand side of that production. Jungers et al. [10] consider the "joint spectral radius" ρ of a finite set of such matrices, distinguishing four cases. In cases (1) and (2) ($\rho = 0$ or $\rho = 1$ with bounded products), the associated DT0L system is finite, whereas in cases (3) and (4) ($\rho > 1$ or $\rho = 1$ with unbounded products), by their Corollary 1 and Proposition 2, assuming every symbol is reachable, the system is pumpable.

A recent result similar to our Corollary 2 appears in [9] in connection with monoids of morphisms.

1.3. Outline of paper

The paper is organized as follows. Section 2 gives preliminary definitions and propositions. Section 3 presents our pumping lemma for T0L systems. Section 4 gives results on infinite subsets of certain classes of L systems. Section 5 categorizes the infinite words determined by the hierarchy of L systems. Section 6 separates and characterizes the classes ω(PD0L), ω(D0L), and ω(CD0L). Section 7 gives our conclusions.

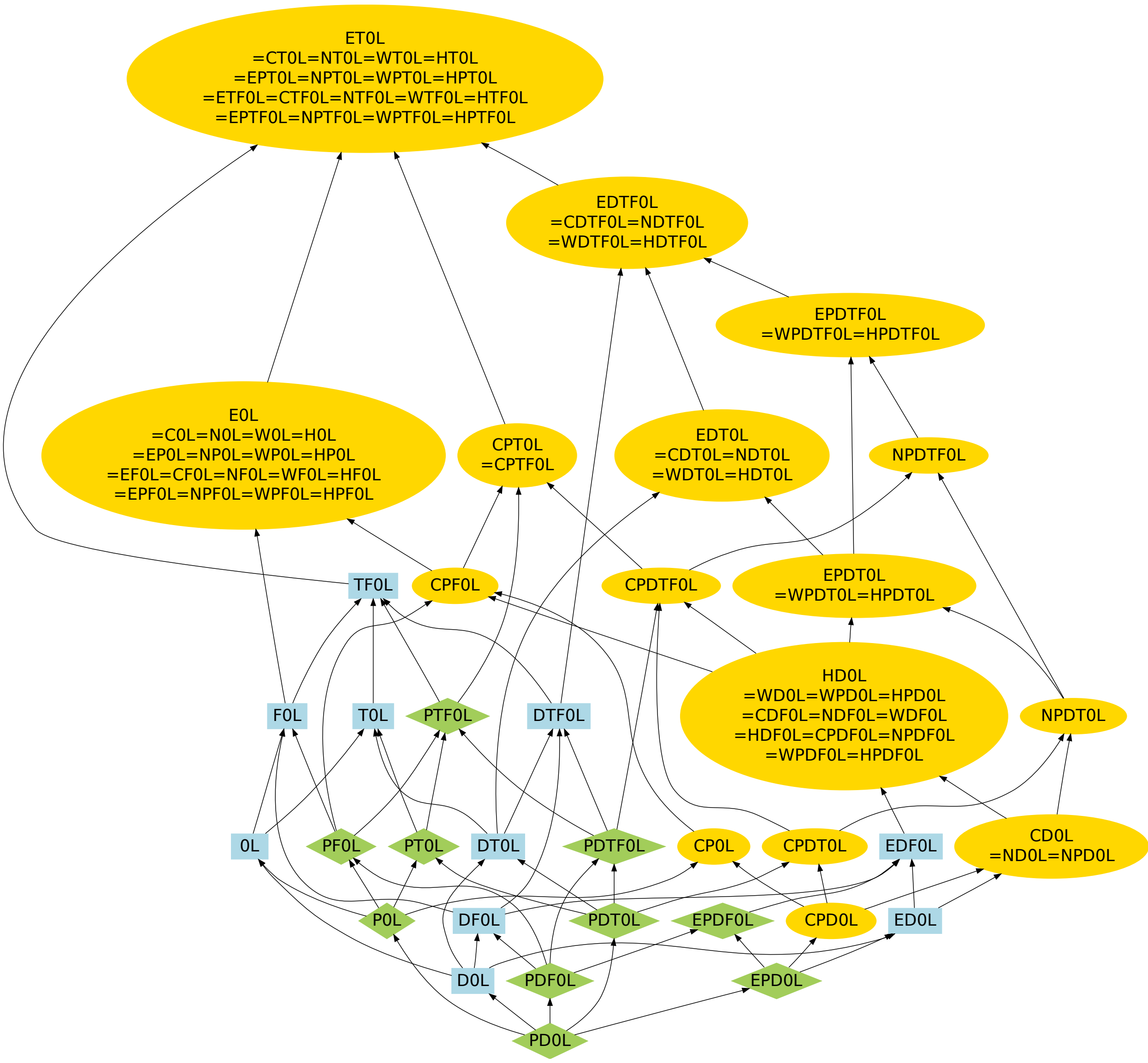

Figure 1: Inclusion diagram showing classes of L systems colored by the infinite words they determine. Green classes (diamonds) determine exactly ω(PD0L), blue classes (rectangles) determine exactly ω(D0L), and yellow classes (ellipses) determine exactly ω(CD0L). Inclusions and equalities are from [11].

2. Preliminaries

Where X and Y are sets, $X \subseteq Y$ means that X is a subset of Y and $X \subset Y$ means that X is a proper subset of Y. We denote the cardinality of X by $|X|$.

An **alphabet** A is a finite set of symbols. A **word** is a concatenation of symbols from A. We denote the set of finite words by A^* and the set of infinite words by A^ω. A **string** x is an element of A^*. The length of x is denoted by $|x|$. We denote the empty string by λ. A **language** is a subset of A^*. A (symbolic) **sequence** S is an element of $A^* \cup A^\omega$. A **prefix** of S is a string x such that $S = xS'$ for some sequence S'. A **subword** (or factor) of S is a string x such that $S = wxS'$ for some string w and sequence S'. For a nonempty string x, x^ω denotes the infinite word $xxx\cdots$. Such a word is called **purely periodic**. An infinite word of the form xy^ω, where x and y are strings and $y \neq \lambda$, is called **ultimately periodic**.

