Contents

PART C

FORMAL LANGUAGES AND BIOLOGY

Chapter 11

Lindenmayer Systems

11.1 The Basic Model – 0L Systems

11.1.1 Two Biological Examples

We start with two biological examples describing the development of an alga and a moss.

In Figure 11.1 the first 10 stages of the development of a red alga is shown.

Any small part represents a cell; thus stage a) is formed by one cell; stage b) consists of two cells and stage c) of four cells. Starting with stage d) we see a branching structure of the alga. Thus the first problem consists in the description of the branching structure. We choose a word over the alphabet consisting of the letters c, (and). c represents a cell and (and) are used to describe the branching. If we have a word $c^r(c^s)c^t$, then the central part of the alga is given by c^rc^t and the subword c^s describes a branch. By this method we do not distinguish between branches to the left or to the right etc. Furthermore, we can iterate the process, i. e., if we have a word $c^n(c^r(c^s)c^t)c^m$, then c^rc^t is a branch of c^nc^m and c^s is a branch of the branch c^rc^t.

Then we can describe the stages given in Figure 11.1 as follows:

a) c
b) cc
c) $cccc$
d) $cc(c)cccc$
e) $cc(cc)cc(c)cccc$
f) $cc(ccc)cc(cc)cc(c)cccc$
g) $cc(cccc)cc(ccc)cc(cc)cc(c)cccc$
h) $cc(ccccc)cc(cccc)cc(ccc)cc(cc)cc(c)cccc$
i) $cc(cccccc)cc(ccccc)cc(cccc)cc(cccc)cc(cc)cc(c)cccc$
j) $cc(ccccccc)cc(cccccc)cc(ccccc)cc(cc(c)cccc)cc(cccc)cc(cc)cc(c)cccc$

The development from stage a) to stage b) can be considered as a division of the cell c resulting in cc. If we apply this division to both cells of stage b), again, then we get the four cells of stage c). But now we cannot continue in this way by two reasons: Stage d) does not consist of eight cells (which would be obtained from the division of four cells) and we cannot model the branching which occurs in stage d). In order to solve this problem one can introduce more rules for the cells, or one makes a further differentiation of the

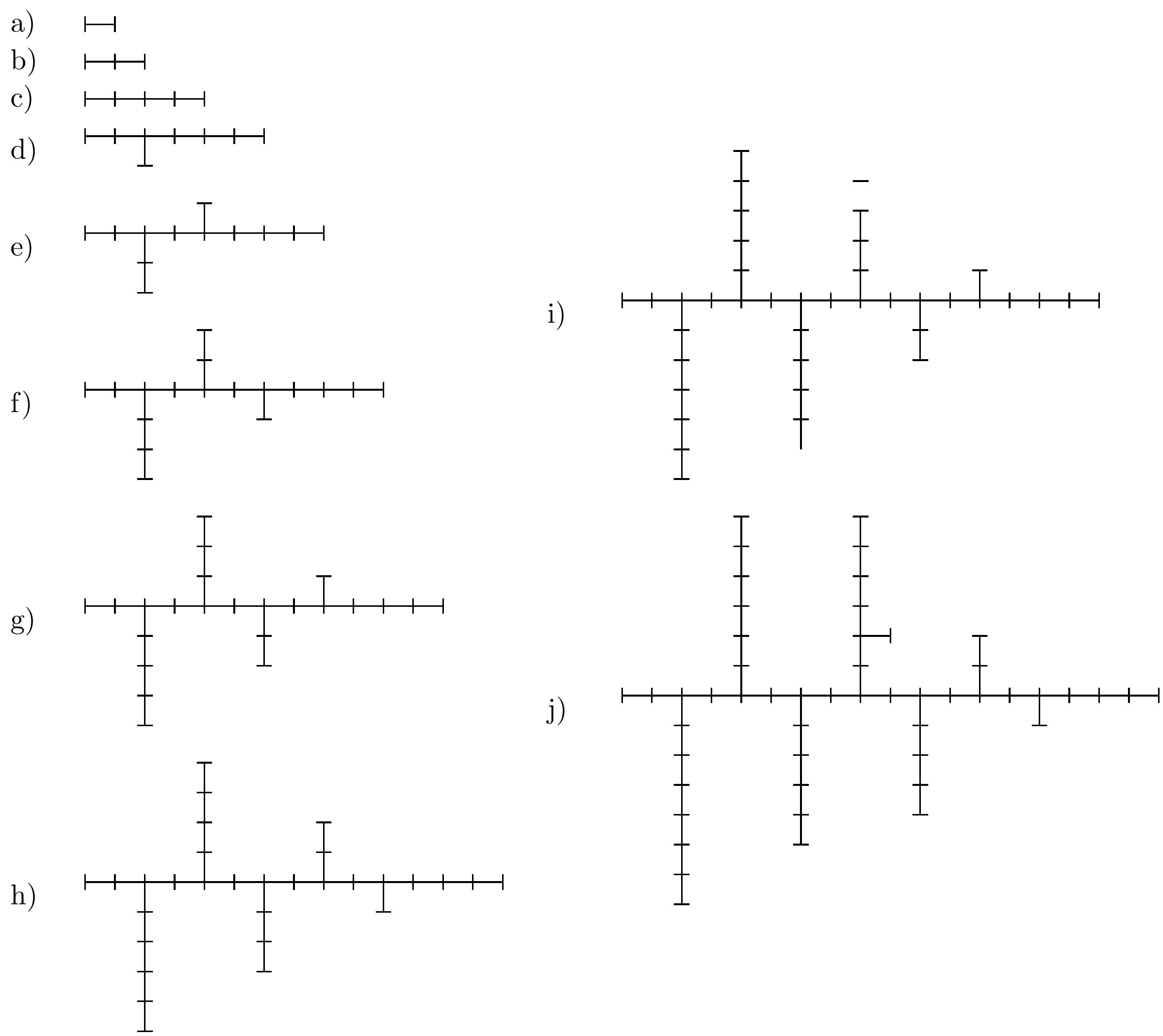

Figure 11.1: First stages of the development of a red alga

cell by introducing some states of the cell and different rules for different states.

We use the second approach and distinguish 10 states of cell c which we denote by the digits

$$0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9$$

of the decimal system. Moreover, we consider the rules

$$\begin{array}{lllll} 0 \to 10 & 1 \to 32 & 2 \to 3(4) & 3 \to 3 & 4 \to 56 \\ 5 \to 37 & 6 \to 58 & 7 \to 3(9) & 8 \to 50 & 9 \to 39 \end{array}$$

for the states where the left hand side gives the state a of the cell and the right hand side gives the part which is obtained from a in one step of the development. The rules for 0 and 1 can be interpreted as divisions of one cell into two cells; the rules for 2 and 7 can be considered as the starting of a branch. The rule $3 \to 3$ can be omitted because it says that c in state 3 is not changed in the sequel. However, if we want to describe the development, then we have to tell what happens with each cell at every moment. Thus we add $3 \to 3$ in order to know what happens to cells in state 3.

Then we obtain the following description of the first stages of the development of the red alga and one sees that this corresponds to the stages given in Figure 11.1:

a) 4
b) 56
c) 3758
d) 33(9)3750
e) 33(39)33(9)3710
f) 33(339)33(39)33(9)3210
g) 33(3339)33(339)33(39)33(4)3210
h) 33(33339)33(3339)33(339)33(56)33(4)3210
i) 33(333339)33(33339)33(3339)33(3758)33(56)33(4)3210
j) 33(3333339)33(333339)33(33339)33(33(9)3750)33(3758)33(56)33(4)3210

We now consider the moss *Phascum cuspidatum*. A typical leaf of Phascum cuspidatum is shown in Figure 11.2. It consists of three types of cells: cells of type I are at the top of the leaf, cells of type II are along the margin of the leaf, and cells of type III form the inner part of the leaf.

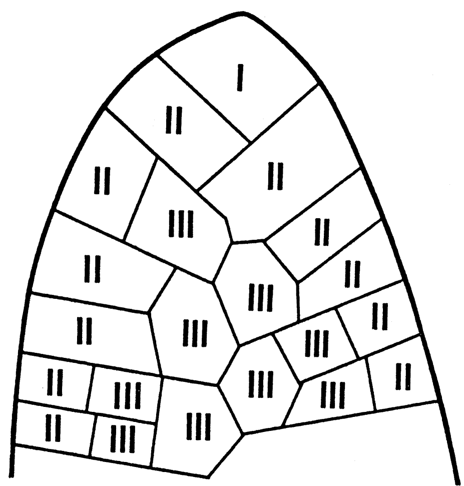

Figure 11.2: Leaf of the moss *Phascum cuspidatum*

The development of Phascum cuspidatum was already considered in 1845 by the Swiss biologist Carl Wilhelm von Nägeli (1817–1891). He noticed that, essentially, we have the developmental rules

$$I \to I + II, \;\; II \to II + II \;\text{ and } II \to II + III$$

and the rule $III \to III$ which says that cells of type III are not changed in the developmental process. However, as in the first example, in order to be precise one has to distinguish different states of the cells, because e. g.
– cells of type II do not changed according to one of the rules above in every step,
– cells of type I are changed in every step, however, they produce the cells of type II alternately to the right and to the left.

We describe a leaf as a square where the upper left corner corresponds to the top of the leaf. We use cells of type I_i and II_i^r where the lower index i is a number and reflects the "age" of the cell and the upper index $r \in \{o, l\}$ gives the margin where l stands for the left margin and o for the upper margin).

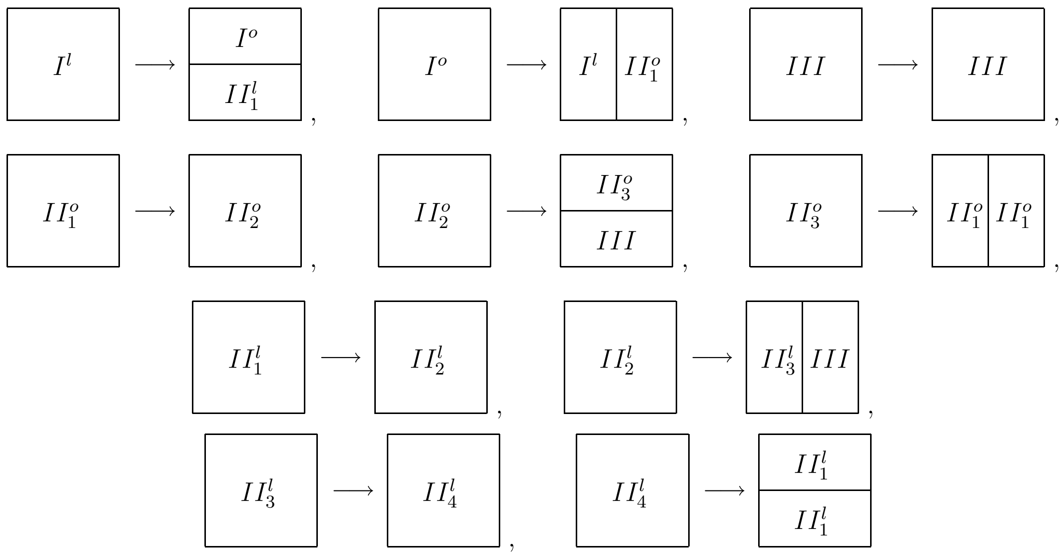

Figure 11.3: Rules for the development of the moss *Phascum cuspidatum*

Figure 11.3 gives the more detailed rules and in Figure 11.4 the first stages of the development according to these rules starting with a single cell of type I are shown. It is easy to see that the last stage corresponds to the leaf given in Figure 11.2.

11.1.2 Definitions and Examples

Looking on the examples presented in the preceding subsection we see that a formalization of them has to take into consideration the following aspects:

- in one step all cells or at least some of them are changed according to the rules in parallel, i. e., the rewriting is not a sequential process as in the case of phrase structure grammars,

- in order to describe an organism we have to take into consideration all cells, independent of the fact whether there exist rules for the cells or the cells do not change in the further development, i. e., we do not distinguish between terminals and nonterminals as in phrase structure grammars.

We now introduce Lindenmayer systems as a new type of rewriting systems. We restrict to the case of words for simplicity. For approaches to multidimensional systems we refer to Section VI.5 of [27], [3] and parallel graph grammars (e. g., [14]). Moreover, we mention that by the method used in the description of the development of some red alga we are able to cover some multidimensional cases as branching structures by means of (linear) words.

Definition 11.1 *A Lindenmayer system without interaction (0L system, for short) is a triple $G = (V, P, \omega)$ where*

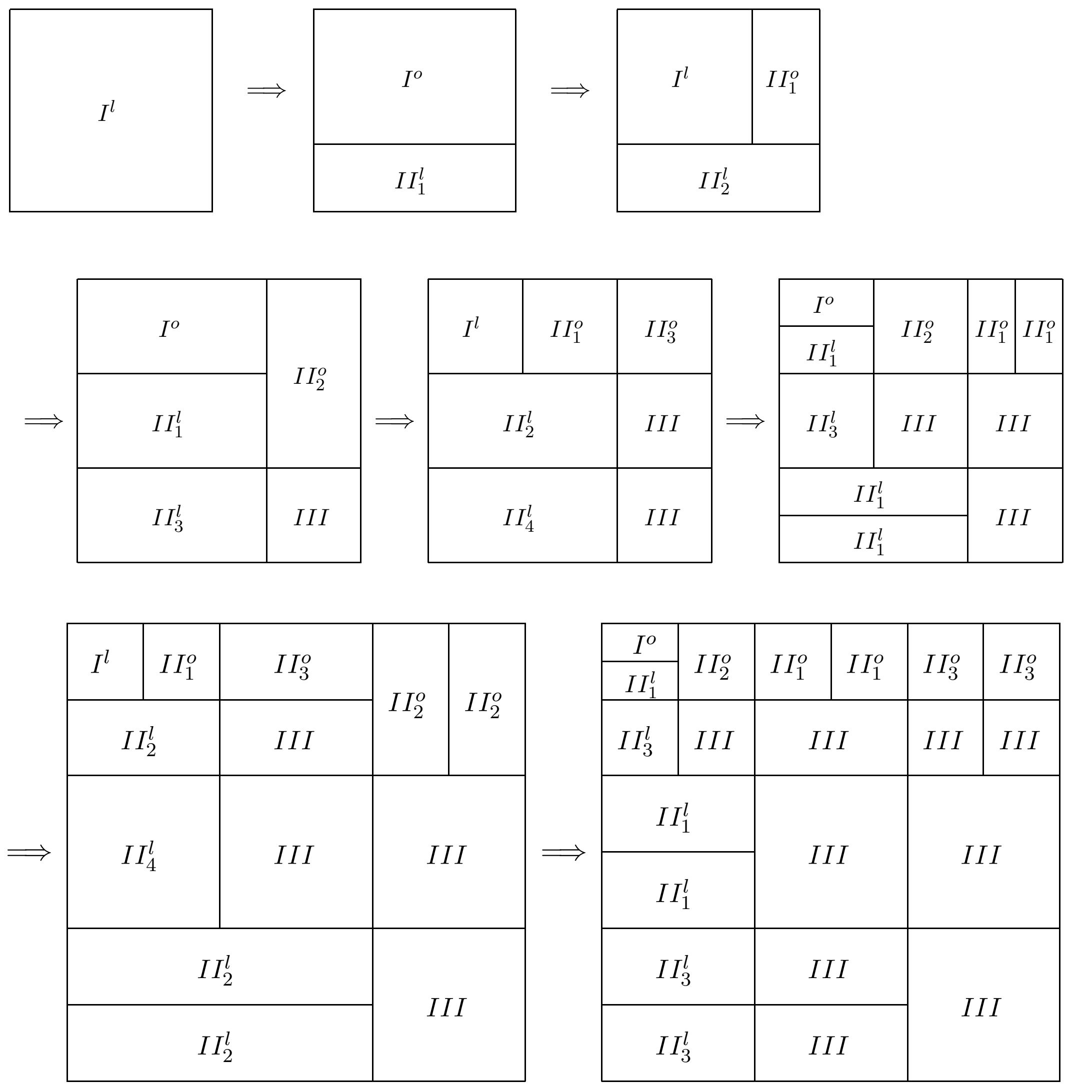

Figure 11.4: First stages of the development of the moss *Phascum cuspidatum*

- *V is an alphabet,*
- *P is a finite complete set of productions over V, i. e., P is a finite subset of $V^+ \times V^*$ and, for any $a \in V$, there is a word w_a such that $(a, w_a) \in P$,*
- *$\omega \in V^+$.*

The elements of the alphabet represent the cells.

Any production of P is a description of a developmental rule. As usual, instead of (a, w) in P we write $a \to w$. Note that by the completeness condition we require that, for any letter or any cell, there is a developmental rule. Thus we have taken the rules $3 \to 3$ and $III \to III$ to describe the development of the red alga and Phascum cuspidatum in the preceding subsection which reflect that the cells are not changed in the further development. However, the set of rules for the red alga is not complete since we have no rules for the letters (and) which are used to model branches. In order to get a complete set one has to add $(\to ($ and $) \to)$ which are clear from the biological motivation since the places of branchings do not move during the development.

The word ω represents the organism which we have in the first stage of the development. We call it the start word of the system. Obviously, it is not necessary that we start with a cell which requires that the start element has to be a (non-empty) word.

We now define the derivation process in a 0L system.

Definition 11.2 *Let $G = (V, P, \omega)$ be a 0L system. For two words $x \in V^+$ and $y \in V^*$, we say that x* directly derives *y in G, written as $x \Longrightarrow_G y$, or $x \Longrightarrow y$ if G is clear from the context, if and only if the following conditions are satisfied:*

– $x = x_1 x_2 \dots x_n$, *where* $x_i \in V$ *for* $1 \leq i \leq n$,
– $y = y_1 y_2 \dots y_n$,
– $x_i \to y_i \in P$ *for* $1 \leq i \leq n$.

Moreover, we sometimes use $\lambda \Longrightarrow_G \lambda$.

By this definition, in every derivation step we replace any letter of x according to rules of P. Thus we have a completely parallel derivation process.

The replacement of a letter x_i of x does not depend on the neighbouring letters x_{i-1} and x_{i+1}; we only have to use a rule of P. Thus there is no interaction between the letters of the word during a derivation. Hence one can say that we have a parallel context-free derivation process. The 0 (zero) in Definition 11.1 stands for no (or 0) interaction.

By $\stackrel{*}{\Longrightarrow}$ we denotes the reflexive and transitive closure of $\Longrightarrow$. Then $x \stackrel{*}{\Longrightarrow} y$ holds if and only if $x = y$ (reflexivity) or there are a natural number $r \geq 1$ and words $z_0, z_1, z_2, \dots, z_r$ such that

$$x = z_0 \Longrightarrow z_1 \Longrightarrow z_2 \Longrightarrow \dots \Longrightarrow z_{r-1} \Longrightarrow z_r = y$$

(transitivity).

Definition 11.3 *Let $G = (V, P, \omega)$ be a 0L system. The* language $L(G)$ generated by G *is defined as*

$$L(G) = \{z \mid \omega \stackrel{*}{\Longrightarrow} z\}\,.$$

By this definition, the language generated by a 0L system consists of all words which can be generated from the start element ω.

We set

$$\begin{aligned} L_0(G) &= \{\omega\}, \\ L_n(G) &= \{z \mid v \Longrightarrow z \text{ for some } v \in L_{n-1}(G)\} \quad \text{for} \quad n \geq 1. \end{aligned}$$

By induction (on n) it is easy to prove that $L_n(G)$ consists of all words y such that there is a derivation

$$\omega = z_0 \Longrightarrow z_1 \Longrightarrow z_2 \Longrightarrow \dots \Longrightarrow z_{n-1} \Longrightarrow z_n = y\,.$$

Thus we get

$$L(G) = \bigcup_{n \geq 0} L_n(G)\,.$$

Before we give some examples we want to mention the differences between 0L systems and the phrase structure grammars.

- We have only one alphabet and no distinction between terminals and nonterminals.
- The language of a 0L system consists of all words generated by the systems, whereas the language generated by a phrase structure grammar only contains words over the terminal alphabet, which is a (proper) subset of all words generated by the grammar.
- In a derivation step of a 0L systems all letters of the current word are replaced, whereas in a derivation step of a phrase structure grammar subwords of a bounded length and in the case of a context-free grammar one letter is only replaced. This means that 0L systems are characterized by a purely parallel derivation process whereas context-free grammars are characterized by a purely sequential process.
- The derivation in a 0L system starts with a non-empty word over the underlying alphabet. In phrase structure grammars the derivation starts with a distinguished nonterminal.

Example 11.4 We consider the 0L system

$$G_1 = (\{a\}, \{a \to a^2\}, a)\,.$$

By induction, we prove that $L_n(G_1) = \{a^{2^n}\}$ for $n \geq 0$. By definition, $L_0(G_1) = \{a\}$ since a is the start word. Thus the basis of the induction is shown. Let $L_n(G_1) = \{a^{2^n}\}$. Because $L_{n+1}(G_1) = \{z \mid a^{2^n} \Longrightarrow z\}$ and $a^{2^n} \Longrightarrow (a^2)^{2^n} = a^{2^{n+1}}$ is the only derivation from a^{2^n}, we get $L_{n+1}(G_1) = \{a^{2^{n+1}}\}$. Therefore the induction step has been proved, too.

Hence we obtain

$$L(G_1) = \bigcup_{n\geq 0} \{a^{2^n}\} = \{a^{2^n} \mid n \geq 0\}\,.$$

Example 11.5 Let

$$G_2 = (\{a, b\}, \{a \to \lambda, b \to ab\}, aab)\,.$$

Then we only have the derivation

$$aab \Longrightarrow \lambda\lambda ab = ab \Longrightarrow \lambda ab = ab \Longrightarrow ab \Longrightarrow ab \Longrightarrow \ldots,$$

which results in

$$L(G_2) = \{aab, ab\}\,.$$

Example 11.6 We consider the 0L system

$$G_3 = (\{a\}, \{a \to a, a \to a^2\}, a)\,.$$

We show that

$$L(G_3) = \{a^n \mid n \geq 1\}. \tag{11.1}$$

This can be seen as follows. First, by induction, we prove $a^n \in L_{n-1}(G_3)$. By definition, we have $L_0(G) = \{a\}$. Further, applying $a \to a$ to the first $n-1$ occurrences of a in a^n and $a \to a^2$ to the last letter of a^n, we get $a^n = a^{n-1}a \Longrightarrow a^{n-1}a^2 = a^{n+1}$. Therefore $a^n \in L_{n-1}(G_3)$ implies $a^{n+1} \in L_n(G_3)$, and the induction step is performed. Thus we have

$$\{a^n \mid n \geq 1\} \subseteq \bigcup_{n\geq 0} L_n(G_3) = L(G_3)\,.$$

On the other hand, obviously from a word a^n we can only generate non-empty words over $\{a\}$ by application of $a \to a$ and $a \to a^2$. Hence (11.1) holds.

Example 11.7 Let

$$G_4 = (\{a,b,c,d,e\}, \{a \to a, b \to ba, c \to cbb, d \to da, e \to cbbd\}, e).$$

By definition, $L_0(G_4) = \{e\}$.

We now prove that, for $n \geq 1$,

$$L_n(G_4) = \{cbb(ba)^2(ba^2)^2 \dots (ba^{n-1})^2 da^{n-1}\}.$$

Because there is only one production for e, we only have the derivation $e \Longrightarrow cbbd$. Therefore $L_1(G_4) = \{cbbd\}$ which proves the basis. Furthermore,

$$\begin{aligned} cbb(ba)^2(ba^2)^2 \dots (ba^{n-1})^2 da^{n-1} &\Longrightarrow cbbbaba(baa)^2(baa^2)^2 \dots (baa^{n-1})^2 daa^{n-1} \\ &= cbb(ba)^2(ba^2)^2 \dots (ba^n)^2 da^n \end{aligned}$$

is the only one step derivation with left hand side $cbb(ba)^2(ba^2)^2 \dots (ba^{n-1})^2 da^{n-1}$. Thus the induction step is shown, too.

Hence we get

$$L(G_4) = \{e\} \cup \{cbbbababa^2ba^2 \dots ba^n ba^n da^n \mid n \geq 0\}.$$

Example 11.8 We consider the 0L system

$$G_5 = (\{a,b,c\}, \{a \to a^2, b \to ab, c \to bc, c \to c\}, abc).$$

We now prove that

$$\begin{aligned} L(G_5) &= \{a^{2^n-1}ba^{2^{n_1}-1}ba^{2^{n_2}-1}b \dots a^{2^{n_r}-1}bbc \mid n > n_1 > n_2 > \dots > n_r \geq 1, r > 0, n \geq 2\} \\ &\quad \cup \{a^{2^n-1}ba^{2^{n_1}-1}ba^{2^{n_2}-1}b \dots a^{2^{n_r}-1}bc \mid n > n_1 > n_2 > \dots > n_r \geq 1, r \geq 0, n \geq 1\}. \end{aligned}$$

Let

$$\begin{aligned} w_{n,n_1,n_2,\dots,n_r} &= a^{2^n-1}ba^{2^{n_1}-1}ba^{2^{n_2}-1}b \dots a^{2^{n_r}-1}bbc, \;\; n \geq 2, \\ w'_{n,n_1,n_2,\dots,n_r} &= a^{2^n-1}ba^{2^{n_1}-1}ba^{2^{n_2}-1}b \dots a^{2^{n_r}-1}bc, \;\; n \geq 1. \end{aligned}$$

Applying $c \to bc$ or $c \to c$, we only get the derivations

$$\begin{aligned} &w_{n,n_1,n_2,\dots,n_r} \Longrightarrow w_{n+1,n_1+1,n_2+1,\dots,n_r+1,1} \text{ and } w_{n,n_1,n_2,\dots,n_r} \Longrightarrow w'_{n+1,n_1+1,n_2+1,\dots,n_r+1,1}\,, \\ &w'_{n,n_1,n_2,\dots,n_r} \Longrightarrow w_{n+1,n_1+1,n_2+1,\dots,n_r+1} \text{ and } w'_{n,n_1,n_2,\dots,n_r} \Longrightarrow w'_{n+1,n_1+1,n_2+1,\dots,n_r+1}\,. \end{aligned}$$

Since the start word is w'_1, we can only generate words of the form $w_{n,n_1,n_2,\dots,n_r}$ or $w'_{n,n_1,n_2,\dots,n_r}$.

It remains to prove that we can obtain all these words. We prove this by induction on the sum $s = n + n_1 + n_2 + \dots + n_r$. If $s = 1$ (i.e., $n = 1$ and $r = 0$), then we have to generate the start word $w'_1 = abc$. We consider two cases:

Case 1: $w_{n,n_1,n_2,\dots,n_r}$, $n_r \geq 2$.
Then $w'_{n-1,n_1-1,n_2-1,\dots,n_r-1} \in L(G_5)$ by induction and $w'_{n-1,n_1-1,n_2-1,\dots,n_r-1} \Longrightarrow w_{n,n_1,n_2,\dots,n_r}$. Therefore $w_{n,n_1,n_2,\dots,n_r} \in L(G_5)$.

Case 2: $w_{n,n_1,n_2,\dots,n_{r-1},1}$.
Then $n_{r-1} \geq 2$ and $w_{n-1,n_1-1,n_2-1,\dots,n_{r-1}-1} \in L(G_5)$ by induction. Because we have the derivation $w_{n-1,n_1-1,n_2-1,\dots,n_{r-1}-1} \Longrightarrow w_{n,n_1,n_2,\dots,n_{r-1},1}$, we get $w_{n,n_1,n_2,\dots,n_r} \in L(G_5)$.

Thus we can obtain all words of the form $w_{n,n_1,n_2,\dots,n_r}$ with $r \geq 1$. Analogously, we can prove that all words of the forms $w'_{n,n_1,n_2,\dots,n_r}$ with $r \geq 1$, w_n and w'_n can be generated.

Example 11.9 We consider the 0L system $G_6 = (\{a,b,c,d,e,f\},\ P_6,\ a)$ with

$$P_6 = \{a \to dabc,\ a \to f,\ a \to e,\ b \to bc,\ c \to \lambda,\ d \to e,\ e \to e,\ f \to f^2\}.$$

It is easy to see that

$$\begin{aligned} L(G_6) &= \{a,e\} \cup \{e^{n-1}da(bc)^n \mid n \geq 1\} \cup \{e^{n+1}(bc)^n \mid n \geq 1\} \\ &\qquad \cup \{e^n f^{2^m}(bc)^n \mid n \geq 1, m \geq 0\} \cup \{f^{2^n} \mid n \geq 0\} \end{aligned}$$

because, for $n \geq 1$ and $m \geq 0$, we have only the following derivations

$$\begin{aligned} &a \Longrightarrow dabc,\ a \Longrightarrow e,\ a \Longrightarrow f, \\ &e^{n-1}da(bc)^n \Longrightarrow e^n da(bc)^{n-1},\ e^{n-1}da(bc)^n \Longrightarrow e^{n+1}(bc)^n,\ e^{n-1}da(bc)^n \Longrightarrow e^n f(bc)^n, \\ &e^{n+1}(bc)^n \Longrightarrow e^{n+1}(bc)^n,\ e^n f^{2^m}(bc)^n \Longrightarrow e^n f^{2^{m+1}}(bc)^n, \text{ and } f^{2^m} \Longrightarrow f^{2^{m+1}}. \end{aligned}$$

Giving the above definitions we followed the method to define phrase structure grammars and their languages. However, we can give a alternative definition of 0L systems based on algebraic concepts.

Let $G = (V,P,\omega)$ be a 0L system. Then we define the substitution $\sigma_G : V^* \to 2^{V^*}$ by

$$\sigma_G(a) = \{w \mid a \to w \in P\}.$$

Then it follows that

$$x \Longrightarrow_G y \text{ if and only if } y \in \sigma_G(x)$$

because in both cases we replace all letters x_i of x by an element of $\sigma_G(x_i)$. Consequently we get

$$\begin{aligned} L_0(G) &= \{\omega\} = \sigma_G^0(\omega)\,, \\ L_1(G) &= \sigma_G(\omega) = \sigma_G^1(\omega)\,, \\ L_2(G) &= \sigma_G(L_1(G)) = \sigma_G(\sigma_G(\omega)) = \sigma_G^2(\omega) \end{aligned}$$

and, by induction,

$$L_n(G) = \sigma_G^n(\omega).$$

This implies

$$L(G) = \bigcup_{n \geq 0} \sigma_G^n(\omega).$$

We now define some special cases.

Definition 11.10 *i) A 0L system $G = (V, P, \omega)$ is called propagating (P0L system, for short) if $a \to w \in P$ implies $w \neq \lambda$.*

ii) A 0L system $G = (V, P, \omega)$ is called deterministic (D0L system, for short) if, for any $a \in V$, $a \to w \in P$ and $a \to v \in P$ imply $w = v$.

iii) A PD0L system is a 0L system which is propagating as well as deterministic.

In Figure 11.5 we summarize to which special cases the grammars of our examples belong.

grammar	PD0L	D0L	P0L
G_1	+	+	+
G_2	−	+	−
G_3	−	−	+
G_4	+	+	+
G_5	−	−	+
G_6	−	−	−

Figure 11.5: A + or a – in the intersection of the row associated with G and the column associated with X indicates that G is an X system or G is not an X system.

Let $X \in \{0L,\ P0L, D0L, PD0L\}$. If L is a language such that $L = L(G)$ for some X system G, then we say that L is an X language. Moreover, by $\mathcal{L}(X)$ we denote the family of all languages generated by X systems. Thus we get the families $\mathcal{L}(PD0L)$, $\mathcal{L}(D0L)$, $\mathcal{L}(P0L)$ and $\mathcal{L}(0L)$ of all PD0L, all D0L, all P0L and all 0L languages, respectively.

11.1.3 The Basic Hierarchy

We start with two lemmas which show that without loss of generality we can assume that derivations of the empty word have a bounded length and that the length of intermediate words in a derivation of x can be bounded linearly in the length of x.

Lemma 11.11 *Let $G = (V, P, \omega)$ be a 0L system with $n = \#(V)$. For $a \in V$, let $G_a = (V, P, a)$. If $\lambda \in L(G_a)$, then $\lambda \in L_m(G_a)$ for some $m \leq n$.*

Proof. We define L_r as the set of all letters $a \in V$ such that $\lambda \in L_m(G_a)$ for some $m \leq r$. Obviously, if $a \in L_r$ then $a \in L_{r+1}$, too. Thus we have $L_r \subseteq L_{r+1}$ for $r \geq 1$.

Let $L_r = L_{r+1}$. Further let $a \in L_{r+2}$. Then there is a derivation

$$a \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \ldots \Longrightarrow w_s = \lambda$$

with $s \leq r+2$. If $s < r+2$, then $a \in L_{r+1}$. Let $s = r+2$. Then $b \in L_{r+1}$ for any letter b which occurs in w_1. By our assumption, $b \in L_r$ for any b occurring in w_1. Hence there is a derivation

$$a \Longrightarrow w_1 \Longrightarrow v_2 \Longrightarrow v_3 \Longrightarrow \ldots \Longrightarrow v_{r-1} \Longrightarrow \lambda\,.$$

This implies $a \in L_{r+1}$. Therefore in both cases we have shown that $a \in L_{r+1}$. This gives $L_{r+2} \subseteq L_{r+1}$ which proves $L_{r+2} = L_{r+1} = L_r$. By induction we can show that $L_{r+k} = L_r$ for all $k \geq 1$.

Since $L_i \subseteq V$, there is a number t such that $1 \le t \le n$ and

$$L_1 \subset L_2 \subset L_3 \subset \cdots \subset L_{t-1} \subset L_t = L_{t+1} = L_{t+2} = \dots .$$

Because L_t is a subset of V, t is smaller than the number of letters of V. Therefore $t \le n$.

Now assume that $\lambda \in L(G_a)$, then $a \in L_t$ and thus $\lambda \in L_m(G_a)$ for some $m \le t \le n$. □

Lemma 11.12 *Let $G = (V, P, \omega)$ be a 0L system. Then there exists a constant C_G such that, for any word $x \in L(G)$, there is a derivation*

$$\omega = w_0 \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \dots \Longrightarrow w_r = x$$

with $|w_i| \le C_G \cdot (|x| + 1)$ for $0 \le i \le r$.

Proof. Let

$$\begin{aligned} n &= \#(V)\,, \\ k &= \max\{|w| \mid a \to w \in P\}\,, \\ l &= \max\{|z| \mid z \in L_m(G) \text{ for } m \le n\}\,, \\ C_G &= \max\{k^n, l\}\,. \end{aligned}$$

Let $x \in L(G)$. Then $x \in L_r(G)$ for some $r \ge 0$. Let

$$\omega = w_0 \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \dots \Longrightarrow w_r = x.$$

Assume that there is some letter $a \in V$ in the word w_j, $1 \le j \le r$, such that the subderivation from a yields the empty word. Then we substitute this subderivation by a derivation of λ which has at most n steps. Such a derivation exists by Lemma 11.11. This procedure is done as long the derivation contains subderivations of the empty word with more than n steps. As a result we obtain a derivation

$$\omega = v_0 \Longrightarrow v_1 \Longrightarrow v_2 \Longrightarrow \dots \Longrightarrow v_s = x$$

with $s \le r$. We now prove that $|v_i| \le C_G(|x| + 1)$.