A **morphism** on an alphabet A is a map h from A^* to A^* such that for all $x, y \in A^*$, $h(xy) = h(x)h(y)$. Notice that $h(\lambda) = \lambda$. The morphism h is **nonerasing** if for all $a \in A$, $h(a) \neq \lambda$. The morphism h is a **coding** if for all $a \in A$, $|h(a)| = 1$. The morphism h is a **weak coding** if for all $a \in A$, $|h(a)| \leq 1$. The morphism h is an **identity** if for all $a \in A$, $h(a) = a$. For a language L, we define $h(L) = \{h(x) \mid x \in L\}$. A string $x \in A^*$ is **mortal** (for h) if there is an $m \geq 0$ such that $h^m(x) = \lambda$. The morphism h is **prolongable** on a symbol a if $h(a) = ax$ for some $x \in A^*$, and x is not mortal. In this case $h^\omega(a)$ denotes the infinite word $a\ x\ h(x)\ h^2(x)\ \cdots$. An infinite word α is **pure morphic** if there is a morphism h and symbol a such that h is prolongable on a and $\alpha = h^\omega(a)$. An infinite word α is **morphic** if there is a morphism h, coding e, and symbol a such that h is prolongable on a and $\alpha = e(h^\omega(a))$. Every purely periodic word is pure morphic, and every ultimately periodic word is morphic. For results on morphic words, see [1].

A **finite substitution** on A is a map σ from A^* to 2^{A^*} (the power set of A^*) such that (1) for all $x \in A^*$, $\sigma(x)$ is finite and nonempty, and (2) for all $x, y \in A^*$, $\sigma(xy) = \{x'y' \mid x' \text{ is in } \sigma(x) \text{ and } y' \text{ is in } \sigma(y)\}$. Notice that $\sigma(\lambda) = \{\lambda\}$. σ is **nonerasing** if for all $a \in A$, $\sigma(a) \not\ni \lambda$. For a language L, we define $\sigma(L) = \{x' \mid x' \text{ is in } \sigma(x) \text{ for some } x \in L\}$.

2.1. Prefix languages

A **prefix language** is a language L such that for all $x, y \in L$, x is a prefix of y or y is a prefix of x. A language L **determines** an infinite word α iff L is infinite and every $x \in L$ is a prefix of α. For example, the infinite prefix language $\{\lambda, \texttt{ab}, \texttt{abab}, \texttt{ababab}, \ldots\}$ determines the infinite word $(\texttt{ab})^\omega$. The following propositions are basic consequences of the definitions.

Proposition 1. *A language determines at most one infinite word.*

Proposition 2. *A language L determines an infinite word iff L is an infinite prefix language.*

Notice that while a language determines at most one infinite word, an infinite word may be determined by more than one language. In particular, we will make use of the following fact.

Proposition 3. *If a language L determines an infinite word α and L' is an infinite subset of L, then L' determines α.*

For a language class C, let

$$\omega(C) = \{\alpha \mid \alpha \text{ is an infinite word and some } L \in C \text{ determines } \alpha\}.$$

2.2. L systems

Many classes of L systems appear in the literature. The simplest L systems are D0L systems (deterministic Lindenmayer systems with 0 symbols of context), in which a morphism is successively applied to a start string or "axiom". The resulting sequence of words comprises the language of the system. Other classes of L systems are formed by adding or removing certain features. Following [11], we consider four production features (D,P,F,T) and five filtering features (E,C,N,W,H). Each production feature may be present or absent, and at most one filtering feature may be present, for a total of $2^4 \cdot 6 = 96$ classes of L systems.

Informally, the meanings of the features are as follows. "D" ("deterministic") means that each string is generated from the previous one by a morphism; if "D" is absent, the morphism is replaced with a finite substition generating multiple strings for each symbol. "P" ("propagating") means that the morphism (or other production mechanism) is nonerasing. "F" ("finite axiom set") means that the system has a finite set of possible start strings. "T" ("tables") means that the system has a finite set of possible morphisms (or finite substitions), from among which one is chosen at each step. "E" ("extended") means that a subset of the system's alphabet is designated as the "target alphabet" and only strings over the target alphabet are considered to belong to the language of the system. "H" ("homomorphism") means that a second morphism is applied after the first; "C" ("coding") means that the second morphism is a coding; "N" ("nonerasing") means that the second morphism is nonerasing; and "W" ("weak coding") means that the second morphism is a weak coding. The meanings of the features are shown in the table below, along with formal definitions for example classes.

Feature	**Meaning**	**Example**
none		A **0L system** is a tuple $G = (A, \sigma, w)$ where A is an alphabet, σ is a finite substitution on A, and w is in A^*. The language of G is $L(G) = \{s \in \sigma^i(w) \mid i \geq 0\}$.
D	Deterministic	A **D0L system** is a tuple $G = (A, h, w)$ where A is an alphabet, h is a morphism on A, and w is in A^*. The language of G is $L(G) = \{h^i(w) \mid i \geq 0\}$.
P	Propagating	A **PD0L system** is a D0L system (A, h, w) such that h is nonerasing.
F	Finite axiom set	A **DF0L system** is a tuple $G = (A, h, F)$ where A is an alphabet, h is a morphism on A, and F is a finite set of strings in A^*. The language of G is $L(G) = \{h^i(f) \mid f \in F \text{ and } i \geq 0\}$.
T	Tables	A **DT0L system** is a tuple $G = (A, H, w)$ where A is an alphabet, H is a finite nonempty set of morphisms on A (called "tables"), and w is in A^*. The language of G is $L(G) = \{s \mid h_i \cdots h_1(w) = s$ for some $h_1, \ldots, h_i \in H$.
E	Extended	An **ED0L system** is a tuple $G = (A, h, w, B)$ where A and B are alphabets and $B \subseteq A$, h is a morphism on A, and w is in A^*. The language of G is $L(G) = \{s \in B^* \mid h^i(w) = s \text{ for some } i \geq 0\}$.
H	Homomorphism	An **HD0L system** is a tuple $G = (A, h, w, g)$ such that $G' = (A, h, w)$ is a D0L system and g is a morphism on A. The language of G is $L(G) = \{g(s) \mid s$ is in $L(G')\}$.
C	Coding	A **CD0L system** is an HD0L system (A, h, w, g) such that g is a coding.
N	Nonerasing	An **ND0L system** is an HD0L system (A, h, w, g) such that g is nonerasing.
W	Weak coding	A **WD0L system** is an HD0L system (A, h, w, g) such that g is a weak coding.