If $i \le n$, then $v_i \in L_i(G)$ and therefore $|v_i| \le l \le C_G \le C_G(|x| + 1)$ which proves the statement of the theorem.

If $i \ge n$, then we consider the word $v_{i-n} = u_1 u_2 \dots u_t$ where $u_j \in V$ for $1 \le j \le t$. Then $x = u_1' u_2' \dots u_t'$ where $u_j' \ne \lambda$ is obtained from u_j or $u_j' = \lambda$ (if from u_j a subderivation starts which yields the empty word). Let h be the number of letters u_j of v_{i-n} such that $u_j' \ne \lambda$. Then $h \le |x|$. Moreover, since the subderivations giving λ are finished after n derivation steps by our construction v_i is build from the words u_j'' which are obtained after n steps from u_j. By definition of k we have $|u_j''| \le k^n$ and hence

$$|v_i| \le hk^n \le C_G(|x| + 1).$$

This proves the theorem. □

We now compare the families generated by Lindenmayer systems with each other and with the families of the Chomsky hierarchy.

$$
\begin{tikzcd}
 & & \mathcal{L}(CS) & \\
 & \mathcal{L}(0L) \arrow[ur] & & \mathcal{L}(CF) \arrow[ul] \\
\mathcal{L}(P0L) \arrow[ur] & & \mathcal{L}(D0L) \arrow[ul] & \mathcal{L}(REG) \arrow[u] \\
 & \mathcal{L}(PD0L) \arrow[ul] \arrow[ur] & & \mathcal{L}(FIN) \arrow[u]
\end{tikzcd}
$$

Figure 11.6: $\mathcal{L}_1 \to \mathcal{L}_2$ denotes a proper inclusion of $\mathcal{L}_1$ in $\mathcal{L}_2$. If two families are not connected by arrows, then they are incomparable.

Theorem 11.13 *The diagram of Figure 11.6 holds.*

Proof. The part $\mathcal{L}(FIN) \subset \mathcal{L}(REG) \subset \mathcal{L}(CF) \subset \mathcal{L}(CS)$ is well-known as a part of the Chomsky hierarchy (see Theorem 2.37).

Since any PD0L system is a P0L system, too, it follows that $\mathcal{L}(PD0L) \subseteq \mathcal{L}(P0L)$. Analogously we obtain the other inclusions between $\mathcal{L}(PD0L)$, $\mathcal{L}(P0L)$, $\mathcal{L}(D0L)$ and $\mathcal{L}(0L)$.

In order to prove the strictness of the inclusions it is sufficient to prove the existence of languages L_1 and L_2 such that

$$L_1 \in \mathcal{L}(P0L),\ L_1 \notin \mathcal{L}(D0L) \text{ and } L_2 \in \mathcal{L}(D0L),\ L_2 \notin \mathcal{L}(P0L).$$

Then $L_1 \in \mathcal{L}(P0L) \setminus \mathcal{L}(PD0L)$ and $L_1 \in \mathcal{L}(0L) \setminus \mathcal{L}(D0L)$ which proves the properness of two inclusions. L_2 can be used to show the strictness of the other two inclusions.

We consider $L_1 = \{a\}^+$. Because $L_1 = L(G_3)$ for the P0L system G_3 from Example 11.6, $L_1 \in \mathcal{L}(P0L)$ by definition. Let us assume that $L_1 \in \mathcal{L}(D0L)$. Then there is a D0L system $G = (\{a\}, \{a \to a^r\}, a^s)$ with $L(G) = L_1$. Since

$$a^s \Longrightarrow a^{sr} \Longrightarrow a^{sr^2} \Longrightarrow a^{sr^3} \Longrightarrow \ldots \Longrightarrow a^{sr^k} \Longrightarrow \ldots$$

is the only derivation in G, we get $L(G) = \{a^{sr^n} \mid n \geq 0\}$. If $r = 1$, then $L(G) = \{a^s\}$ which contradicts $L_1 = L(G)$. If $r \geq 2$, then $sr \leq sr + 1 \leq sr^2$. Hence $a^{sr+1} \in L_1$, but $a^{sr+1} \notin L(G)$ in contrast to $L(G) = L_1$.

Hence $L_1 \notin \mathcal{L}(D0L)$.

Let $L_2 = \{aab, ab\}$. By $L_2 = L(G_2)$ for the D0L system G_2 of Example 11.5, we have $L_2 \in \mathcal{L}(D0L)$. If $L_2 \in \mathcal{L}(P0L)$, then $L(G') = L_2$ for some P0L system $G' = (\{a, b\}, P, \omega)$. By the completeness we have rules $a \to w_a$ and $b \to w_b$ in P. Then we obtain $aab \Longrightarrow w_a w_a w_b$ and $w_a w_a w_b \in L(G') = L_2$. Since G' is propagating, w_a and w_b are non-empty words which implies that the length of $w_a w_a w_b$ is at least 3. Therefore $w_a = a$ and $w_b = b$. Thus $aab \Longrightarrow_{G'} aab$ and $ab \Longrightarrow_{G'} ab$ are the only direct derivation steps. This implies $L(G') = \{\omega\}$, i. e., $L(G')$ consists of one word, which contradicts $L(G') = L_2$ since L_2 contains two words. Hence $L_2 \notin \mathcal{L}(P0L)$.

Let $X \in \{DP0L, P0L, D0L, 0L\}$ and $Y \in \{FIN, REG, CF\}$. In order to prove that $\mathcal{L}(X)$ and $\mathcal{L}(Y)$ are incomparable, it is sufficient to present languages

$$L_3 \in \mathcal{L}(FIN),\ L_3 \notin \mathcal{L}(0L) \text{ and } L_4 \in \mathcal{L}(PD0L),\ L_4 \notin \mathcal{L}(CF).$$

We choose $L_3 = \{a^2, a^4\}$. Obviously, $L_3 \in \mathcal{L}(FIN)$. If $L_3 \in \mathcal{L}(0L)$, then there is a 0L system $H = (\{a\}, P, \omega)$ with $L_3 = L(H)$. Let $a \to w_a \in P$. Then we get $a^4 \Longrightarrow (w_a)^4 \in L_3$. If $|w_a| \leq 2$, then $|(w_a)^4| \geq 8$. Therefore $(w_a)^4 \notin L_3$ which contradicts $L(H) = L_3$. Thus the possible rules are $a \to a$ and $a \to \lambda$. We consider the three following possible cases.

Case 1. $P = \{a \to a\}$.
Then $\omega \Longrightarrow \omega$ holds, which yields $L(H) = \{\omega\}$ in contrast to $L(H) = L_3$.

Case 2. $P = \{a \to \lambda\}$.
Then $\omega \Longrightarrow \lambda$ holds which gives $L(H) = \{\omega, \lambda\}$ in contrast to the choice of H.

Case 3. $P = \{a \to a, a \to \lambda\}$.
Then $a^4 = aaaa \Longrightarrow aaa\lambda = a^3$. This implies $a^3 \in L(H)$ which contradicts $L(H) = L_3$.

Therefore $L_3 \notin \mathcal{L}(0L)$.

Let $L_4 = \{a^{2^n} \mid n \geq 2\}$. By Example 11.4, $L_4 = L(G_1)$ for the PD0L system G_1 and thus $L_4 \in \mathcal{L}(PD0L)$. On the other hand, $L_4 \notin \mathcal{L}(CF)$ is not a semi-linear language, and hence it is not context-free.

Let $G = (V, P, \omega)$ be a 0L system. We construct a phrase structure grammar $H = (N, V, P', S)$ with $L(H) = L(G)$ as follows. We set $N = \{A, B, C, D, E\}$ and define P' as the set of all rules of the following types:

a) $S \to AD\omega B$,
b) $AD \to AC$, $AD \to AE$,
c) $Ca \to wC$ for $a \to w \in P$ and $CB \to DB$,
d) $aD \to Da$ for $a \in V$,
e) $AEa \to aE$ and $Ea \to aE$ for $a \in V$, $EB \to \lambda$,
f) $S \to \lambda$, if $\lambda \in L(G)$.

Any derivation in H starts with $S \underset{H}{\Longrightarrow} AD\omega B$. Let us now assume that we have generated a word $ADa_1a_2 \dots a_nB$ with $a_i \in V$ for $1 \leq i \leq n$. Now we can only apply the rules of type b) and we obtain $ACa_1a_2 \dots a_nB$ or $AEa_1a_2 \dots a_nB$.

In the former case we have to continue with rules of type c) which gives

$$\begin{aligned} ACa_1a_2 \dots a_nB &\underset{H}{\Longrightarrow} Aw_1Ca_2 \dots a_nB \underset{H}{\Longrightarrow} \dots \underset{H}{\Longrightarrow} Aw_1w_2 \dots w_nCB \\ &\underset{H}{\Longrightarrow} Aw_1w_2 \dots w_nDB \end{aligned}$$

where $a_i \to w_i \in P$ for $1 \leq i \leq n$. By rules of type d) we shift the letter D to the left and obtain $ADw_1w_2 \dots w_nB$. That is, we have obtained a word of the form we start with and – besides the nonterminals – we have simulated the derivation $a_1a_2 \dots a_n \underset{G}{\Longrightarrow} w_1w_2 \dots w_n$.

We consider the latter case. If $n = 0$, we get AEB in H and the derivation cannot be terminated (the only applicable rule $EB \to \lambda$ gives A and the derivation is now blocked). If $n \geq 1$, we get the derivation

$$AEa_1a_2 \dots a_nB \underset{H}{\Longrightarrow} a_1Ea_2 \dots a_nB \underset{H}{\Longrightarrow} \dots \underset{H}{\Longrightarrow} a_1a_2 \dots a_nEB \Longrightarrow_H a_1a_2 \dots a_n.$$

That is, we only delete the nonterminals. (Note that other derivations are not possible since the application of $Ea_1 \to a_1E$ would not delete the letter A which remains which means that the derivation cannot terminate.)

Therefore, for any derivation

$$\omega \underset{G}{\Longrightarrow} v_1 \underset{G}{\Longrightarrow} v_2 \underset{G}{\Longrightarrow} \ldots \underset{G}{\Longrightarrow} v_n$$

with $v_n \neq \lambda$ in G, there is a derivation

$$S \underset{H}{\Longrightarrow} AD\omega B \overset{*}{\underset{H}{\Longrightarrow}} ADv_1B \overset{*}{\underset{H}{\Longrightarrow}} ADv_2B \overset{*}{\underset{H}{\Longrightarrow}} \ldots \overset{*}{\underset{H}{\Longrightarrow}} ADv_nB \overset{*}{\underset{H}{\Longrightarrow}} AEv_nB \overset{*}{\underset{H}{\Longrightarrow}} v_n$$

and conversely. Thus the languages $L(G)$ and $L(H)$ coincide in the non-empty words. If $\lambda \in L(G)$, then $\lambda \in L(H)$ by application of the rule f). If $\lambda \notin L(G)$, then H does not contain the rule f) and thus $\lambda \notin L(H)$. Hence we have $L(G) = L(H)$.

Let x be a non-empty word of $L(G)$. By Lemma 11.12, there is a derivation of x such that any intermediate word has a length bounded by $C_G(|x| + 1)$. Therefore all the intermediate words of the corresponding derivation of x in H have at most the length $C_G(|x| + 1) + 3 \leq C|x|$ for an appropriate constant C. By the workspace theorem (see Theorem 3.22), the language of all non-empty words of $L(H)$ is context-sensitive. □

11.1.4 Adult Languages

The language generated by a 0L system consists of all words which can be derived from the start word. Thus it takes into consideration all phases of the development, e. g. in case of a flower the "green" phase, which produces the handle or stem and the leaves, as well as the "flowering" phase, where the blossom is build and the parts of the "green" phase are not changed. Especially, one is interested in the final stages or adult stages which are not changed (at least for a long time).

Modelling this aspect within the framework of 0L systems we are interested in those strings w which belong to the language and only allow the derivation $w \Longrightarrow w$. This leads to the following definition.

Definition 11.14 *The* adult language $L_A(G)$ *of a 0L system G is the set of all words $w \in L(G)$ such that $w \Longrightarrow v$ implies $w = v$.*

By definition, the empty word cannot be contained in an adult language.

The adult alphabet $V_A(G)$ is set of all letters of V, which occur in words of $L_A(G)$.

Let $X \in \{0L, P0L, D0L, PD0L\}$. By $\mathcal{L}(AX)$ we denote the family of adult languages generated by X systems.

Example 11.15 We consider the 0L system G_6 from Example 11.9 Since

$$L(G_6) = \{a, e\} \cup \{e^{n-1}da(bc)^n \mid n \geq 1\} \cup \{e^n(bc)^n \mid n \geq 1\} \cup \{e^n f^{2^m}(bc)^n \mid n \geq 1, m \geq 0\},$$

we obtain the adult language

$$L_A(G_6) = \{e\} \cup \{e^n(bc)^n \mid n \geq 1\}.$$

Note that the adult language of G_6 is a context-free language whereas $L(G_6)$ is not context-free.

The aim of this section is to show that the fact seen in the example holds in general, i. e., any adult language of a 0L system is a context-free language, and conversely, any context-free language is an adult language of some 0L system.

Theorem 11.16 *i) For any context-free grammar G such that $\lambda \notin L(G)$, there is a propagating 0L system G' such that $L_A(G') = L(G)$. ii) For any context-free grammar G, there is a 0L system G' such that $L_A(G') = L(G)$.*

Proof. i) Let $G = (N, T, P, S)$ be a context-free grammar with $\lambda \notin L(G)$. First we construct a context-free grammar $G'' = (N'', T, P'', S'')$ in Chomsky normal form such that $L(G'') = L(G)$ (see Theorem 2.26). Since $\lambda \notin L(G)$, P'' contains no rule of the form $A \to \lambda$ with $A \in N''$. We define the 0L system G' by

$$G' = (N'' \cup T, P'' \cup \{a \to a \mid a \in N'' \cup T\}, S''). \tag{11.2}$$

Since P'' contains no erasing rule, G' is propagating.

Let $x \underset{G''}{\Longrightarrow} y$. Then $x = u_1 A u_2$ and $y = u_1 w u_2$ for some rule $A \to w \in P''$. Then we also have $x \underset{G'}{\Longrightarrow} y$ by applying $a \to a$ to all letters of u_1 and u_2 and $A \to w \in P''$ to the distinguished occurrence of A in x. Thus any sentential form of G'' is contained in $L(G')$.

Moreover, if $x \Longrightarrow_{G'} y$, then $x = x_1 x_2 \dots x_n$, $y = y_1 y_2 \dots y_n$ and $x_i \to y_i \in P'$ for $1 \leq i \leq n$. Let $M = \{i_1, i_2, \dots, i_r\}$ be the subset of $\{1, 2, \dots, n\}$ such that $x_i \to y_i \neq x_i$ for $i \in M$ and $x_j \to x_j$ for $j \in \{1, 2, \dots, n\} \setminus M$. Then $i \in M$ implies $x_i \in N''$ and we have in G'' the derivation

$$\begin{array}{rcl}
x & = & u_1 x_{i_1} u_2 x_{i_2} \dots u_r x_{i_i} u_{r+1} \\
 & \underset{G''}{\Longrightarrow} & u_1 y_{i_1} u_2 x_{i_2} \dots u_r x_{i_i} u_{r+1} \\
 & \underset{G''}{\Longrightarrow} & u_1 y_{i_1} u_2 y_{i_2} u_3 x_{i_3} \dots u_r x_{i_i} u_{r+1} \\
 & & \dots \\
 & \underset{G''}{\Longrightarrow} & u_1 y_{i_1} u_2 y_{i_2} \dots u_{r-1} y_{i_{r-1}} u_r x_{i_i} u_{r+1} \\
 & \underset{G''}{\Longrightarrow} & u_1 y_{i_1} u_2 y_{i_2} \dots u_{r-1} y_{i_{r-1}} u_r y_{i_i} u_{r+1} \\
 & = & y
\end{array}$$

This proves that $L(G')$ is the set of all sentential forms of G''.

If $x = u_1 A u_2$ is a sentential form of G'' with $A \in N''$, then we apply to all letters of u_1 and u_2 rules of the form $a \to a$ and to A a rule $A \to w$ with $w \neq A$ (such a rule exists since G'' is in Chomsky normal form) and obtain $u_1 w u_2 \neq x$. Hence x is not in the adult language of G'. On the other hand, if a sentential form x' only contains terminals, then we can only apply identity rules to the letters of x' which gives $x' \in L_A(G')$.

Therefore $L_A(G')$ consists of all sentential forms which only contain terminals, i. e., $L_A(G') = L(G'') = L(G)$.

ii) If $L(G)$ does not contain the empty word, we repeat the construction given above. If $L(G)$ contains the empty word, then the grammar in Chomsky normal form contains in addition the rule $S \to \lambda$ and S does not occur on the right hand side of a production, I. e., $S \to \lambda$ is only used in the derivation $S \Longrightarrow \lambda$ of the empty word. Now we repeat the above construction and add the rule $S \to \lambda$ to the production set of the 0L system. Thus the obtained system is not propagating. However, in the =L system, the additional rule can only be used to derive the empty word in one step from the axiom S, too. Hence we get the empty word which satisfies $\lambda \Longrightarrow \lambda$ in the adult language. $\square$

Lemma 11.17 *Let $G = (V, P, \omega)$ be a 0L system. Let $m = \#(V_A(G))$. For any letter $a \in V_A(G)$, let $G_a = (V, P, a)$. Then, for any $a \in V_A(G)$, one of the following two cases appears:*

- *$\lambda \in L_t(G_a)$ for some $t \leq m$ or*
- *$L_m(G_a)$ contains exactly one word z_a and, for all words $v \in V^*$, $z_a \Longrightarrow v$ implies $v = z_a$.*

Proof. Let $a \in V_A(G)$. Then there is exactly one rule $a \to w_a$ in P. Assume the contrary, i.e., $a \to w_1 \in P$ and $a \to w_2 \in P$ with $w_1 \neq w_2$. Since $a \in V_A(G)$, there is a word $x \in L_A(G)$ where a occurs in x. Let $x = x_1 a y_1$. Then have two derivation $x_1 a y_1 \Longrightarrow x_2 w_1 y_2$ and $x_1 a y_1 \Longrightarrow x_2 w_2 y_2$ with $x_2 w_1 y_2 \neq x_2 w_2 y_2$ in contrast to the property of the words of $L_A(G)$.

Since $x = x_1 a y_1 \Longrightarrow x_2 w_a y_2 = x \in L_A(G)$, all letters of w_a belong to the adult alphabet $V_A(G)$. Consequently, all words of $L(G_a)$ only contain letters of $V_A(G)$. Because there is only one rule for any letter of $V_A(G)$, there is a unique derivation in G_a. Therefore, for any $t \geq 0$, $L_t(G)$ contains at most one word.

If w_a contains two occurrences of a, then we have a derivation

$$\begin{aligned} w_a &= u_1 a u_2 a u_3 \Longrightarrow u_1' w_a u_2' w_a u_3' \\ &= u_1' u_1 a u_2 a u_3 u_2' u_1 a u_2 a u_3 u_3' \Longrightarrow u_1'' u_1' w_a u_2' w_a u_3' u_2'' u_1' w_a u_2' w_a u_3' u_3'' \\ &= u_1'' u_1' u_1 a u_2 a u_3 u_2' u_1 a u_2 a u_3 u_3' u_2'' u_1' u_1 a u_2 a u_3 u_2' u_1 a u_2 a u_3 u_3' u_3'' \\ & \quad \cdot\cdot\cdot\cdot\cdot \end{aligned}$$

and therefore from w_a we can generate a word with an arbitrarily large number of occurrences of a. Therefore, from x we can also generate a word with an arbitrarily large number of occurrences of a. However, x derives only x and x has a bounded number of occurrences of a. This contradiction shows that w_a contains at most one occurrence of a.

Now assume that w_a contains exactly one occurrence of a. Let $w_a = p_1 a q_1$. Then we have the derivation

$$x = x_1 a y_1 \Longrightarrow x_2 p_1 a q_1 y_2 \Longrightarrow x_3 p_2 p_1 a q_1 q_2 y_3 \Longrightarrow \ldots \Longrightarrow x_m p_{m-1} p_{m-2} \ldots p_1 a q_1 q_2 \ldots q_{m-1} y_m .$$

Obviously, if $p_i \neq \lambda$ or $q_i \neq \lambda$ for $i \geq 1$, then we can generate arbitrarily long words from x in contrast to $x \in L_A(G)$. If λ can be generated from p_1 and q_1, by Lemma 11.11, we have derivations

$$p_1 \Longrightarrow p_2 \Longrightarrow p_3 \Longrightarrow \ldots \Longrightarrow p_s \Longrightarrow \lambda \text{ and } q_1 \Longrightarrow q_2 \Longrightarrow q_3 \Longrightarrow \ldots \Longrightarrow q_t \Longrightarrow \lambda$$

with $s < m$ and $t < m$ and non-empty words $p_1, p_2, \ldots, p_s, q_1, q_2, \ldots, q_t$. Then we get from a in at most m steps the word $p_s p_{s-1} \ldots p_1 a q_1 q_2 \ldots q_t$ which has the property

$$p_s p_{s-1} \ldots p_1 a q_1 q_2 \ldots q_t \Longrightarrow p_s p_{s-1} \ldots p_1 a q_1 q_2 \ldots q_t.$$

Thus $z_a = p_s p_{s-1} \ldots p_1 a q_1 q_2 \ldots q_t$ has the properties required in the statement.

Now assume that w_a contains no occurrence of a. For $i \geq 1$, let z_i be the unique word in $L_i(G_a)$. We first prove that z_i has no occurrence of a for $i \geq 1$. Let x be a word of $L_A(G)$ containing a. Then $x = x_0ax_1ax_2 \dots ax_r$ with $r \geq 1$ and x_i not containing a for $0 \leq i \leq r$. If u defined by $x_0 \Longrightarrow u$ contains an occurrence of a and z_j contains a for some $j \geq 1$, $x \Longrightarrow^j u_0z_ju_1z_ju_2z_j \dots z_ju_r = x$. Because u_0 starts with u which contains a and any z_j contains an a, the generated word x contains at least $r+1$ occurrences of a which is impossible. Analogously we get a contradiction if u contains no a, i. e., all a's are contained in the word derived from $x_1ax_2a \dots ax_r$.

Now we have $x = x_1ay_1 \Longrightarrow x_2z_1y_2 = x$. Because a does not occur in $z_1 = w_a$, a has to occur in at least one of the words x_2 and y_2. We only discuss the case that a occurs in x_2, the other case can be handled analogously. Then $x = v_1av_2z_1y_2$. Now we get $x \Longrightarrow v_1'z_1v_2'z_2y_2' = x$. Because a occurs in x and $\#_a(z_1) = \#_a(z_2) = 0$, a occurs in v_1' or v_2' or y_2'. Let us assume that a occurs in y_2' (the other cases can be handled analogously). Then we obtain

$$x = v_1'z_1v_2'z_2v_3'av_4' \Longrightarrow v_1''z_2v_2''z_3v_3''z_1v_4''.$$

Continuing in this way we get that x contains all the words $z_1, z_2, z_3, \dots$ which is only possible if $z_i = \lambda$ for some i. From Lemma 11.11 we know that $\lambda \in L_t(G_a)$ for some $t \leq n$. □

Lemma 11.18 *For any 0L system G, there is a 0L system $G' = (V, P', S)$ such that $L_A(G') = L_A(G)$ and, for any $a \in V_A(G')$, the only production in P' is $a \to a$.*

Proof. Let $G = (V, P, S)$. Without loss of generality we assume that the start word of G is a letter S not belonging to the adult alphabet $V_A(G)$ (if this is not the case, we add S to the alphabet and $S \to \omega$ to the set of productions, where ω is the original start word; these additions do not change the adult language). Let $n = \#(V_A(G))$. For any letter $a \in V_A(G)$, let $G_a = (V, P, a)$.

We define the homomorphism h by

$$\begin{aligned} &h(a) = \lambda \text{ if } a \in V_A(G) \text{ and } \lambda \in L_m(G_a) \text{ for some } m \leq n\,, \\ &h(a) = z_a \text{ if } a \in V_A(G) \text{ and } z_a \in L_n(G_a)\,, \\ &h(a) = a \text{ if } a \notin V_A(G)\,. \end{aligned}$$

By Lemma 11.17 the homomorphism is well defined. We set $G' = (V, P', S)$ with

$$P' = \{a \to h(w) \mid a \to w \in P \text{ and } a \notin V_A(G)\} \cup \{a \to a \mid a \in V_A(G)\}.$$

One can easily prove by induction on the number i of derivation steps that

$$S \Longrightarrow_G^i x \text{ if and only if } S \Longrightarrow_{G'}^i h(x)\,. \tag{11.3}$$

Now let $x = x_1x_2 \dots x_m$ be a word of $L_A(G)$ with $x_i \in V_A(G)$ for $1 \leq i \leq m$. Then $x_1x_2 \dots x_m \Longrightarrow_{G'} x_1x_2 \dots x_m$ by the definition of P', and this is the only possible derivation from x. Therefore $x \in L_A(G')$. Hence $L_A(G) \subseteq L_A(G')$.

Now let w be a word of $L_A(G')$. By (11.3), there is a word w' such that $w' \in L(G)$ and $w = h(w')$. We assume that $w' = w_0B_1w_1B_2w_2 \dots w_{t-1}B_tw_t$ with $t \geq 0$, $w_i \in V_A(G)^*$

for $0 \le i \le t$ and $B_j \notin V_A(G)$ for $1 \le j \le t$. For $0 \le i \le t$, we define z_i as follows. If $w_i = \lambda$, then we set $z_i = \lambda$. If $w_i \ne \lambda$, then w_i consists only of letters $V_A(G)$. Then we define z_i as the only word which can be generated from w_i in m steps in G. Therefore

$$h(w_i) = z_i \quad \text{and} \quad z_i \Longrightarrow_G z_i \tag{11.4}$$

for $1 \le i \le n$. Then

$$h(w') = z_0 B_1 z_1 B_2 z_2 \dots z_{t-1} B_t z_t = w\,.$$

For $0 \le i \le t$, by the definition of P', if $z_i \ne \lambda$,

$$z_i \Longrightarrow_{G'} z_i \tag{11.5}$$

holds, because all letters of z_i belong to $V_A(G)$. Let $B_r B_{r+1} \dots B_s$ be a subword of w only consisting of letters not in $V_A(G)$, i.e $z_r = z_r + 1 = \dots z_{s-1} = \lambda$, and the letters before B_r and after B_s in w – if they exist – are from $V_A(G)$. Since $w \in L_A(G')$, we have $w \Longrightarrow_{G'} w$. Taking into consideration (11.5) we get

$$B_r B_{r+1} \dots B_s \Longrightarrow_{G'} B_r B_{r+1} \dots B_s\,. \tag{11.6}$$

For $r \le i \le s$, if $B_i \to y_i$ in P, then $B_i \to h(y_i) \in P'$. Thus

$$B_r B_{r+1} \dots B_s \Longrightarrow_{G'} h(y_r) h(y_{r+1}) \dots h(y_s)\,.$$

By (11.6),

$$B_r B_{r+1} \dots B_s = h(y_r) h(y_{r+1}) \dots h(y_s)\,. \tag{11.7}$$

Therefore, for $r \le i \le s$, $h(y_i)$ contains only letters not in $V_A(G)$, which implies $h(y_i) = y_i$ by the definition of h (since letters of $V_A(G)$ occur in $h(y_i)$ otherwise). By (11.7) this yields

$$B_r B_{r+1} \dots B_s \Longrightarrow_G y_r y_{r+1} \dots y_s = h(y_r) h(y_{r+1}) \dots h(y_s) = B_r B_{r+1} \dots B_s\,.$$

If we combine this relation with (11.4) we get $w \Longrightarrow_G w$ is the only derivation for $w = h(w')$ which proves that $w \in L_A(G)$. Thus we obtain $L_A(G') \subseteq L_A(G)$. □

We note that we obtain a context-free grammar without rules of the form $A \to \lambda$ if we start from a propagating system. Therefore, for any propagating system G there is a context-free grammar H such that $L(H) = L_A(G)$ and $L(H)$ does not contain the empty word.

Theorem 11.19 *i) For any 0L system G, there is a context-free grammar G'' such that $L(G'') = L_A(G)$.*

Proof. First, for G, we consider the 0L system $G' = (V, P', S)$ according to Lemma 11.18, i. e., $a \to a$ is the only rule for $a \in V_A(G')$ and $L_A(G') = L_A(G)$. Then we construct the context-free grammar

$$G'' = (V \setminus V_A(G'), V_A(G'), P' \setminus \{a \to a \mid a \in V_A(G')\}, S).$$

It is easy to show that $L(G'') = L_A(G')$. □

Theorem 11.20 *i)* $\mathcal{L}(A0L) = \mathcal{L}(CF)$ *and* $\mathcal{L}(AP0L) = \{L \mid L \in \mathcal{L}(CF),\ \lambda \notin L\}$.

ii) The family $\mathcal{L}(AD0L)$ *consists the empty set and all languages* $\{w\}$ *for some word* w, *and* $\mathcal{L}(APD0L)$ *consists the empty set and all languages* $\{w\}$ *for some non-empty word* w.

Proof. i) By Theorem 11.16, we get that $\mathcal{L}(CF) \subseteq \mathcal{L}(AP0L)$. Theorem 11.19 yields the converse inclusion $\mathcal{L}(A0L) \subseteq \mathcal{L}(CF)$. If we consider propagating systems, in both statements we are restricted to languages without the empty word.

ii) If $G = (V, P, \omega)$ is a D0L system, then we have only one derivation

$$\omega = w_0 \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow w_3 \Longrightarrow \dots .$$

If there is an i such that $w_i = w_{i+1}$, then we have $L_A(G) = \{w_i\}$. If $w_i \neq w_{i+1}$ for any i, then $L_A(G) = \emptyset$. Thus the adult language of a deterministic 0L system is empty or contains exactly one word.

In the propagating case, obviously, the empty word as the only word of the adult language is impossible.

We now show that all singletons and the empty language occur as adult languages of propagating D0L systems. Let w be a non-empty word over some alphabet V. Then we consider the D0L system $G = (V, \{a \to a \mid a \in V\}, w)$. Obviously, $w \Longrightarrow w \Longrightarrow w \Longrightarrow \dots$ is the only derivation in G. Thus $L_A(G) = \{w\}$. For the D0L system $G' = (V, \{a \to \lambda \mid a \in V\}, w)$, we have the unique derivation $w \Longrightarrow \lambda \Longrightarrow \lambda \Longrightarrow \dots$, i.e., $L_A(G') = \{\lambda\}$. Furthermore, for G_1 Example 11.4, $L_A(G_1)$ is empty. □

We note that $\mathcal{L}(AD0L)$ is a proper subfamily of the family of all finite languages.

11.1.5 Decision Problems

In this section we want to discuss the decidability status of the classical decision problems considered in the theory of formal languages for interactionless Lindenmayer systems. For $X \in \{0L, P0L, D0L, PD0L\}$, we regard the following problems.

Membership problem:	Given X system $G = (V, P, \omega)$ and $w \in V^*$, decide whether or not $w \in L(G)$.
Finiteness problem:	Given X system $G = (V, P, \omega)$, decide whether or not $L(G)$ is finite.
Equivalence problem:	Given X systems $G = (V, P, \omega)$ and $H = (V, P', \omega')$, decide whether or not $L(G) = L(H)$.

For the sequential grammars of the Chomsky hierarchy, we have also considered the emptiness problem. This is not of interest for Lindenmayer systems, because any language generated by a Lindenmayer system $G = (V, P, \omega)$ contains at least the axiom ω and is therefore not empty.

We mention that the membership problem and the finiteness problem have some biological relevance. Let us assume that we have a 0L system G which we want use as a model for the development of some filamentous organism or alga etc. Usually such a model is obtained by an analysis of the first steps of the development of the biological object. Now the membership problem is the question whether or not a later stage of the development can be got by the model G. The finiteness problem has an negative answer if and only if the development does not be finished after a certain time and at least one branch of the development produces larger and larger plants. By such an interpretation, there is an interest from a biological point of view in these questions. The equivalence problem is the test whether or not two given models describe the same development.

However, one has to note that in biology one is more interested in the sequences of the stages of the development instead of the set of all stages. In this lecture we shall only consider the language theoretic part, for a discussion of decidability for sequences instead of languages we refer to [11] and [27].

By Theorem 11.13, $\mathcal{L}(0L)$ is contained in $\mathcal{L}(CS)$. If we consider the proof we see that, for a given 0L system G, we can construct a context-sensitive grammar H with $L(G) = L(H)$. It is known that it is decidable, whether or not a given word w belongs to the language generated by a given context-sensitive grammar. Thus the membership for 0L systems is decidable, too. However, all known algorithms for the membership problem for context-sensitive grammar have at least exponential time complexity.

By Lemma 11.12, there is a natural algorithm. We construct all derivations which only contain words of length $C_G|w|$. If w is obtained the answer to the membership is "yes", otherwise $w \notin L(G)$ holds. However, this algorithm is also exponential in time because we have to consider exponentially many derivation steps.

In [25] it has been shown that the Cocke-Younger-Kasami algorithm known for context-free grammars can be translated to interactionless 0L systems. However, by the parallelism in the derivation in 0L system we get the complexity $O(|w|^4)$ for a fixed 0L system (in the case of context-free grammars, the complexity is $O(|w|^3)$.

Theorem 11.21 *For 0L systems, the membership problem is decidable in polynomial time* $O(|w|^4)$. □

Theorem 11.22 *For 0L systems, the finiteness problem is decidable in polynomial time.*

Proof. Let $G = (V, P, \omega)$ be a 0L system. Let $n = \#(V)$. For any letter $a \in V$, we set $G_a = (V, P, a)$.

We call a letter $a \in V$ *surviving* in G iff $L_i(G_a)$ contains a non-empty word for any $i \geq 0$. Let $V_s(G)$ be the set of all surviving letters of G. (Note that $V = V_s(G)$ for a propagating 0L system.)

We construct the directed graph $H_G = (V, E)$ where the set of vertices coincides with the set of all letters of V and $(a, b) \in E$ if and only if there is a production $a \to x_1 b x_2 \in P$ with $x_1, x_2 \in V^*$.

Claim 1: a is surviving if and only if there is an infinite path in H_G which starts in a.

Let H_G contain an infinite path starting in a. Let b be the letter which is obtained by the beginning of length i of the this path. Then, by the definition of H_G, there is a word in $L_i(G)$ which contains b. This shows the non-emptiness of $L_i(G_a)$ for any $i \geq 0$.