These features combine to form complex L systems. For example, an **EPD0L system** is an ED0L system (A, h, w, B) such that h is nonerasing. A **T0L system** is a tuple $G = (A, T, w)$ where A is an alphabet, T is a finite nonempty set of finite substitutions on A (called "tables"), and w is in A^*. The language of G is

$L(G) = \{s \mid \sigma_i \cdots \sigma_1(w) \ni s \text{ for some } \sigma_1, \ldots, \sigma_i \in T\}$. If for all $\sigma \in T$, σ is nonerasing, then G is a **PT0L system**. An **ET0L system** is a tuple $G = (A, T, w, B)$ where (A, T, w) is a T0L system G' and $B \subseteq A$. The language of G is $L(G) = L(G') \cap B^*$. See [14] and [11] for more definitions.

We call an L system G infinite iff $L(G)$ is infinite. Finiteness of all the L systems considered in this paper is decidable from Theorem 4.1 of [11]. When speaking of language classes, we denote the class of D0L languages simply by D0L, and similarly with other classes. An **L system feature set** is a subset of {D,P,F,T} ∪ {E,C,N,W,H} containing at most one of {E,C,N,W,H}. Let $\mathscr{L}(S)$ be the language class of L systems with feature set S. For example, $\mathscr{L}(\{\text{C,D,T}\})$ = CDT0L. From the definitions of the features, we have the following inclusions.

Proposition 4 (Structural inclusions). *Let S be an L system feature set. Then:*

- $\mathscr{L}(S \cup \{D\}) \subseteq \mathscr{L}(S)$,
- $\mathscr{L}(S \cup \{P\}) \subseteq \mathscr{L}(S)$,
- $\mathscr{L}(S) \subseteq \mathscr{L}(S \cup \{F\})$, *and*
- $\mathscr{L}(S) \subseteq \mathscr{L}(S \cup \{T\})$.

Let S be an L system feature set containing none of {E,C,N,W,H}. Then:

- $\mathscr{L}(S) \subseteq \mathscr{L}(S \cup \{E\})$,
- $\mathscr{L}(S) \subseteq \mathscr{L}(S \cup \{C\})$,
- $\mathscr{L}(S \cup \{C\}) \subseteq \mathscr{L}(S \cup \{N\}) \subseteq \mathscr{L}(S \cup \{H\})$, *and*
- $\mathscr{L}(S \cup \{C\}) \subseteq \mathscr{L}(S \cup \{W\}) \subseteq \mathscr{L}(S \cup \{H\})$.

Beyond these structural inclusions, many relationships are known among the language classes; see [11]. In comparing L system classes, [11] considers two languages to be equal if they differ by the empty word only; otherwise, propagating classes would be automatically different from nonpropagating ones. See Figure 1 for a depiction of the known inclusions and equalities.

3. Pumping Lemma for T0L Systems

In this section we prove a pumping lemma for T0L systems, which we will apply in Section 4 to obtain results about infinite subsets of certain classes of L systems. The notion of pumping used in our lemma is as follows. We say that a T0L system $G = (A, T, w)$ is **pumpable** iff there are $a, b \in A$ such that (1) some $s_0 \in L(G)$ contains a, and (2) for some composition t of tables from T, $t(a)$ includes a string s_1 containing distinct occurrences of a and b and $t(b)$ includes a string s_2 containing b. See Figure 2 for a depiction of the pumping process.

Lemma 1. *Suppose the T0L system $G = (A, T, w)$ is pumpable. Then $L(G)$ is infinite.*

PROOF. Since s_0 is in $L(G)$ and t is a composition of tables from T, $t^i(s_0) \subseteq L(G)$ for every $i \geq 0$. A simple induction shows that for all $i \geq 0$, $t^i(s_0)$ includes a string containing a and at least i copies of b. Hence $L(G)$ is infinite.

To prove that every infinite T0L system is pumpable, we will need some definitions regarding derivations of T0L systems. A **T0L derivation** D is a forest (disjoint union of trees) such that for some $n \geq 0$, every path in D from a root to a leaf has length n. The roots correspond to symbols of the start string and n is the number of steps in the derivation. For $0 \leq i \leq n$, let level(i) be the set of nodes at distance i from a root. The nodes in level(0) are the roots of D and those in level(n) are the leaves. Each node is labeled with a string of length at most 1. For a node v, let label(v) be the label of v and let children(v) be the set

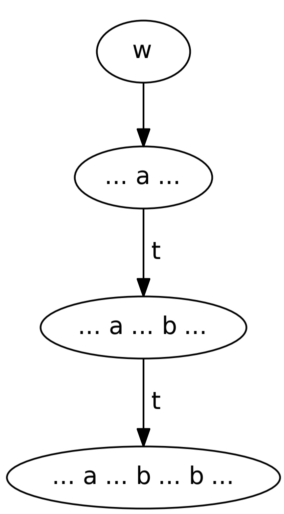

Figure 2: Schematic of our T0L pumping operation. We begin with the start string w, reach a string containing the symbol a, and then repeatedly apply the composition t of tables to obtain more and more occurrences of the symbol b.

of children of v. For a set of nodes S, let join(S) be the string formed by concatenating the labels of the nodes in S in the order in which they would be encountered in a depth-first traversal of D. (An ordering is assumed on the roots and on the children of each node.)

Let $G = (A, T, w)$ be a T0L system. For $x \in A^*$, we say that D is a derivation of x in G if

- for each node v of D, label(v) is in $A \cup \{\lambda\}$,
- join(level(0)) $= w$,
- for each $0 \leq i < n$, there is a $t \in T$ such that for each node $v \in$ level(i), there is an $s \in t$(label(v)) such that join(children(v)) $= s$, and
- join(level(n)) $= x$.

Lemma 2. *Suppose the T0L system $G = (A, T, w)$ is infinite. Then G is pumpable.*

Proof. Our proof adapts the technique of "marked sets" (sets of symbols appearing at different levels of the derivation and making a certain contribution to the derived string) from [13]. Let p be the highest $|s|$ such that s is in $t(c)$ for some $t \in T$ and $c \in A$. Let $m = |A| \cdot 2^{|A|}$. Since G is infinite, there is an $x \in A^*$ and derivation D of x in G such that $|x| > |w| \cdot p^m$. Without loss of generality, choose D such that for every node v in D with label λ, v is not a root and v is the only child of its parent. Then D consists of $|w|$ trees each of maximum outdegree at most p. We call a node **marked** if (1) it is a leaf and its label is not λ, or (2) it is the ancestor of a marked leaf. Call any node with more than one marked child a **branch node**. Since $|x| > |w| \cdot p^m$, some tree in D has more than p^m marked leaves. Then by Lemma 14 of [13], D has a path from a root to a leaf with more than m branch nodes. For $0 \leq i \leq n$, let marked(i) be the set $\{c \in A \mid$ for some marked $v \in$ level(i), label(v) $= c\}$. Then there are $0 \leq l_1 < l_2 < n$, branch nodes $v_1 \in$ level(l_1) and $v_2 \in$ level(l_2), and $a \in A$ such that v_1 is an ancestor of v_2, label(v_1) $=$ label(v_2) $= a$, and marked(l_1) $=$ marked(l_2). For each $l_1 \leq i < l_2$, let t_i be any $t \in T$ such that for each node $v \in$ level(i), there is an $s \in t$(label(v)) such that join(children(v)) $= s$. Let t be the composition $t_{l_2-1} \circ t_{l_2-2} \circ \cdots \circ t_{l_1}$. We have join(level($l_2$)) $\in t$(join(level(l_1))).

Now, since v_1 is a branch node, its set of descendants in level(l_2) contains, in addition to v_2, a marked node labelled by some $d \in A$. Since marked(l_1) $=$ marked(l_2), every $c \in A$ which labels a marked node in level(l_2) also labels a marked node in level(l_1). By definition, every marked node in level(l_1) has a marked descendant in level(l_2). A simple induction then shows that for every $i \geq 0$, there is a $c \in A$ such that level(l_2) contains a marked node labelled by c, and some $s \in t^i(d)$ contains c. Hence for every $i \geq 0$, $t^i(d)$ contains a non-empty string. So there are $j \geq 0$, $k \geq 1$ and $b \in A$ such that $t^j(d)$ includes a string containing b and $t^k(b)$ includes a string containing b. Then since $t(a)$ includes a string containing distinct occurrences of a and d, $t^{j+1}(a)$ includes a string containing distinct occurrences of a and b. Then $t^{k(j+1)}(a)$ includes a string containing distinct occurrences of a and b and $t^{k(j+1)}(b)$ includes a string containing b. So G is pumpable.

Theorem 1. *A T0L system is infinite iff it is pumpable.*

PROOF. Immediate from Lemmas 1 and 2.

Corollary 1. *A 0L system $G = (A, \sigma, w)$ is infinite iff there are $a, b \in A$ such that (1) some $s \in L(G)$ contains a, and (2) for some $i \geq 0$, $\sigma^i(a)$ includes a string containing distinct occurrences of a and b and $\sigma^i(b)$ includes a string containing b.*

Corollary 2. *A DT0L system $G = (A, H, w)$ is infinite iff there are $a, b \in A$ such that (1) some $s \in L(G)$ contains a, and (2) for some composition h of morphisms from H, $h(a)$ contains distinct occurrences of a and b and $h(b)$ contains b.*

4. Infinite Subsets of L Systems

In this section we show that all infinite languages in certain classes of L systems have infinite subsets in certain smaller classes of L systems. This limits the infinite words of the larger class to the infinite words of the smaller class. We make use of the pumping lemma for T0L systems proved in Section 3.

Theorem 2. *Every infinite T0L language has an infinite D0L subset.*

PROOF. Take any infinite T0L language L with T0L system $G = (A, T, w)$. By Theorem 1, G is pumpable for some $a, b \in A$, $s_0, s_1, s_2 \in A^*$, and composition t of tables from T. Let h be a morphism on A such that $h(a) = s_1$, $h(b) = s_2$ unless $a = b$, and for every other $c \in A$, $h(c) = s$ for some $s \in t(c)$. Consider the language L' of the D0L system (A, h, s_0). For all $c \in A$, $h(c)$ is in $t(c)$, so since s_0 is in L, we have $L' \subseteq L$. A simple induction shows that for all $i \geq 0$, $h^i(s_0)$ contains a and at least i copies of b. Then L' is an infinite D0L subset of L.