Conversely, let us assume that a is surviving. Then $L_{n+1}(G_a)$ contains a non-empty word w. Let b be a letter of w. Then there is a derivation

$$a = a_0 \Longrightarrow u_1 a_1 v_1 \Longrightarrow u_2 a_2 v_2 \Longrightarrow \ldots \Longrightarrow u_{n+1} a_{n+1} v_{n+1} = w$$

such that $a_{n+1} = b$ and $(a_i, a_{i+1}) \in E$ for $0 \le i \le n$. Clearly, there are integers i and j, $0 \le i < j \le n+1$, such that $a_i = a_j$. Therefore the path

$$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_{n+1}$$

contains a cycle and can be continued to an infinite path.

Now we interpret H_G as the graph of a (nondeterministic) finite automaton $\mathcal{A} = (\{X\}, V, a, V, \delta)$ where all edges are labelled by X, the start state is a and all states are accepting states. Then the existence of an infinite path starting from a is equivalent to the infinity of the (regular) language accepted by $\mathcal{A}$. Because it is decidable whether or not the language accepted by a finite automaton is finite or not (see Theorem 5.9 and Claim 1 holds, there is an polynomial algorithm which decides whether or not $a \in V_s(G)$. Thus we can algorithmically construct the set $V_s(G)$ in polynomial time.

Now we construct the directed graph $H'_G = (V_s(G), E')$ where E' is the restriction of E to $V_s(G) \times V_s(G)$. Further, we define a labelling of the edges of E' by the letters X and Y. We label $(a, b) \in E'$ by X if and only if there is a production $a \rightarrow x_1 b x_2 \in P$ where $x_1 x_2$ contains a letter of $V_s(G)$. Otherwise, we label (a, b) by Y.

Claim 2: $L(G)$ is infinite if and only there is an infinite path in H'_G starting in a letter occurring in ω and containing an infinite number of occurrences of edges labelled by X.

Assume that there is an infinite path $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots$ starting from $a = a_0$ which occurs in ω. Let $j \ge 0$. Then a_j occurs in some word of w_j of $L_j(G)$. Let $w_j = x_0 b_1 x_1 b_2 x_2 \ldots b_r x_r$ for some $r \ge 1$, some words $x_k \in (V \setminus V_s(G))^*$, $0 \le k \le r$ and some letters $b_l \in V_s(G)$, $1 \le l \le r$. Let $a_j = b_s$. Then we consider a derivation $w_j \Longrightarrow w_{j+1} = x'_0 y_1 x'_1 y_2 x'_2 \ldots y_r x'_r$, where any subword y_l, $1 \le l \le r$ contains at least one letter of $V_s(G)$ (such rules exist for letters of $V_s(G)$ by definition) and a rule $a_j \rightarrow x_1 a_{j+1} x_2$ to $b_s = a_j$. If (a_j, a_{j+1}) is labelled by X, then $y_s = x_1 a_{j+1} x_2$ contains at least two occurrences of surviving letters. Therefore w_{j+1} contains at least $r+1$ occurrences of surviving letters. Continuing in this way we obtain words in $L(G)$ with an arbitrarily large number of occurrences of letters of $V_s(G)$. Thus $L(G)$ has to be infinite.

Conversely, let us assume that $L(G)$ is infinite. Then, for any number $i \ge 1$, $L(G)$ contains a word of length $\ge i$. We can improve this statement to the following one: For any number $i \ge 1$, $L(G)$ contains a word with at least i occurrences of letters of $V_s(G)$. If we assume the contrary, then there is a number j such that any word of $L(G)$ contains at most j letters of $V_s(G)$. Then any word of $L(G)$ can be written as $W = x_0 b_1 x_1 b_2 x_2 \ldots b_r x_r$ for some $r \ge 1$, some words $x_k \in (V \setminus V_s(G)^*$, $0 \le k \le r$ and some letters $b_l \in V_s(G)$, $1 \le l \le r$, where $r \le j$. Since there is a number t such that $L_t(G_a) = \{\lambda\}$ for all $a \notin V_s(G)$, we have $w \overset{t}{\underset{G}{\Longrightarrow}} z_1 z_2 \ldots z_r$ where $b_i \overset{t}{\underset{G}{\Longrightarrow}} z_i$ for $1 \le i \le r$. If

$$K = \max\{w_a \mid a \rightarrow w_a \in P, a \in V\},$$

then $|z_1 z_2 \ldots z_r| \le rK^t \le jK^t$. This yields a bound for the length of the words in $L(G)$ in contrast to the infinity of $L(G)$.

Now let p be a sufficient large number, and let w be a word of $L(G)$ containing at least p occurrences of letters from $V_s(G)$. Then we have a derivation

$$\omega = u_0 a_0 v_0 \Longrightarrow u_1 a_1 v_1 \Longrightarrow u_2 a_2 v_2 \Longrightarrow \ldots \Longrightarrow u_m a_m v_m = w$$

such that $(a_i, a_{i+1}) \in E$ for $0 \leq i \leq m$ and the path

$$a_0 \to a_1 \to a_2 \to \ldots \to a_m$$

contains at least $n^2 + 1$ edges labelled by X. Let $a_{i_1}, a_{i_2}, \ldots, a_{i_t}$ be the nodes such that (a_{i_q}, a_{i_q+1}) is labelled by X. Then $t \geq n^2 + 1$. Hence there are f and g such that $1 \leq f < g \leq t$ and $(a_{i,f}, a_{i_f+1}) = (a_{i,g}, a_{i_g+1})$. Therefore there is an edge labelled by X which occurs at least two times in the path. Thus the path contains a cycle with an edge labelled by X. Therefore it can be extended to an infinite path which contains infinitely often edges with label X.

Again, we interpret H'_G as a nondeterministic finite automaton $\mathcal{B} = (\{X, Y\}, V, a, V, \delta')$ taking the above labelling of the edges. The existence of a path having the properties mentioned in Claim 2 is equivalent to the property that,

$$\text{for any } i \geq 0, \text{ there is a word } w_i \text{ in } T(\mathcal{B}) \text{ with } \#_X(w_i) \geq i. \tag{11.8}$$

We define the homomorphism $h : \{X, Y\}^* \to \{X\}^*$ by $h(X) = X$ and $h(Y) = \lambda$. Then (11.8) is equivalent to the infinity of $h(T(\mathcal{B}))$. By the proofs of the closure properties we can construct a regular grammar H or a deterministic finite automaton $\mathcal{B}'$ such that $L(G) = T(\mathcal{B}') = h(T(\mathcal{B}))$. The decidability of the finiteness problem for regular languages implies that we can decide whether or not $L(G)$ is infinite. $\square$

Theorem 11.23 *i) For P0L systems, the equivalence problem is undecidable.*
ii) For D0L systems, the equivalence problem is decidable.

Proof. We only prove i). The proof for ii) is omitted because it is long if it is based on the elementary knowledge and formal language theory or it is based on deep results of (mathematical) group theory. A proof of ii) can be found in [27].

We prove i) by reduction to the Post Correspondence Problem. Let

$$U = \{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$$

be a set of pairs of words with $u_i, v_i \in \{a, b\}^*$ for $1 \leq i \leq n$.

We consider the 0L systems

$$G_1 = (V, P, S) \quad \text{and} \quad G_2 = (V, P', S)$$

with

$$\begin{aligned}
V &= \{S, S', S'', S_u, S_r, S_l, S', a, b, c\}\,,\\
P &= \{S \to S', S \to S'', S_u \to c, S_l \to c, S_r \to c\}\\
&\quad \cup \bigcup_{x \in \{a,b\}} \{S' \to xS'x, S' \to xS_l, S' \to S_r x, S_l \to xS_l, S_r \to S_r x, S_u \to S_r x, S_u \to xS_l\}\\
&\quad \cup \{S' \to xS_u y : x, y \in \{a, b\}, x \neq y\} \cup \{S_u \to xS_u y : x, y \in \{a, b\}\}\\
&\quad \cup \bigcup_{i=1}^{n} \{S'' \to u_i S'' v_i^R\}\,,\\
P' &= P \cup \bigcup_{i=1}^{n} \{S'' \to u_i c v_i^R\}\,.
\end{aligned}$$

It is easy to show that

$$\begin{aligned}
L(G_1) &= \{S, S', S''\} \cup \{\alpha S' \alpha^R : \alpha \in \{a,b\}^+\} \\
&\cup \{\alpha S_u \beta^R : \alpha, \beta \in \{a,b\}^+,\ |\alpha| = |\beta|,\ \alpha \neq \beta\} \\
&\cup \{\alpha S_r \beta^R : \alpha, \beta \in \{a,b\}^+,\ |\alpha| < |\beta|\} \\
&\cup \{\alpha S_l \beta^R : \alpha, \beta \in \{a,b\}^+,\ |\alpha| > |\beta|\} \\
&\cup \{\alpha c \beta^R : \alpha, \beta \in \{a,b\}^+,\ \alpha \neq \beta\} \\
&\cup \{u_{i_1} u_{i_2} \dots u_{i_k} S'' v_{i_k} v_{i_{k-1}} \dots v_{i_1} : k \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq k\}
\end{aligned}$$

and

$$L(G_2) = L(G_1) \cup \{u_{i_1} u_{i_2} \dots u_{i_k} c v_{i_k} v_{i_{k-1}} \dots v_{i_1} : k \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq k\}.$$

Obviously, $L(G_1) \subseteq L(G_2)$, and $L(G_1) = L(G_2)$ holds if and only if the part added to $L(G_1)$ to obtain $L(G_2)$ is contained in $\{\alpha c \beta^R : \alpha, \beta \in \{a,b\}^+,\ \alpha \neq \beta\}$ (see the proof of Theorem 5.10). Thus we get $L(G_1) = L(G_2)$ iff the Post Correspondence Problem has no solution. □

Note that Theorem 11.23 implies that the equivalence problem for PD0L systems is decidable and that the equivalence problem for 0L systems is undecidable.

We now give the decidability status of the above problems with respect to adult languages.

Theorem 11.24 *i) Given a 0L system G and a word w, it is decidable whether or not $w \in L_A(G)$ holds.*

ii) Given a 0L system G, it is decidable whether or not $L_A(G)$ is empty.

iii) Given a 0L system G, it is decidable whether or not $L_A(G)$ is finite.

iv) Given two P0L systems G_1 and G_2, it is undecidable whether or not $L_A(G_1) = L_A(G_2)$.

v) For two given D0L systems G_1 and G_2 it is decidable whether or not $L_A(G_1) = L_A(G_2)$.

Proof. Since the transformations of an 0L system G into a context-free grammar G' and of a context-free grammar G' (with $\lambda \notin L(G')$) into a (P)0L system G such that $L_A(G) = L(G')$ are constructive, we get the first four results from the decidability results for context-free languages.

The equivalence of two D0L system with respect to adult languages is decidable, because we can first check whether both system generate a non-empty adult language (consisting of one word) and then we determine the adult languages and compare them. □

11.1.6 Growth Functions

A very important field in the study of the development of filamentous organisms and plants is the growth of the organism or plant. Usually, as a measure of the size of the plant one takes the number of cells which build it and the growth is measured by a function which associates with a given time moment the size of the plant at this moment.

We now formalize this concept. In order to get a function one has to ensure that at every moment only one organism exists, i. e. the Lindenmayer system has to generate exactly one word. Therefore we have to restrict to deterministic Lindenmayer systems. For a D0L system $G = (V, P, \omega)$, we have a uniquely determined derivation

$$\omega = w_0 \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \ldots \Longrightarrow w_m \Longrightarrow \ldots, \tag{11.9}$$

and thus $L_m(G)$ contains exactly one element w_m. Conversely, let H be a 0L system which, for some letter $a \in V$, has two rules $a \to w_1$ and $a \to w_2$ with $w_1 \neq w_2$ in its set of productions, and let a occur in some word w of $L(H)$ (otherwise we can omit a and its rules). Then we can generate two words from w because we have the derivations $w = x_1ax_2 \Longrightarrow x_1'w_1x_2'$ and $w = x_1ax_2 \Longrightarrow x_1'w_2x_2'$. Hence $L_m(H)$ for some $m \geq 1$ contains at least two words.

Definition 11.25 *The* growth function $f_G : \mathbb{N} \to \mathbb{N}$ *of a deterministic 0L system G is defined by*

$$f_G(m) = |w_m|.$$

Example 11.26 We consider the deterministic 0L systems

$$\begin{aligned} G_1 &= (\{a\}, \{a \to a^2\}, a)\,, \\ G_2 &= (\{a, b\}, \{a \to \lambda, b \to ab\}, aab)\,, \\ G_4 &= (\{a, b, c, d, e\}, \{a \to a, b \to ba, c \to cbb, d \to da, e \to cbbd\}, e) \end{aligned}$$

given in the Examples 11.4, 11.5 and 11.7.

In G_1, the only derivation is

$$a \Longrightarrow a^2 \Longrightarrow a^4 \Longrightarrow a^8 \Longrightarrow \ldots,$$

which results in

$$f_{G_1}(m) = 2^m \quad \text{for} \quad m \geq 0.$$

In G_2, we only have the derivation

$$aab \Longrightarrow \lambda\lambda ab = ab \Longrightarrow \lambda ab = ab \Longrightarrow ab \Longrightarrow ab \Longrightarrow \ldots,$$

which gives

$$f_{G_2}(0) = 3 \quad \text{and} \quad f_{G_2}(m) = 2 \text{ for } m \geq 1.$$

In Example 11.7, we have shown that

$$\begin{aligned} L_0(G_4) &= \{e\}\,, \\ L_m(G_4) &= \{cbb(ba)^2(ba^2)^2 \ldots (ba^{m-1})^2da^{m-1}\} \text{ for } m \geq 1\,. \end{aligned}$$

Thus we get $f_{G_4}(0) = |e| = 1$ and, for $m \geq 1$,

$$\begin{aligned} f_{G_4}(m) &= |cbb(ba)^2(ba^2)^2 \dots (ba^{m-1})^2 da^{m-1}| \\ &= 1 + 2 \cdot 1 + 2 \cdot 2 + 2 \cdot 3 + \cdots + 2 \cdot m + 1 + (m-1) \\ &= m + 1 + 2 \cdot \sum_{i=1}^{m} i = m + 1 + 2 \cdot \frac{m(m+1)}{2} = m^2 + 2m + 1 \\ &= (m+1)^2 . \end{aligned}$$

This gives

$$f_{G_4}(m) = (m+1)^2 \quad \text{for} \quad m \geq 0.$$

In the examples we have determined the growth function by a determination of the sequence of words generated by the system and obtained $f_G(m)$ as the length of $|w_m|$ according to the definition. However, in biology one is interested in a computation of $f_G(m)$ for arbitrary m, especially for large m, without a determination of w_m. Such a computation can easily be done if one has a formula for f_G. In the sequel we shall present some such formulae.

In the sequel we shall assume that the D0L system under consideration is given as

$$G = (\{a_1, a_2, \dots, a_n\}, \{a_1 \to v_1, a_2 \to v_2, \dots, a_n \to v_n\}, \omega) \tag{11.10}$$

and that its only derivation is given by (11.9).

Definition 11.27 *Let G be a D0L system as in (11.10). Then we define the growth matrix M_G of G as the (n,n)-matrix*

$$M_G = (a_{i,j}) = (\#_{a_j}(v_i))_{n,n}.$$

Example 11.28 Again, we consider the systems G_1, G_2 and G_4. We get

$$M_{G_1} = (2),\ M_{G_2} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix},\ M_{G_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 2 & 1 & 1 & 0 \end{pmatrix}$$

Theorem 11.29 *Let G be a D0L system as in (11.10), and let M_G be its growth matrix. Then, for $m \geq 0$,*

$$f_G(m) = \Psi(\omega)(M_G)^m(\underbrace{1, 1, \dots, 1}_{n \text{ times}})^T.$$

Proof. First we note that

$$\begin{aligned} \Psi(w_m) \cdot (1, 1, \dots, 1)^T &= (\#_{a_1}(w_m), \#_{a_2}(w_m), \dots, \#_{a_n}(w_m))(1, 1, \dots, 1)^T \\ &= \sum_{i=1}^{n} \#_{a_i}(w_m) = |w_m| \\ &= f_G(m) . \end{aligned}$$

Thus it is sufficient to to prove that, for $m \geq 1$,

$$\Psi(w_m) = \Psi(\omega) M_G^m.$$

This will be done by induction on m.

$m = 0$. We have

$$\Psi(w_0) = \Psi(\omega) = \Psi(\omega) \cdot E = \Psi(\omega) M_G^0,$$

where E is the unit matrix. Thus the induction basis is shown.