Theorem 3. *Every infinite PT0L language has an infinite PD0L subset.*

PROOF. Take any infinite PT0L language L with PT0L system $G = (A, T, w)$. Follow the proof of Theorem 2, and note that since t is a composition of tables from T and every table in T is now nonerasing, t is nonerasing, hence h is nonerasing. Then L' is an infinite PD0L subset of L.

The following theorem can be obtained using a well-known fact about periodicity of alphabets in D0L sequences (Theorem 1.1 of [14]), but we give an independent proof for completeness.

Theorem 4. *Let $G = (A, h, w, B)$ be an infinite ED0L system. Then there are $k \geq 0, p \geq 1$ such that the language of the D0L system $(A, h^p, h^k(w))$ is an infinite subset of $L(G)$.*

PROOF. Let alph(s) be the set of symbols which appear in the string s. Since $L(G)$ is infinite, there is an $m \geq 0$ such that the sequence w, $h(w)$, $h^2(w)$, $\ldots$, $h^m(w)$ contains more than $2^{|B|}$ strings in $L(G)$. For every $s \in L(G)$, alph(s) $\subseteq B$. Hence there is a $C \subseteq B$ and i, j such that $0 \leq i < j \leq m$ and alph($h^i(w)$) = alph($h^j(w)$) = C. Then for any string s such that alph(s) = C, alph($h^{j-i}(s)$) = C. Let $k = i$ and $p = j - i$. Then for every $n \geq 0$, alph($h^{k+pn}(w)$) = C. Hence for every $n \geq 0$, $h^{k+pn}(w)$ is in $L(G)$. So take the D0L system $G' = (A, h^p, h^k(w))$. We have $L(G') \subseteq L(G)$. Suppose some string s occurs twice in the derivation sequence of G'. Then s occurs twice in the derivation sequence of G, making $L(G)$ finite, a contradiction. So $L(G')$ is infinite. Therefore $L(G')$ is an infinite subset of $L(G)$.

Corollary 3. *Every infinite ED0L language has an infinite D0L subset.*

Corollary 4. *Every infinite EPD0L language has an infinite PD0L subset.*

Proof. Take any infinite EPD0L system $G = (A, h, w, B)$. By Theorem 4, there are $k \geq 0, p \geq 1$ such that the language of the D0L system $G' = (A, h^p, h^k(w))$ is an infinite subset of $L(G)$. Since h is nonerasing, h^p is nonerasing. Hence $L(G')$ is an infinite PD0L subset of $L(G)$.

Theorem 5. *Every infinite ET0L language has an infinite CD0L subset.*

Proof. Take any infinite ET0L language L. By Theorem 2.7 of [11], ET0L = CT0L. Hence there is a coding e and T0L language L' such that $L = e(L')$. Since L is infinite, L' is infinite. Then by Theorem 2, L' has an infinite D0L subset L''. Since L'' is infinite and e is a coding, $e(L'')$ is infinite. Since $L'' \subseteq L'$, $e(L'') \subseteq e(L')$. Therefore $e(L'')$ is an infinite CD0L subset of L.

Theorem 6. *Let S be an L system feature set not containing F. Then every infinite $\mathscr{L}(S \cup \{F\})$ language has an infinite $\mathscr{L}(S)$ subset.*

Proof. Take any infinite L system G with feature set $S \cup \{\mathrm{F}\}$. Since G has a finite axiom set, $L(G)$ is a finite union of $\mathscr{L}(S)$ languages. Then since $L(G)$ is infinite, at least one of these $\mathscr{L}(S)$ languages is infinite. Therefore $L(G)$ has an infinite $\mathscr{L}(S)$ subset.

Theorem 7. *Let C and D be language classes such that every infinite language in C has an infinite subset in D. Then $\omega(C) \subseteq \omega(D)$.*

Proof. Take any $\alpha \in \omega(C)$. Some $L \in C$ determines α. Then L is infinite, so L has an infinite subset L' in D. Then L' determines α. So α is in $\omega(D)$. Hence $\omega(C) \subseteq \omega(D)$.

5. Categorizations

In this section we categorize the infinite words determined by each class of L systems, making use of the results about infinite subsets from the previous section. We partition the 96 classes into three sets, called Set_1, Set_2, and Set_3, and show that for every $C_1 \in Set_1$, $C_2 \in Set_2$, and $C_3 \in Set_3$, $\omega(C_1) = \omega(\mathrm{PD0L})$, $\omega(C_2) = \omega(\mathrm{D0L})$, and $\omega(C_3) = \omega(\mathrm{CD0L})$.

5.1. PD0L classes

Let $Set_1 = \{$PD0L, PDF0L, P0L, PF0L, PDT0L, PDTF0L, PT0L, PTF0L, EPD0L, EPDF0L$\}$.

Theorem 8. *For every $C \in Set_1$, every infinite C language has an infinite PD0L subset.*

Proof. Take any $C \in Set_1$. By structural inclusion, $C \subseteq$ PTF0L or $C \subseteq$ EPDF0L. By Theorem 6, every infinite PTF0L language has an infinite PT0L subset. By Theorem 3, every infinite PT0L language has an infinite PD0L subset. Hence every infinite PTF0L language has an infinite PD0L subset. By Theorem 6, every infinite EPDF0L language has an infinite EPD0L subset. By Corollary 4, every infinite EPD0L language has an infinite PD0L subset. Hence every infinite EPDF0L language has an infinite PD0L subset. Hence every infinite C language has an infinite PD0L subset.

Theorem 9. *For every $C \in Set_1$, $\omega(C) = \omega(PD0L)$.*

Proof. Take any $C \in Set_1$. By structural inclusion, PD0L $\subseteq$ C. Hence $\omega(\mathrm{PD0L}) \subseteq \omega(C)$. By Theorem 8, every infinite C language has an infinite PD0L subset. Then by Theorem 7, $\omega(C) \subseteq \omega(\mathrm{PD0L})$. Therefore $\omega(C) = \omega(\mathrm{PD0L})$.

5.2. D0L classes

Let Set_2 = {D0L, DF0L, 0L, F0L, DT0L, DTF0L, T0L, TF0L, ED0L, EDF0L}.

Theorem 10. *For every $C \in Set_2$, every infinite C language has an infinite D0L subset.*

PROOF. Take any $C \in Set_2$. By structural inclusion, $C \subseteq$ TF0L or $C \subseteq$ EDF0L. By Theorem 6, every infinite TF0L language has an infinite T0L subset. By Theorem 2, every infinite T0L language has an infinite D0L subset. Hence every infinite TF0L language has an infinite D0L subset. By Theorem 6, every infinite EDF0L language has an infinite ED0L subset. By Corollary 3, every infinite ED0L language has an infinite D0L subset. Hence every infinite EDF0L language has an infinite D0L subset. Hence every infinite C language has an infinite D0L subset.

Theorem 11. *For every $C \in Set_2$, $\omega(C) = \omega$(D0L).*

PROOF. Take any $C \in Set_2$. By structural inclusion, D0L $\subseteq$ C. Hence ω(D0L) $\subseteq \omega(C)$. By Theorem 10, every infinite C language has an infinite D0L subset. Then by Theorem 7, $\omega(C) \subseteq \omega$(D0L). Therefore $\omega(C)$ = ω(D0L).

5.3. CD0L classes

Let Set_3 = {CD0L, ND0L, WD0L, HD0L, CPD0L, NPD0L, WPD0L, HPD0L, CDF0L, NDF0L, WDF0L, HDF0L, CPDF0L, NPDF0L, WPDF0L, HPDF0L, E0L, C0L, N0L, W0L, H0L, EP0L, CP0L, NP0L, WP0L, HP0L, EF0L, CF0L, NF0L, WF0L, HF0L, EPF0L, CPF0L, NPF0L, WPF0L, HPF0L, EDT0L, CDT0L, NDT0L, WDT0L, HDT0L, EPDT0L, CPDT0L, NPDT0L, WPDT0L, HPDT0L, EDTF0L, CDTF0L, NDTF0L, WDTF0L, HDTF0L, EPDTF0L, CPDTF0L, NPDTF0L, WPDTF0L, HPDTF0L, ET0L, CT0L, NT0L, WT0L, HT0L, EPT0L, CPT0L, NPT0L, WPT0L, HPT0L, ETF0L, CTF0L, NTF0L, WTF0L, HTF0L, EPTF0L, CPTF0L, NPTF0L, WPTF0L, HPTF0L}.

Theorem 12. *For every $C \in Set_3$, every infinite C language has an infinite CD0L subset.*

PROOF. Take any $C \in Set_3$. By structural inclusion, $C \subseteq$ ETF0L or $C \subseteq$ HTF0L. By Theorem 2.7 of [11], ETF0L = HTF0L = ET0L. So $C \subseteq$ ET0L. By Theorem 5, every infinite ET0L language has an infinite CD0L subset. Hence every infinite C language has an infinite CD0L subset.

Theorem 13. *For every $C \in Set_3$, $\omega(C) = \omega$(CD0L).*

PROOF. Take any $C \in Set_3$. By Theorem 12, every infinite C language has an infinite CD0L subset. Then by Theorem 7, $\omega(C) \subseteq \omega$(CD0L).

Next, by structural inclusion, CPD0L $\subseteq C$ or EP0L $\subseteq C$ or EPDT0L $\subseteq C$. By Theorem 2.4 of [11], EP0L = C0L, so CPD0L $\subseteq$ EP0L. By Theorem 2.6 of [11], CPDT0L $\subseteq$ EPDT0L, so CPD0L $\subseteq$ EPDT0L. Hence CPD0L $\subseteq C$. Now by Theorem 2.3 of [11], CPDF0L = CDF0L. Hence CD0L $\subseteq$ CPDF0L. Hence ω(CD0L) $\subseteq \omega$(CPDF0L). By Theorem 6, every infinite CPDF0L language has an infinite CPD0L subset. Then by Theorem 7, ω(CPDF0L) $\subseteq \omega$(CPD0L). Hence ω(CD0L) $\subseteq \omega$(CPD0L) $\subseteq \omega(C)$.

Therefore $\omega(C) = \omega$(CD0L).

For any classes C_1, C_2 in the same Set_i, the above theorems show that $\omega(C_1) = \omega(C_2)$. Notice that this does not imply that the class of C_1 prefix languages equals the class of C_2 prefix languages. For example, from Theorem 11, we have ω(0L) = ω(D0L), but where c is a symbol, the prefix language $c^* = \{\lambda, c, cc, ccc, \dots\}$ is in 0L but not in D0L.

6. ω(PD0L), ω(D0L), and ω(CD0L)

In this section, we separate the three classes of infinite words obtained in the previous section, giving ω(PD0L) $\subset \omega$(D0L) $\subset \omega$(CD0L). We observe that ω(D0L) properly contains the pure morphic words and we show that ω(CD0L) contains exactly the morphic words.

6.1. Separating the classes

From Theorem 2.3 of [12], the infinite words generated by iterating nonerasing morphisms are a proper subset of the pure morphic words, which in turn are a proper subset of the morphic words. Our classes ω(PD0L), ω(D0L), and ω(CD0L) are defined more generally using prefix languages, but similar arguments serve to separate them.

Theorem 14. *ω(PD0L) $\subset$ ω(D0L).*

PROOF. By structural inclusion, ω(PD0L) $\subseteq$ ω(D0L). To separate the two classes, we use an infinite word from [4]. Let $A = \{0, 1, 2\}$. Let f be a morphism on A such that $f(0) = 01222$, $f(1) = 10222$, and $f(2) = \lambda$. Let $\alpha = f^\omega(0) = 01222102221022201222\cdots$. Then α is a pure morphic word, hence α is in ω(D0L). In [4] it is shown that there is no nonerasing morphism g on A such that $g^\omega(0) = \alpha$. We generalize this result to show that α is not in ω(PD0L). First, we show that if g is a nonerasing morphism on A and $g(\alpha) = \alpha$, then g is an identity morphism. We adapt the proof of Example 3 in [4].