$m > 0$. By induction hypothesis,

$$\Psi(\omega) M_G^m = \Psi(\omega) M_G^{m-1} M_G = \Psi(w_{m-1}) M_G \tag{11.11}$$

Further, any occurrence of a letter a_i, $1 \leq i \leq n$, in w_{m-1} contributes $\#_{a_j}(v_i)$ occurrences of a_j, $1 \leq j \leq n$, in w_m. Thus

$$\#_{a_j}(w_m) = \sum_{i=1}^{n} \#_{a_i}(w_{m-1}) \#_{a_j}(v_i).$$

This implies

$$\Psi(w_m) = \Psi(w_{m-1}) M_G.$$

Together with (11.11) we get

$$\Psi(w_m) = \Psi(\omega) M_G^m.$$

□

Let

$$\chi_{M_G}(x) = \det(M_G - xE) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

be the characteristic function of M_G. By the Cayley-Hamilton Theorem (see Theorem 1.2),

$$O = \chi_{M_G}(M_G) = a_n M_G^n + a_{n-1} M_G^{n-1} + \cdots + a_1 M_G + a_0 E.$$

For $k \geq 0$, by left and right multiplication with $\Psi(\omega) M_G^k$ and $(1, 1, \ldots 1)^T$, respectively, and Theorem 11.29, we obtain

$$\begin{aligned} 0 &= a_n \Psi(\omega) M_G^{k+n} (1,1,\ldots,1)^T + a_{n-1} \Psi(\omega) M_G^{k+n-1} (1,1,\ldots,1)^T + \ldots \\ &\quad a_1 \Psi(\omega) M_G^{k+1} (1,1,\ldots,1)^T + a_0 \Psi(\omega) M_G^k (1,1,\ldots,1)^T \\ &= a_n f_G(k+n) + a_{n-1} f_G(k+n-1) + \cdots + a_1 f_G(k+1) + a_0 f_G(k)\,. \end{aligned}$$

Thus the growth function f_G satisfies the difference equation

$$0 = a_n h(k+n) + a_{n-1} h(k+n-1) + \cdots + a_1 h(k+1) + a_0 h(k).$$

Using the theory of difference equation (see Section 1.2) we get

$$h(x) = \sum_{i=1}^{s} (\beta_{i,0} + \beta_{i,1} x + \beta_{i,2} x^2 + \ldots \beta_{i,t_i-1} x^{t_i-1}) \mu_i^x$$

where, for $1 \le i \le s$, μ_i is a root of multiplicity t_i of

$$g(y) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

$\sum_{i=1}^{s} t_i = n$, and $\beta_{i,j}$, $1 \le i \le s$, $0 \le j \le t_i - 1$ are n real constants, which are uniquely determined by the values $h(0), h(1), \ldots, h(n-1)$.

If we take into consideration that $\chi_{M_G} = g$ holds, then the roots β_i, $1 \le i \le s$ are the eigenvalues of M_G. Thus we obtain the following theorem.

Theorem 11.30 *Let G be a D0L system as in (11.10), and let M_G be its growth matrix. For $1 \le i \le s$, let μ_i be a eigenvalue of M_G of multiplicity t_i such that $\sum_{i=1}^{s} t_i = n$. Then*

$$f_G(m) = \sum_{i=1}^{s} (\beta_{i,0} + \beta_{i,1} m + \beta_{i,2} m^2 + \ldots \beta_{i,t_i-1} m^{t_i-1}) \mu_i^m$$

for certain constants $\beta_{i,j}$, $1 \le i \le s$, $0 \le j \le t_i - 1$. □

Example 11.31 We apply the theory developed up to this point to the our D0L systems G_1, G_2 and G_4. In order to simplify the notation, we shall sometimes only use the indexes 1, 2 and 4 to refer to G_1, G_2 and G_4, respectively.

Then we get

$$\chi_1(x) = \det(M_{G_1} - xE) = \det(2 - x) = 2 - x\,.$$

The only eigenvalue is $\mu_1 = 2$ of multiplicity 1. Thus we get $f_{G_1}(m) = \beta_0 2^m$. Since $f_{G_1}(0) = 1 = \beta_0 2^0 = \beta_0$ we obtain $\beta_0 = 1$ which yields $f_{G_1}(m) = 2^m$ for $m \ge 0$.

Considering G_2 we have

$$\chi_2(x) = \det(M_{G_2} - xE) = \det \begin{pmatrix} -x & 0 \\ 1 & 1-x \end{pmatrix} = -x(1-x)\,.$$

Therefore the eigenvalues of G_2 are $\mu_1 = 0$ and $\mu_2 = 1$ which both are of multiplicity 1. Thus

$$f_{G_2}(m) = \beta_{1,0} 0^m + \beta_{2,0} 1^m.$$

In the sequel we shall assume that $0^0 = 1$ (note that 0^0 is an indefinite expression whose value depends on the context). Then we have

$$\begin{array}{rrcl} \beta_{1,0} & +\beta_{2,0} & = & 3 = f_{G_2}(0) \\ & \beta_{2,0} & = & 2 = f_{G_2}(1). \end{array}$$

The solutions of this system of linear equations are $\beta_{1,0} = 1$ and $\beta_{2,0} = 2$. Consequently,

$$f_{G_2}(0) = 3 \quad \text{and} \quad f_{G_2}(m) = 2 \text{ for } m \ge 1.$$

For G_4, we get

$$\chi_4(x) = \det(M_{G_4} - xE) = \det \begin{pmatrix} 1-x & 0 & 0 & 0 & 0 \\ 1 & 1-x & 0 & 0 & 0 \\ 0 & 2 & 1-x & 0 & 0 \\ 1 & 0 & 0 & 1-x & 0 \\ 0 & 2 & 1 & 1 & -x \end{pmatrix} = (1-x)^4(-x)\,.$$

Thus the eigenvalues of M_{G_4} are $\mu_1 = 1$ of multiplicity 4 and $\mu_2 = 0$ of multiplicity 1. Therefore we get

$$f_{G_4}(m) = \beta_{1,0}0^m + (\beta_{2,0} + \beta_{2,1}m + \beta_{2,2}m^2 + \beta_{2,3}m^3)1^m.$$

The constants $\beta_{1,0},\ \beta_{2,0},\ \beta_{2,1},\ \beta_{2,2},\ \beta_{2,3}$ can be determined as the solutions of the following of linear equations

$$\begin{array}{cccccccc}
\beta_{1,0} & +\beta_{2,0} & & & & = & 1 = f_{G_4}(0) \\
 & \beta_{2,0} & +\beta_{2,1} & +\beta_{2,2} & +\beta_{2,3} & = & 4 = f_{G_4}(1) \\
 & \beta_{2,0} & +2\beta_{2,1} & +4\beta_{2,2} & +8\beta_{2,3} & = & 9 = f_{G_4}(2) \\
 & \beta_{2,0} & +3\beta_{2,1} & +9\beta_{2,2} & +27\beta_{2,3} & = & 16 = f_{G_4}(3) \\
 & \beta_{2,0} & +4\beta_{2,1} & +16\beta_{2,2} & +64\beta_{2,3} & = & 25 = f_{G_4}(4)\ .
\end{array}$$

We obtain

$$\beta_{1,0} = 0,\ \beta_{2,0} = 1,\ \beta_{2,1} = 2,\ \beta_{2,2} = 1,\ \beta_{2,3} = 0,$$

which leads $f_{G_4}(m) = 1 + 2m + m^2 = (m+1)^2$ for $m \geq 1$.

We mention that the formulae given in Theorems 11.29 and 11.30 have some nice and some bad features. One formula uses the Parikh vector of ω and growth matrix, which both can directly be obtained from the given system, however, the computation of $f_G(m)$ requires the calculation of the m-th power of a matrix. By the other formula, it is easy to compute the value of the growth function for an arbitrarily given argument, however, we note that it is hard to compute the eigenvalues of a growth matrix since they can be complex and the degree of the characteristic function will be arbitrarily large for arbitrarily large alphabets.

Theorem 11.32 *Let G be a D0L system. Then the growth f_G satisfies one of the following conditions:*
a) there is a constant c such that $f_G(m) \leq c$ for all m
b) there are constants c_1, c_2 and p such that $c_1 m^p \leq f_G(m) \leq c_2 m^p$ for large m,
c) there are constants c_1 and c_2 such that $c_1^m \leq f_G(m) \leq c_2^m$ for large m.

Proof. We only give the proof for the case that the eigenvalues of M_G are non-negative real numbers, for a general proof (including complex roots) one has to consider the absolute values $|\mu|$ of the eigenvalues μ of M_G.

If G generates a finite language, then case a) holds obviously.

Let us now assume that G generates an infinite language. Let μ be the largest eigenvalue of M_G. Then

$$f_G(m) = (\beta_0 + \beta_1 m + \beta_2 m^2 + \cdots + \beta_{t-1}m^{t-1})\mu^m + R(m)$$

where t is the multiplicity of μ and $R(m)$ is asymptotically smaller than $(\beta_0 + \beta_1 m + \cdots + \beta_{t-1}m^{t-1})\mu^m$.

If $\mu > 1$ holds, then f_G is asymptotically equal to $(\beta_0 + \beta_1 m + \cdots + \beta_{t-1}m^{t-1})\mu^m$. We choose d such that

$$\beta_0 + \beta_1 m + \beta_2 m^2 + \cdots + \beta_{t-1}m^{t-1} \leq d^m$$

for large m. Then we get

$$f_G(m) \leq d^m \mu^m = (d\mu)^m = c_2^m$$

for large m. On the other hand $\beta_0 + \beta_1 m + \cdots + \beta_{t-1} m^{t-1} > 0$ for large m. Therefore, we also have

$$\mu^m = c_1^m \leq f_G(m)$$

for large m. Thus we have case c).

If $\mu = 1$ holds, then f_G is asymptotically equal to $\beta_0 + \beta_1 m + \beta_2 m^2 + \cdots + \beta_{t-1} m^{t-1}$. Then it is easy to see that case b) holds if $t \geq 2$ and that a) holds if $t = 1$.

If $\mu < 1$, then $(\beta_0 + \beta_1 m + \beta_2 m^2 + \cdots + \beta_{t-1} m^{t-1})\mu^m$ tends to zero and the same holds for $R(m)$. Since the range of f_G is the set of non-negative integers, this situation cannot occur (since we assume that the generated language is infinite). □

Theorem 11.32 says that we have only three different types of growth functions of D0L systems. Thus there is no DOL system G such that $f_G(m) = \log(m)$ holds. If we have seen that the real growth of a plant is logarithmic, then we cannot take a D0L system to model the development.

11.2 Lindenmayer Systems with Interaction

11.2.1 Definitions and Examples

It is a well-known fact that in reality other growth function also occur, for example there are organisms with logarithmic growth. The development of such an organism cannot be modelled by D0L systems.

In order to obtain more powerful systems one can take into consideration the context of a cell, i. e., the rules for the development of a cell does not only depend on the cell itself, it also depends on neighbouring cells. This reflects the biological situation much better than the case without interaction considered in the preceding section.

Again, we model the cells by elements of an alphabet and the organisms by words. However, we assume that the development of a cell in an organism depends on its k left neighbours and its l right neighbours. Obviously, the first and last letters do not have k and l neighbours, respectively. Therefore we add a new letter \$ and prolong the word by powers of \$ to the right and to the left such that any letter has k left and l right neighbours. Furthermore, we require a completeness condition to ensure that we have a rule for any situation which can occur. Then, for any letter in a word, we have k left and l right neighbours and a rule with respect to these neighbours. Again, the application of rules is a purely parallel process of rewriting.

Formally we get the following concepts.

Definition 11.33 *Let k and l be two non-negative integers. A $\langle k, l\rangle$* Lindenmayer system *($\langle k, l\rangle L$ system for short) is a quadruple $G = (V, \$, P, \omega)$ where*

1. *V is an alphabet, and \$ is a symbol not occurring in V (used as an endmarker),*
2. *P is a finite set of quadruples (u, a, v, w) where*
   (a) *$u = \$^r u'$ for some $r \in \mathbb{N}_0$ and some $u' \in V^*$ with $|u'| = k - r$,*

(b) $a \in V$,
(c) $v = v'\$^s$ *for some* $s \in \mathbb{N}_0$ *and some* $v' \in V^*$ *with* $|v'| = l - s$,
(d) $w \in V^*$

and, for any triple (u, a, v) *with the properties a), b) and c), there is a* $w \in V^*$ *such that* $(u, a, v, w) \in P$.

3. ω *is a non-empty word over* V.

As usual we write $(u, a, v) \to w$ instead of (u, a, v, w). Moreover, if we consider a $\langle k, 0\rangle$L or $\langle 0, l\rangle$L system, then we omit the non-existing context to the right or to the left, and write only $(u, a) \to w$ or $(a, v) \to w$, respectively.

Definition 11.34 *Let* G *be a* $\langle k, l\rangle L$ *system as in Definition 11.33.*

i) Let x *be a non-empty word over* V *and* $y \in V^*$. *We say that* x directly derives y *(written as* $x \underset{G}{\Longrightarrow} y$ *or* $x \Longrightarrow y$ *if* G *is understood) if the following conditions are satisfied:*

– $x = a_1 a_2 \dots a_n$ *with* $a_i \in V$ *for* $1 \le i \le n$,
– $y = y_1 y_2 \dots y_n$,
– $(u_i, a_i, v_i) \to y_i \in P$ *where*

$$u_i = \begin{cases} \$^{k-i+1} a_1 a_2 \dots a_{i-1} & \text{for } 1 \le i \le k \\ a_{i-k} a_{i-k+1} \dots a_{i-1} & \text{for } k < i \end{cases}$$

and

$$v_i = \begin{cases} a_{i+1} a_{i+2} \dots a_{i+l} & \text{for } i + l \le n \\ a_{i+1} a_{i+2} \dots a_n \$^{l+i-n} & \text{for } n < i + l \end{cases}$$

ii) The language $L(G)$ generated by G *is defined as*

$$L(G) = \{z \mid \omega \Longrightarrow_G^* z\}$$

where $\overset{*}{\underset{G}{\Longrightarrow}}$ *denotes the reflexive and transitive closure of* $\underset{G}{\Longrightarrow}$.

Example 11.35 We consider the $\langle 1, 0\rangle$L system $G_7 = (\{a, b, c\}, \$, P_7, c)$ with

$$\begin{aligned} P_7 \;=\; & \{(\$, a) \to a^2, (\$, b) \to b, (\$, c) \to a, (\$, c) \to ba^2, (a, a) \to a^2\} \\ & \cup \{(p, q) \to q \mid (p, q) \in \{a, b, c\} \times \{a, b, c\} \setminus \{(a, a)\}\}\,. \end{aligned}$$

First we have the derivations $c \Longrightarrow a$ and $c \Longrightarrow ba^2$. If we have a word a^n, then any letters is doubled according to the rules, which leads to a^{2n}. Starting from a we get all words a^{2^n} for $n \ge 0$. If we have a word ba^m with $m \ge 1$, then we replace b by b, the first a by a, and all remaining a's by a^2. Thus we get

$$ba^{2^n+1} = baa^{2^n} \Longrightarrow baa^{2^{n+1}} = ba^{2^{n+1}+1}.$$

Therefore we obtain

$$L(G_7) = \{c\} \cup \{a^{2^n} \mid n \ge 0\} \cup \{ba^{2^n+1} \mid n \ge 0\}.$$

Example 11.36 We consider the $\langle 1,1\rangle$L system $G_8 = (\{a,b\}, \$, P_8, ab^2)$ with P consisting of the following rules:

$$\begin{array}{ll} (u,a,b) \rightarrow a^2 & \text{for} \quad u \in \{a,b,\$\}, \\ (a,b,v) \rightarrow b^3 & \text{for} \quad v \in \{a,b,\$\}, \\ (u,z,v) \rightarrow z & \text{in all other cases} \end{array}$$

Assume that we have a word $a^n b^{2n}$. Then we have to replace the last letter a by a^2, the first letter b by b^3 and the remaining letters x by x. Therefore

$$a^n b^{2n} = a^{n-1} abb^{2n-1} \Longrightarrow a^{n-1} a^2 b^3 b^{2n-1} = a^{n+1} b^{2(n+1)}$$

for $n \geq 1$, and hence

$$L(G_8) = \{a^n b^{2n} \mid n \geq 1\}.$$

Example 11.37 We consider the $\langle 1,0\rangle$L system $G_9 = (\{a,b,o,r\}, \$, P_9, ar)$ with P_9 consisting of the following rules:

$$\begin{array}{ll} (\$,a) \rightarrow o,\ (o,a) \rightarrow b,\ (o,b) \rightarrow o,\ (o,r) \rightarrow ar, & \\ (u,o) \rightarrow a & \text{for} \quad u \in \{a,b,o,r,\$\}, \\ (u,z) \rightarrow z & \text{in all other cases} \end{array}$$

We note that the system is deterministic because, for any pair (u,a), there is exactly one rule $(u,a) \rightarrow w$. Then we get the only derivation

$$\begin{array}{lll} ar & \Longrightarrow & or \Longrightarrow aar \Longrightarrow oar \Longrightarrow abr \Longrightarrow obr \Longrightarrow aor \\ & \Longrightarrow & oaar \Longrightarrow abar \Longrightarrow obar \Longrightarrow aoar \Longrightarrow oabr \Longrightarrow abbr \\ & \Longrightarrow & obbr \Longrightarrow aobr \Longrightarrow oaor \Longrightarrow abaar \Longrightarrow obaar \Longrightarrow \dots \end{array}$$

We do not determine the language in detail, but we note some properties of the sequence generated.

Fact 1: Each word of $L(G_9)$ starts with o or a.

The statement holds for the start word, and in the sequel o and a alternate as the first letter by the rules $(\$,a) \rightarrow o$ and $(\$,o) \rightarrow a$.

Fact 2: No word of $L(G_9)$ has the subword oo.

If we want to produce an o which is not in the beginning of the word, then we have to apply the rule $(o,b) \rightarrow o$. This requires that the word to which we apply the rule is of the form $x_1 obx_2$ for some words x_1 and x_2. If x_1 ends on a letter different from o, then we get $x_1' aox_2'$. That means, in order to produce oo as a subword in v' with $v \Longrightarrow v'$ the word v has already to contain the subword oo. Because the start word does not contain oo as a subword, no word of $L(G_9)$ contains oo.