Let τ be the Thue-Morse word $\tau = 01101001\cdots = u^\omega(0)$, where u is a morphism on $\{0, 1\}$ such that $u(0) = 01$ and $u(1) = 10$. Let d be a morphism on A such that $d(0) = 0$, $d(1) = 1$, and $d(2) = \lambda$. As observed by [4], $d(\alpha) = \tau$. Notice that neither 2222 nor 212 is a subword of α.

Suppose g is a nonerasing morphism on A and $g(\alpha) = \alpha$. Since α starts with 0, $g(0) = 0x$ for some $x \in A^*$. If $x \neq \lambda$, then $g^\omega(0) = \alpha$, contradicting the result from [4] that there is no such nonerasing morphism g. Therefore $g(0) = 0$.

Suppose $g(2)$ is not in 2^*. Let $s = d(g(2))$. Then s is not empty. Since 222 is a subword of α and $g(\alpha) = \alpha$, $g(222)$ is a subword of α. Then since $d(\alpha) = \tau$, τ contains $d(g(222)) = sss$, a contradiction, since τ is known to be cubefree. So $g(2)$ is in 2^*. Then since α contains $g(222)$, and 2222 is not a subword of α, and g is nonerasing, $g(2) = 2$.

Suppose $g(1) \neq 1$. Then since α begins with 012, $g(1) = 12z$ for some $z \in A^*$. Since 2221 is a subword of α, $g(2221) = 22212z$ is a subword of α, a contradiction, since α does not contain the subword 212. So $g(1) = 1$. We now have $g(0) = 0$, $g(1) = 1$, and $g(2) = 2$, so g is an identity morphism.

So suppose α is in ω(PD0L). Then there is a PD0L system $G = (A, h, w)$ such that $L(G)$ determines α. Then h is a nonerasing morphism on A and $h(\alpha) = \alpha$, so by what we have shown, h is an identity morphism. But then $h(w) = w$, so $L(G)$ is finite, a contradiction. Therefore α is not in ω(PD0L). Hence ω(PD0L) $\subset$ ω(D0L).

Theorem 15. *ω(D0L) $\subset$ ω(CD0L).*

PROOF. By structural inclusion, ω(D0L) $\subseteq$ ω(CD0L). Let $\alpha = \mathtt{abba}^\omega$. Since α is ultimately periodic, α is morphic, hence α is in ω(CD0L). Suppose α is in ω(D0L). Then there is a D0L system $G = (A, h, w)$ such that $L(G)$ determines α. Clearly $h(\mathtt{a})$ cannot include b, and if $h(\mathtt{a}) = \lambda$, $L(G)$ is finite, a contradiction. So since $h(\mathtt{a})$ must be a prefix of α, $h(\mathtt{a}) = \mathtt{a}$. Then a $h(\mathtt{b})$ $h(\mathtt{b})$ is a prefix of α, hence $h(\mathtt{b}) = \lambda$ or $h(\mathtt{b}) = \mathtt{b}$. But then $L(G)$ is finite, a contradiction. So α is not in ω(D0L). Hence ω(D0L) $\subset$ ω(CD0L).

It is a consequence of Theorem 6 that the F feature does not affect the determined infinite words. For all of the other features (D,P,T,E,C,N,W,H), according to the above theorems and those of Section 5, there is at least one case in which the addition of that feature affects the class of determined infinite words. For example, for D we have ω(ED0L) $\subset$ ω(E0L), for P we have ω(PD0L) $\subset$ ω(D0L), for T we have ω(ED0L) $\subset$ ω(EDT0L), for E we have ω(0L) $\subset$ ω(E0L), for C we have ω(D0L) $\subset$ ω(CD0L), for N we have ω(D0L) $\subset$ ω(ND0L), for W we have ω(D0L) $\subset$ ω(WD0L), and for H we have ω(D0L) $\subset$ ω(HD0L).

6.2. Characterizing the words in each class

That ω(D0L) includes every pure morphic word is immediate from the definitions. In [8], the infinite word $\mathtt{aab}^\omega$ is given as an example of an infinite D0L word which is not pure morphic. Hence ω(D0L) properly contains the pure morphic words. Next, we show that ω(CD0L) contains exactly the morphic words. The **adherence** of a language L, denoted Adherence(L), is the set $\{\alpha \mid \alpha$ is an infinite word and for every prefix p of α, there is an $s \in L$ such that p is a prefix of $s\}$.

Lemma 3. *Suppose L is in D0L and α is in* Adherence(L). *Then α is morphic.*

PROOF. From [7], either (1) α is ultimately periodic, or (2) $\alpha = w\ x\ h(x)\ h^2(x)\ \cdots$ for some morphism h and strings w, x such that $h(w) = wx$ and x is not mortal. If (1), α is morphic. If (2), α is an infinite D0L word, so by Proposition 10.2.2 of [8], α is morphic.

Theorem 16. *α is in ω(CD0L) iff α is morphic.*

PROOF. That ω(CD0L) includes every morphic word is immediate from the definitions. So take any $\alpha \in \omega$(CD0L). Then there is a CD0L system $G = (A, h, w, e)$ such that $L(G)$ determines α. Then $L(G)$ is infinite. Hence the language L of the D0L system (A, h, w) is infinite. As noted in [7], a language has empty adherence iff the language is finite. Therefore there is an $\alpha' \in$ Adherence(L). By Lemma 3, α' is morphic. Now for any prefix p of α', there is a string s in L with p as a prefix. Then $e(p)$ is a prefix of $e(s)$. Then since $e(s)$ is in $L(G)$, $e(p)$ is a prefix of α. So for every prefix p of α', $e(p)$ is a prefix of α. Since e is a coding, $e(\alpha')$ is infinite. So $e(\alpha') = \alpha$. Then because a coding of a morphic word is still a morphic word, α is morphic. Hence α is in ω(CD0L) iff α is morphic.