Fact 3: For any words $u,v \in \{a,b,o\}^+$ and $z \in \{a,b\}^*$ we have derivations $ubzr \overset{*}{\Longrightarrow} u'ozr$ and $vazr \overset{*}{\Longrightarrow} v'abzr$ for some u' and v' with $|u'| = |u|$ and $|v'| = |v| - 1$.

We prove the statement by simultaneous induction on the length of u and v.

Let $|u| = 1$. By Fact 1, $u = o$ or $u = a$, we have the derivations $obzr \Longrightarrow aozr$ and $abzr \Longrightarrow obzr \Longrightarrow aozr$, respectively. Thus the induction basis holds for $ubzr$. Analogously we prove it for $vazr$.

Let $|v| \geq 2$. We distinguish three cases.

Case 1: $v = v_1o$. Then $vazr = v_1oazr \Longrightarrow v_1'abzr$, and $|v_1'| = |v_1| = |v| - 1$. Therefore the induction step is done.

Case 2: $v = v_1b$. Then $vazr = v_1bazr$. Now we apply the induction assumption for $v_1bz'r$ with $z' = az$ (this can be done since $|v_1| = |v| - 1 < |v|$)and get $v_1bz'r \stackrel{*}{\Longrightarrow} v_1'oz'r = v_1'oazr \Longrightarrow v_1''abzr$ and $|v_1''| = |v_1'| = |v_1| = |v| - 1$.

Case 3: $v = v_1a$. Then $vazr = v_1aazr \stackrel{*}{\Longrightarrow} v_1'abazr$ by induction hypothesis. Moreover, $|v_1'ab| = |v_1'| + 2 = |v_1| + 1 = |v|$. By Case 2 (with $v' = v_1'ab$ and $|v'| = |v|$), we know that $v_1'abazr = v'azr \stackrel{*}{\Longrightarrow} v''abzr$ with $|v''| = |v'| - 1 = |v_1'ab| - 1 = |v| - 1$.

Analogously we prove the statement for $|u| \geq 2$.

Now assume that in some step of the derivation we have an extension of the word with respect to the length. By the rules, the only possibility is $xor \Longrightarrow x'aar = yar$. Let $|yar| = s + 2$. Now, by Fact 3, we have the following derivation

$$yar \stackrel{*}{\Longrightarrow} y_1abr \stackrel{*}{\Longrightarrow} y_2abbr \stackrel{*}{\Longrightarrow} y_3abbbr \stackrel{*}{\Longrightarrow} \ldots \stackrel{*}{\Longrightarrow} y_{s-1}ab^{s-1}r \stackrel{*}{\Longrightarrow} ab^sr \tag{11.12}$$

where $s = |y| = |y_i| + i$ for $1 \leq i \leq s - 1$, followed by the derivation

$$ab^sr \Longrightarrow ob^sr \Longrightarrow p_1ob^{s-1}r \Longrightarrow p_2ob^{s-2}r \Longrightarrow \ldots \Longrightarrow p_sor \Longrightarrow p_{s+1}aar \tag{11.13}$$

where $|p_i| = i$ for $1 \leq i \leq s$ and $|p_{s+1}| = s$. Therefore we get a word of length $s + 3$.

This proves that $L(G_9)$ is infinite.

We note that it requires at least $2s + 2$ derivation steps to reach a word of length $s + 3$ from a word of length $s + 2$.

Example 11.38 For any $k \in \mathbb{N}_0$ and any $l \in \mathbb{N}_0$, the context-free language

$$L = \{a^nb^{2n} \mid n \geq 1\} \cup \{a^{2n}b^n \mid n \geq 1\}$$

cannot be generated by a $\langle k, l\rangle$L systems. This can be seen as follows.

Assume the contrary, i.e., there is a $\langle k, l\rangle$L system $G = (\{a, b\}, \$, P, \omega)$ for some non-negative integers k and l such that $L(G) = L$.

To words a^nb^{2n} and $a^{2n}b^n$ with sufficiently large n, we can only apply rules with left hand sides $(\$^ra^{k-r}, a, a^l)$, (a^k, a, a^sb^{l-s}), (a^rb^{k-r}, b, b^l) and $(b^l, b, b^s\$^{l-s})$ with $0 \leq r \leq k$ and $0 \leq s \leq l$. We prove some facts on the rules with these left hand sides.

Fact 1: If $(\$^ra^{k-r}, a, a^l) \to w \in P$*, then* $w \in \{a\}^*$*, and if* $(b^k, b, b^r\$^{l-r}) \to v \in P$*, then* $v \in \{b\}^*$*. Moreover, there are rules* $(a^k, a, a^l) \to w \in \{a\}^+$ *and* $(b^k, b, b^l) \to v \in \{b\}^+$*.*

We only give a proof for the part where a is replaced; the proof concerning b can be given analogously.

We first show the statement for (a^k, a, a^l).

Let $n > k$. Let $a^nb^{2n} = a^ka^{n-k-l}a^lb^{2n}$. Let $(a^k, a, a^l) \to w \in P$ and let w_1 and w_2 be the words obtained from the first k letters a, from the subword a^lb^{2n} and from the last l letters b using some fixed rules, respectively. Then $a^nb^{2n} \Longrightarrow w_1w^{n-k-l}w_2 = z$. If

w contains a as well as b, then a b occurs before an a in w^2 and therefore in z which contradict $z \in L$. If $w \in \{b\}^+$, then w^{n-k-l} contains at least $n-k-l$ occurrences of b and w_2 is also in $\{b\}^*$. Therefore z contains at most $|w_1|$ occurrences of a. Because $|w_1| < n-k-l$ for large n, we get a contradiction to $z \in L$, again. Thus $(a^k, a, a^l) \to w$ implies $w \in \{a\}^*$.

Analogously, we prove that $(b^k, b, b^l) \to v$ implies $v \in \{b\}^*$.

Let us assume that $(a^k, a, a^l) \to \lambda$ is the only rule with left side (a^k, a, a^l). Let w_2' be a word derivable from the subword $a^l b^k$ and w_3' a word derivable from b^l. Then

$$a^n b^{2n} = a^k a^{n-k-l} a^l b^k b^{2n-k-l} b^l \Longrightarrow w_1 w_2' v^{2n-k-l} w_3'$$

for some $v \in \{b\}^*$. If v contains a b, then – as above – the word $w_1 w_2'$ of bounded length has to contain at least $\frac{2n-k-l}{2}$ occurrences of a which is impossible. If $(b^k, b, b^l) \to \lambda$ is the only rule for b in that context, then we can only obtain words $w_1 w_2' w_3'$, where w_1 , w_2', and w_3' are derivable from the subwords a^k, $a^l b^k$, and b^l, respectively, of $a^n b^{2n}$ and $a^{2n} b^n$. Thus we cannot generate infinitely many words. This contradicts the infinity of L.

Thus we have a rule $(a^k, a, a^l) \to a^s$ with $s \leq 1$. Analogously, we show that there is a rule $(b^k, b, b^l) \to b^t$ with $t \geq 1$.

Using these two rules we get $a^n b^{2n} \Longrightarrow w_1 a^{(n-k-l)s} w_2' b^{(2n-k-l)t} w_3'$ which implies $w_1 \in \{a\}^*$ and $w_3' \in \{b\}^*$. Thus, for $r \geq 1$, $(\$^r a^{k-r}, a, a^l) \to w \in P$ implies $w \in \{a\}^*$, and $(b^k, b, b^r \$^{l-r}) \to v \in P$ implies $v \in \{b\}^*$.

Fact 2: If $(a^k, a, a^l) \to w_1 \in P$ and $(a^k, a, a^l) \to w_2 \in P$, then $w_1 = w_2$, and if $(b^k, b, b^l) \to v_1 \in P$ and $(b^k, b, b^l) \to v_2 \in P$, then $v_1 = v_2$.

We prove the statement only for (a^k, a, a^l). Let $(a^k, a, a^l) \to a^{s_1}$ and $(a^k, a, a^l) \to a^{s_2}$ be two rules in P with $s_1 < s_2$. Let $n \geq k+l+2$, then we have the derivations $a^n b^{2n} = a^k a a^{n-k-1} b^{2n} = u_1 a^{s_1} u_2$ and $a^n b^{2n} \Longrightarrow u_1 a^{s_2} u_2$ where u_1 and u_2 are obtained from a^k and $a^{n-k-1} b^{2n}$, respectively. Obviously, $u_1 = a^x$ for some $x \geq 0$ and $u_2 = a^y b^k$ for some $y \geq (n-k-l-1)s_2$ (if we apply $(a^k, a, a^l) \to a^{s_2}$ to the first $n-k-l-1$ letters of $a^{n-k-1} b^{2n}$) and $k \geq 1$. Thus we have

$$a^n b^{2n} \Longrightarrow a^{x+s_1+y} b^k = z_1 \text{ and } a^n b^{2n} \Longrightarrow a^{x+s_2+y} b^k = z_2.$$

If $k = 2(x+s_2+y)$ Then $2(x+s_1+y) < k$ and therefore $z_1 \neq a^{k/2} b^k$ and $z_1 \neq a^{2k} b^k$ which contradicts $z_1 \in L$. If $2k = x+x_2+y$, then $2k > x+s_1+y$. Therefore $z_1 \neq a^{2k} b^k$. Hence we get $z_1 = a^{k/2} b^k$. We now get $2(x+s_1+y) = k$ and then $4(x+s_1+y) = 2k = x+s_2+y$. Now we have $s_2 - s_1 - 3x = 3y \geq 3(n-k-l-1)s_2$ which is impossible since $s_2 - s_1 - 3x$ is bounded whereas $3(n-k-l-1)s_2$ can be arbitrarily large.

Fact 3: For sufficiently large n, the prefix a^k, the subword $a^k b^l$ and the suffix b^l of $a^n b^{2n}$ or $a^{2n} b^n$ generate uniquely determined words a^p, $a^{p'} b^{q'}$, and b^q, respectively, for some $p \geq 0$, $p' \geq 0$ $q \geq 0$, and $q' \geq 0$.

The proof is analogous to that of Fact 2.

Let $(a^k, a, a^l) \to a^s$ and $(b^k, b, b^l) \to b^t$ be the only rules for (a^k, a, a^l) and (b^k, b, b^l), respectively. We note that $s \geq 1$ and $t \geq 1$ by Fact 1.

Thus, for sufficiently large n, we have the unique derivations

$$a^n b^{2n} \Longrightarrow a^p a^{(n-k-l)s} a^{p'} b^{q'} b^{(2n-k-l)t} b^q = a^{p+p'+(n-k-l)s} b^{q+q'+(2n-k-l)t} \quad (11.14)$$

and

$$a^{2n}b^n \Longrightarrow a^p a^{(2n-k-l)s} a^{p'} b^{q'} b^{(n-k-l)t} b^q = a^{p+p'+(2n-k-l)s} b^{q+q'+(n-k-l)t}. \tag{11.15}$$

Fact 4: There is a number n_0 such that, for all $n \geq n_0$, any word $a^n b^{2n}$ generates only words $a^m b^{2m}$ for some m and any word $a^{2n}b^n$ generates only words $a^{2m'}b^{m'}$ for some m'.

Assume the contrary, i. e., $a^n b^{2n} \Longrightarrow a^{2m}b^m$ for some m or $a^{2n}b^n \Longrightarrow a^{m'}b^{2m'}$ for some m'. We only discuss the former case; the latter one can be handle analogously. By (11.14), we get

$$2m = p + p' + (n-k-l)s \text{ and } m = q + q' + (2n-k-l)t.$$

By an easy calculation we get

$$n = \frac{p + p' - 2q - 2q' + k(2t-s) + l(2t-s)}{4t-s}.$$

This is a contradiction, since the left side is unbounded, but the right side is a constant.

Fact 5: $s = t = 1$.

Let n be sufficiently large. By (11.14), from $a^n b^{2n}$ for sufficiently large n we derive $a^m b^{2m}$ with $m = p + p' + (n-k-l)s$. Let $w = a^{n'}b^{2n'}$ be the word which generates $a^{m+1}b^{2(m+1)}$ (w has to have this form by Fact 4). By (11.14) we have

$$p + p' + (n'-k-l)s = m + 1 = p + p' + (n-k-l)s + 1.$$

Thus $(n'-n)s = 1$. This can only hold iff $s = 1$ and $n' = n+1$.

Analogously, we show $t = 1$.

Let $a^n b^{2n} \Longrightarrow a^m b^{2m}$ and $a^{2n}b^n \Longrightarrow a^{2m'}b^{m'}$. Then we have

$$\begin{aligned} m &= p + p' + (n-k-l), & 2m &= q + q' + (2n-k-l), \\ 2m' &= p + p' + (2n-k-l), & m' &= q + q' + (n-k-l). \end{aligned}$$

by (11.14), (11.15) and Fact 5. By an easy calculation one gets $p + p' = q + q'$ and then

$$m = 2m - m = q + q' + (2n-k-l) - (p + p' + (n-k-l) = n.$$

Therefore we only generate a finite language in contrast to the infinity of $L = L(G)$.

11.2.2 Some Results on Lindenmayer Systems with Interaction

For $k \in \mathbb{N}_0$ and $l \in \mathbb{N}_0$, by $\mathcal{L}(\langle k,l\rangle L)$ we denote the family of all languages generated by $\langle k,l\rangle$L systems. Further we set

$$\mathcal{L}(IL) = \bigcup_{k\geq 0, l\geq 0} \mathcal{L}(\langle k,l\rangle L).$$

From the definitions we get directly the following statement.

Corollary 11.39 *i) $\mathcal{L}(\langle 0,0\rangle L) = \mathcal{L}(0L)$.*
ii) $\mathcal{L}(\langle k,l\rangle L) \subseteq \mathcal{L}(\langle k',l'\rangle L) \subseteq \mathcal{L}(IL)$ for any $k,k',l,l' \in \mathbb{N}_0$, $k \leq k'$ and $l \leq l'$. □

First we study the relations between families of Lindenmayer languages with interaction and languages of the Chomsky hierarchy.

Lemma 11.40 *For any recursively enumerable language $L \subseteq T^*$ there is a $\langle 1,1\rangle L$ system G such that $L(G) \cap T^* = L$.*

Proof. Let L be a recursively enumerable language. Then $L = L(H)$ for some grammar $H = (N, T, P, S)$ in Kuroda normal form (see Theorem 2.19). With any production $p = AB \to CD \in P$ we associate the two new letters $A_{l,p}$ and $B_{r,p}$. We define

$$\begin{aligned}
N' &= \{A' \mid a \in N\},\\
N_l &= \{A_{l,p} \mid p = AB \to CD \in P\},\\
N_r &= \{B_{r,p} \mid p = AB \to CD \in P\},\\
V &= N \cup N' \cup N_l \cup N_r \cup T \cup \{F\},\\
V' &= V \cup \{\$\},\\
P_T &= \{(u, a, v) \to a \mid a \in T, u, v \in V'\},\\
P_N &= \{(u, A, v) \to \overline{A} \mid \overline{A} = A' \text{ or } \overline{A} = A_{r,p} \text{ or } \overline{A} = A_{l,q} \text{ for some } p, q \in P,\ u, v \in V'\},\\
P_{N'} &= \{(u, A', v) \to A \mid A \in N, u, v \in V'\} \cup \{(u, A', v) \to w \mid A \to w \in P, u, v \in V'\},\\
P_{r,l} &= \{(u, A_{l,p}, B_{r,p}) \to C \mid p = AB \to CD \in P, u \in V'\}\\
&\quad \cup \{(A_{l,p}, B_{r,p}, v) \to D \mid p = AB \to CD \in P, v \in V'\}\\
&\quad \cup \{(u, A_{l,p}, v) \to F \mid u \in V', v \in V' \setminus \{B_{r,p}\}\}\\
&\quad \cup \{(u, B_{r,p}, v) \to F \mid u \in V' \setminus \{A_{l,p}\}, v \in V'\},\\
P' &= P_T \cup P_N \cup P_{N'} \cup P_{r,l} \cup \{(u, F, v) \to F^2 \mid u, v \in V\}
\end{aligned}$$

and consider the $\langle 1,1\rangle$L system $G = (V, \$, P', S)$.

Let w be a sentential form generated by H and assume that $w \in L(G)$ (note that these requirements hold for the axiom) and let $w \Longrightarrow_H w'$ by an application of the rule $p = AB \to CD \in P$. Then we replace the occurrences of A and B to which p is applied by $A_{l,p}$ and $B_{r,p}$, respectively, all remaining nonterminals E by the associated E' and any terminal a by a. This corresponds to a derivation step in G which yields a word w''. To any occurrence of a symbol E' in w'' we apply $(u, E', v) \to E$, to any terminal a in w'' we apply $(u, a, v) \to a$, and we apply $(u, A_{l,p}, B_{r,p}) \to C$ and $(A_{l,p}, B_{r,p}, v) \to D$. This leads to w'. Analogously, we can prove that derivation steps in H with an application of rules of the forms $A \to B$ or $A \to a$ or $A \to \lambda$ can be simulated in G. Thus any sentential form of H belongs to $L(G)$, too. Since $L(H)$ is the intersection of all sentential forms of H with T^*, we have $L(H) \subseteq L(G) \cap T^*$.

Conversely, by arguments as above, it is easy to see that word obtained by an even number of derivation steps in G is a sentential form of H or it contains the letter F and that any word obtained by a odd number of derivation steps in G contains at least one symbol of $V \setminus (N \cup T)$. Thus $L(G) \cap T^* \subseteq L(H)$.