7. Conclusion

In this paper we have categorized the infinite words determined by L systems, showing that a variety of classes of L systems collapse to just three classes of infinite words. To associate L systems with infinite words, we used the concept of prefix languages. This concept can be applied not just to L systems, but to arbitrary language classes, offering many opportunities for further research. That is, where C is any language class, we denote by $\omega(C)$ the class of infinite words determined by the prefix languages in C. Then for a given language class, we can ask what class of infinite words it determines. From the other direction, for a given infinite word, we can ask in what language classes it can be determined. It is hoped that work in this area will help to build up a theory of the complexity of infinite words with respect to what language classes can determine them. See [17] and [18] for progress along these lines.

Acknowledgments.

I want to thank my advisor, Rajmohan Rajaraman, for supporting this work, encouraging me, and offering many helpful comments and suggestions. I would also like to thank the anonymous reviewers for their valuable input.

References

[1] Allouche, J.-P., & Shallit, J. (2003). *Automatic Sequences: Theory, Applications, Generalizations*. New York, NY, USA: Cambridge University Press.
[2] Berstel, J. (1989). Properties of infinite words : Recent results. In *STACS 89* (pp. 36–46). Springer Berlin Heidelberg volume 349 of *Lecture Notes in Computer Science*. doi:10.1007/BFb0028971.
[3] Book, R. V. (1977). On languages with a certain prefix property. *Mathematical Systems Theory*, *10*, 229–237.
[4] Cassaigne, J., & Nicolas, F. (2003). Quelques propriétés des mots substitutifs. *Bulletin of the Belgian Mathematical Society-Simon Stevin*, *10*, 661–677.
[5] Culik, K., & Karhumäki, J. (1994). Iterative devices generating infinite words. *Int. J. Found. Comput. Sci.*, *5*, 69–97.
[6] Culik, K., & Salomaa, A. (1982). On infinite words obtained by iterating morphisms. *Theoretical Computer Science*, *19*, 29–38. doi:10.1016/0304-3975(82)90013-5.
[7] Head, T. (1984). Adherences of D0L languages. *Theoretical Computer Science*, *31*, 139–149. doi:10.1016/0304-3975(84)90131-2.
[8] Honkala, J. (2010). The equality problem for purely substitutive words. In *Combinatorics, automata, and number theory* (pp. 505–529). Cambridge: Cambridge University Press.
[9] Honkala, Juha (2015). The finiteness problem for monoids of morphisms. *RAIRO-Theor. Inf. Appl.*, *49*, 61–65. doi:10.1051/ita/2014028.
[10] Jungers, R. M., Protasov, V., & Blondel, V. D. (2008). Efficient algorithms for deciding the type of growth of products of integer matrices. *Linear Algebra and its Applications*, *428*, 2296–2311. doi:10.1016/j.laa.2007.08.001.
[11] Kari, L., Rozenberg, G., & Salomaa, A. (1997). L systems. In G. Rozenberg, & A. Salomaa (Eds.), *Handbook of Formal Languages, Vol. 1* (pp. 253–328). New York, NY, USA: Springer-Verlag New York, Inc.

[12] Pansiot, J.-J. (1985). On various classes of infinite words obtained by iterated mappings. In M. Nivat, & D. Perrin (Eds.), *Automata on Infinite Words* (pp. 188–197). Springer Berlin Heidelberg volume 192 of *Lecture Notes in Computer Science*. doi:10.1007/3-540-15641-0_34.

[13] Rabkin, M. (2012). Ogden's lemma for ET0L languages. In *Proceedings of the 6th International Conference on Language and Automata Theory and Applications* LATA'12 (pp. 458–467). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-28332-1_39.

[14] Rozenberg, G., & Salomaa, A. (1980). *Mathematical Theory of L Systems*. Orlando, FL, USA: Academic Press, Inc.

[15] Smith, T. (2013). Infiniteness and boundedness in 0L, DT0L, and T0L systems. In A.-H. Dediu, C. Martín-Vide, & B. Truthe (Eds.), *Language and Automata Theory and Applications* (pp. 535–546). Springer Berlin Heidelberg volume 7810 of *Lecture Notes in Computer Science*. doi:10.1007/978-3-642-37064-9_47.

[16] Smith, T. (2013). On infinite words determined by L systems. In J. Karhumäki, A. Lepistö, & L. Zamboni (Eds.), *Combinatorics on Words* (pp. 238–249). Springer Berlin Heidelberg volume 8079 of *Lecture Notes in Computer Science*. doi:10.1007/978-3-642-40579-2_25.

[17] Smith, T. (2013). On Infinite Words Determined by Stack Automata. In *FSTTCS 2013* (pp. 413–424). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik volume 24 of *Leibniz International Proceedings in Informatics (LIPIcs)*. doi:10.4230/LIPIcs.FSTTCS.2013.413.

[18] Smith, T. (2014). On infinite words determined by indexed languages. In E. Csuhaj-Varjú, M. Dietzfelbinger, & Z. Ésik (Eds.), *Mathematical Foundations of Computer Science 2014* (pp. 511–522). Springer Berlin Heidelberg volume 8634 of *Lecture Notes in Computer Science*. doi:10.1007/978-3-662-44522-8_43.

[19] Yamakami, T., & Suzuki, T. (2005). Resource bounded immunity and simplicity. *Theoretical Computer Science*, *347*, 90–129. doi:10.1016/j.tcs.2005.03.055.