Therefore, $L(G) \cap T^* = L(H) = L$. □

Theorem 11.41 *The diagram of Figure 11.7 holds.*

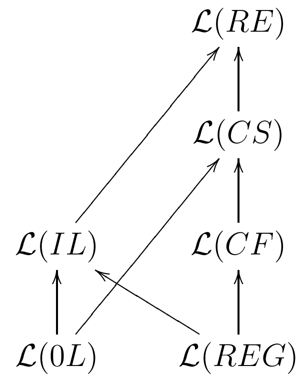

Figure 11.7: Relations between families of Lindenmayer languages with interaction and languages of the Chomsky hierarchy

Proof. i) $\mathcal{L}(REG) \subset \mathcal{L}(IL)$.
By Theorem 11.13, there exists a 0L language L which is not regular. Since $L \in \mathcal{L}(IL)$ by Corollary 11.39, we have a language in $\mathcal{L}(IL) \setminus \mathcal{L}(REG)$. Thus it is sufficient to prove the inclusion $\mathcal{L}(REG) \subseteq \mathcal{L}(IL)$.

Assume that $K \subset V^*$ is a regular language. Then K is accepted by a deterministic finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$. Let $n = \#(Z)$.

We first note that K contains a word whose length is at most n. Assume the contrary, i. e., the shortest word w of K has a length $r \geq n+1$. Let $w = a_1a_2\dots a_r$. We consider the states $z_i = \delta(z_0, a_1a_2\dots a_i)$ for $1 \leq i \leq r$. We have at least $n+1$ elements z_i, but only n states. Thus there are two numbers i and j, $1 \leq i < j \leq r$ such that $z_i = z_j$. By $w \in K$, we have

$$\delta(z_0, a_1\dots a_r) = \delta(\delta(z_0, a_1\dots a_j), a_{j+1}\dots a_r) = \delta(z_j, a_{j+1}\dots a_r) \in F\,.$$

Furthermore,

$$\begin{aligned}\delta(z_0, a_1\dots a_i a_{j+1}\dots a_r) &= \delta(\delta(z_0, a_1\dots a_i), a_{j+1}\dots a_r)\\ &= \delta(z_i, a_{j+1}\dots a_r)\\ &= \delta(z_j, a_{j+1}\dots a_r) \in F\,.\end{aligned}$$

Therefore $v = a_1a_2\dots a_i a_{j+1}a_{j+2}\dots a_r \in K$ and $|v| = r - (j-i) < r$ which contradicts the choice of w as a shortest word in K.

Analogously, we prove that, for any state z, there is a word w of length at most n with $\delta(z, w) = z$ or there is no word v with $\delta(z, v) = z$.

Now we construct the $\langle n+1, n\rangle$L system $H = (V, \$, P, \omega)$ where ω is one word in K with length at most n and P consists of all rules of the form

a1) $(\$^{n+1}, b_1, b_2b_3\dots b_s\$^{n-s+1}) \to w$,
where $s \leq n$ and w is a word of K of length at most $2n$,

a2) $(\$^{n-r+1}b_1b_2\dots b_r, b_{r+1}, b_{r+2}b_{r+3}\dots b_s\$^{n-s+r}) \to \lambda$,
where $r+1 \leq s \leq n$

(by rules of these types we generate all words of K of length at most $2n$ from a word of length at most n),

b1) $(\$^{n+1}, a_0, a_1a_2\dots a_n) \to w$,
where $a_i \in V$ for $0 \le i \le n$ and $w = a_0a_1\dots a_tva_{t+1}a_{t+2}\dots a_n$
for some $v \in V^*$ with $|v| \le n$, $\delta(z_0, a_0\dots a_t) = \delta(z_0, a_0\dots a_tv)$,

b2) $(\$^{n-r+1}c_1c_2\dots c_r, a, d_1d_2\dots d_n) \to \lambda$,
where $1 \le r \le n$, $c_i \in V$ for $1 \le i \le r$, $d_i \in V \cup \{\$\}$ for $1 \le i \le n$,
$d_1d_2\dots d_n \in V^s\{\$\}^{n-s}$, $r + s \ge n$

b3) $(c_1c_2\dots c_{n+1}, a, d_1d_2\dots d_n) \to a$,
where $c_i \in V$ for $1 \le i \le n+1$, $d_i \in V \cup \{\$\}$ for $1 \le i \le n$,
$d_1d_2\dots d_n \in V^s\{\$\}^{n-s}$, $s \ge 0$

(by these rules, for a word x of length at most $n+1$, i. e., $x = a_0a_1\dots a_nx'$, we have a derivation

$$a_0a_1\dots a_ta_{t+1}a_{t+2}\dots a_nx' \Longrightarrow a_0a_1\dots a_tva_{t+1}a_{t+2}\dots a_nx' \tag{11.16}$$

where v is an arbitrary word with

$$\delta(z_0, a_0\dots a_t) = \delta(z_0, a_0\dots a_tv) \text{ and } |v| \le n\,.) \tag{11.17}$$

We now prove that $L(H) \subseteq K$. By definition, the start word belongs to K. Moreover, all words generated from the start word by an application of rules of type a1) and a2) yield a word of K, and rules of types b1), b2) and b3) cannot be applied to the start word. Further, if $x \in K$ and we apply rules of type b1), b2) and b3) to x, then

$$\delta(z_0, a_0a_1\dots a_ta_{t+1}a_{t+2}\dots a_nx') = \delta(z_0, a_0a_1\dots a_tva_{t+1}a_{t+2}\dots a_nx')$$

which implies that the generated word $a_0a_1\dots a_tva_{t+1}a_{t+2}\dots a_nx'$ belongs to $T(\mathcal{A}) = K$, too. Thus we produce only words of K.

Conversely, $K \subseteq L(H)$ also holds. This can easily be proved by induction on the length of the words of K. If $w \in K$ has a length at most $2n$, then w can be produced by a1) and a2) applied to the start word. Thus the induction basis is satisfied. If $w \in K$ has a length r with $r > 2n$, i. e., $w = e_1e_2\dots e_{n+1}v$, then there are integers i and j with $1 \le i < j \le n+1$ and $\delta(z_0, e_1e_2\dots e_i) = \delta(z_0, e_1e_2\dots e_j)$. Thus $w' = e_1e_2\dots e_ie_{j+1}e_{j+2}\dots e_{n+1}v$ belongs to K. By induction hypothesis, $w' \in L(H)$. Now we are able to produce w from w' by an applications of rules of type b1), b2) and b3). Therefore $w \in L(H)$.

ii) $\mathcal{L}(0L) \subset \mathcal{L}(IL)$.
The inclusion holds by definition. Since, by Theorem 11.13, there is a regular language R which is not in $\mathcal{L}(0L)$. By part i) of this proof $R \in \mathcal{L}(IL) \setminus \mathcal{L}(0L)$ holds. Thus the inclusion is proper.

iii) $\mathcal{L}(IL)$ and $\mathcal{L}(CF)$ are incomparable.
Since $\mathcal{L}(0L)$ contains a non-context-free language, it follows that $\mathcal{L}(IL)$ as a superset of $\mathcal{L}(0L)$ contains a non-context-free language.

On the other hand by Example 11.38 the context-free language

$$\{a^nb^{2n} \mid n \ge 1\} \cup \{a^{2n}b^n \mid n \ge 1\}$$

is not a $\langle k,l\rangle$L language for any $k \in \mathbb{N}_0$ and $l \in \mathbb{N}_0$.

iv) $\mathcal{L}(IL) \subset \mathcal{L}(RE)$.
In analogy to the proof that any 0L language can be generated by a phrase structure grammar, we can show that any $\langle k,l\rangle$L language is in $\mathcal{L}(RE)$. Therefore $\mathcal{L}(IL) \subseteq \mathcal{L}(RE)$. The strictness of this inclusion follows from the Example 11.38.

v) $\mathcal{L}(IL)$ and $\mathcal{L}(CS)$ are incomparable.

The existence of a context-sensitive language which is not in $\mathcal{L}(IL)$ follows by Example 11.38.

Now let M be a set with $M \in \mathcal{L}(RE)$ and $M \notin \mathcal{L}(CS)$. Such a set exists by the proper inclusion of $\mathcal{L}(CS)$ in $\mathcal{L}(RE)$ (see Theorem 5.3). By Lemma 11.40, there is a $\langle 1,1\rangle$L system G and a set T with $L(G) \cap T^* = M$. If $L(G)$ is context-sensitive, then $M \in \mathcal{L}(CS)$ by the known closure of $\mathcal{L}(CS)$ under intersection by regular sets (see Theorem 4.6). Thus $L(G) \notin \mathcal{L}(CS)$. Therefore $\mathcal{L}(IL)$ contains a non-context-sensitive language. □

We now compare the families $\mathcal{L}(\langle k,l\rangle L)$ with each other.

Lemma 11.42 *For any $k,k',l,l' \in \mathbb{N}$ with $k+l = k'+l'$, $\mathcal{L}(\langle k,l\rangle L) = \mathcal{L}(\langle k',l'\rangle L)$.*

Proof. We first prove $\mathcal{L}(\langle k,l\rangle L) = \mathcal{L}(\langle k+1,l-1\rangle L)$ for $k \geq 1$ and $l \geq 2$. Let $G = (V,\$,P,\omega)$ be a $\langle k,l\rangle$L system. Then we construct the $\langle k+1,l-1\rangle$L system $G' = (V,\$,P',\omega)$ where P' consists of all rules of the form

- $(\$^{k+1},a,v) \to \lambda$ where $|v| = l-1$,
- $(ub,a,v) \to w$ where $(u,b,av) \to w \in P$, $|u| = k$, $|v| = l-1$, $v \neq \$^{l-1}$,
- $(cub,a,\$^{l-1}) \to w_1w_2$ where $(cu,b,a\$^{l-1}) \to w_1 \in P$, $(ub,a,\$^l) \to w_2 \in P$, $|c| = 1$, $|u| = k-1$.

Obviously, $z \underset{G}{\Longrightarrow} z'$ if and only if $z \underset{G'}{\Longrightarrow} z'$. The only difference is that in G' the first letter is replaced by λ, the ith letter is replaced by w in G' if and only if the $(i-1)$st letter is replaced by w in G, and the last letter is replaced by w_1w_2 in G' if and only if the last two letters are replaced by w_1 and w_2, respectively, in G. Therefore, $L(G) = L(G')$.

By an iterated application of equalities of this type, we get

$$\mathcal{L}(\langle k,1\rangle L) = \mathcal{L}(\langle k-1,2\rangle L) = \mathcal{L}(\langle k-2,3\rangle L) = \cdots = \mathcal{L}(\langle 1,k\rangle L).$$

□

For $k \geq 2$, we set $\mathcal{L}(kL) = \mathcal{L}(\langle 1,k-1\rangle L)$.
By Lemma 11.42, $\mathcal{L}(kL) = \mathcal{L}(\langle s,r\rangle L)$ for any $s \in \mathbb{N}$ and $r \in \mathbb{N}$ with $s+r = k$.

Lemma 11.43 *For any $k,k',l,l' \in \mathbb{N}_0$ with $k \leq k'$, $l \leq l'$ and $k+l\langle k'+l'$,*

$$\mathcal{L}(\langle k,l\rangle L) \subset \mathcal{L}(\langle k',l'\rangle L).$$

Proof. For a proof of this lemma, we refer to [11]. □

The following theorem relates the families $\mathcal{L}(\langle k,l\rangle L)$ to each other.

Theorem 11.44 *The diagram of Figure 11.8 holds.*

$$\begin{tikzcd}
 & \vdots & \\
\vdots & \mathcal{L}(4L) \arrow[u] & \vdots \\
\mathcal{L}(\langle 3,0\rangle L) \arrow[u] \arrow[ur] & \mathcal{L}(3L) \arrow[u] & \mathcal{L}(\langle 0,3\rangle L) \arrow[u] \arrow[ul] \\
\mathcal{L}(\langle 2,0\rangle L) \arrow[u] \arrow[ur] & \mathcal{L}(2L) \arrow[u] & \mathcal{L}(\langle 0,2\rangle L) \arrow[u] \arrow[ul] \\
\mathcal{L}(\langle 1,0\rangle L) \arrow[u] \arrow[ur] & & \mathcal{L}(\langle 0,1\rangle L) \arrow[u] \arrow[ul] \\
 & \mathcal{L}(0L) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Figure 11.8: Relations between families of Lindenmayer languages with interaction

Proof. All inclusions and their strictnesses follow by Lemmas 11.42 and 11.43.

We now prove the existence of a language $L \in \mathcal{L}(\langle 1,0\rangle L)$ which is not contained in $\mathcal{L}(\langle 0,l\rangle L)$ for any $l \geq 1$. This shows that no family of the left chain is contained in some family of the right chain.

Let

$$L = \{c\} \cup \{a^{2^n} \mid n \geq 0\} \cup \{ba^{2^n+1} \mid n \geq 0\}.$$

By Example 11.35, $L = L(G_7)$ for the $\langle 1,0\rangle$L system G_7. Therefore $L \in \mathcal{L}(\langle 1,0\rangle L)$.

Now assume that $L \in \mathcal{L}(\langle 0,l\rangle L)$ for some $l \geq 1$. Let $G = (\{a,b,c\}, \$, P, \omega)$ be the $\langle 0,l\rangle$L system generating L. It is easy to see that $(a,v) \to w_{a,v} \in P$ and $(b,v) \to w_{b,v} \in P$ imply $w_{a,v} \in a^*$ and $w_{b,v} \in ba^*$ (otherwise, e. g., a^{2^n}, $n \geq l$, would derive a word with at least two occurrences of b). Moreover, for any v, $w_{a,v}$ and $w_{b,v}$ are uniquely determined. E.g., if $(a,a^l) \to w_1$ and $(a,a^l) \to w_2$, then we derive $w_1' = w_1 w$ and $w_2' = w_2 w$ from a^{2^n} with sufficiently large n where w originates from the last $2^n - 1$ letters. Since $||w_1'| - |w_2'|| = ||w_1| - |w_2||$ and the length between different words over $\{a\}$ in L grows unbounded, we obtain a contradiction.

Let $(a,a^l) \to a^r$. If $r = 0$, then we cannot generate words with an unbounded number of occurrences of a. If $r = 1$, then the increase of the length originates only from the first letter b and/or the last l letters such that the increase is bounded in contrast to the structure of the words of L.

Now assume that $a^{2^n} \Longrightarrow a^{2^m}$ with $m \geq n$ and $(b,a^l) \to ba^s$. Then $baa^{2^n} \Longrightarrow ba^s a^r a^{2^m} = ba^{2^m+r+s}$. Thus $r + s = 1$ which gives $r \leq 1$ which is impossible as shown above.

Hence in all cases we got a contradiction which shows $L \notin \mathcal{L}(\langle 0,l\rangle L)$.

Taking L^R, by analogous arguments one can show that $L^R \in \mathcal{L}(\langle 0,1\rangle L)$ and $L^R \notin \mathcal{L}(\langle k,0\rangle L)$ for any $k \in \mathbb{N}$ which proves that no family of the right chain is contained in some family of the left chain.

We omit the proof of the incomparability of $\mathcal{L}(\langle k,0\rangle L)$ with $\mathcal{L}(kL)$ and that of $\mathcal{L}(\langle 0,k\rangle L)$ with $\mathcal{L}(kL)$. □

Finally, we present some results on topics which we studied in Sections 11.1.4, 11.1.5and 11.1.6 for 0L systems. We omit the exact formal definitions of adult languages and growth functions of Lindenmayer systems with interaction. They can given by a straightforward translation from the concepts for (deterministic) 0L systems.

We start with a characterization of adult languages of L systems with interaction.

By $\mathcal{L}(AIL)$ we denote the family of all adult languages which can be generated by $\langle k,l\rangle$L systems with $k\in\mathbb{N}_0$ and $l\in\mathbb{N}_0$.

Theorem 11.45 $\mathcal{L}(AIL)=\{L \mid L\in\mathcal{L}(RE)$ *and* $\lambda\notin L\}$.

Proof. Let L be an arbitrary language of $\mathcal{L}(RE)$ such that $\lambda\notin L$. We consider the $\langle 1,1\rangle$L system constructed G constructed in the proof of Lemma 11.40. It is easy to see that $L_A(G)=(L(G)\cap T^*)=L$. Thus $\{L \mid L\in\mathcal{L}(RE)$ and $\lambda\notin L\}\subseteq\mathcal{L}(AIL)$.

Let H be an arbitrary $\langle k,l\rangle$L system. Then $L(H)\in\mathcal{L}(RE)$ by Theorem 11.41. We construct a Turing machine M which checks for a word w whether or not w derives only w according to the rules of H (as in the case of 0L system, if $w\Longrightarrow w$ is the only derivation from w, then there is exactly one rule for any letter and its context, and thus M has only to simulate the derivation and reject if there are more rules or one does not get w). Because Turing machines accept recursively enumerable languages, we have $T(M)\in\mathcal{L}(RE)$. Since $L_A(H)=L(H)\cap T(M)$ and $\mathcal{L}(RE)$ is closed under intersection (see Theorem 4.3), we get $L_A(H)\in\mathcal{L}(RE)$. Therefore $\mathcal{L}(AIL)\subseteq\mathcal{L}(RE)$. Because adult languages do not contain the empty word, we have that $\mathcal{L}(AIL)\subseteq\{L \mid L\in\mathcal{L}(RE)$ and $\lambda\notin L\}$. □

By Theorem 11.41 and Theorems 11.20 and 11.45, we know that 0L systems generate a smaller family of languages and a smaller family of adult languages than L systems with interaction. We now show that this also holds with respect to growth functions.

Theorem 11.46 *There is a deterministic $\langle 1,0\rangle L$ system G such that its growth function is not a growth function of a D0L system. More precisely, f_G is not bounded by a constant and, for any polynomial p with $p(m)\geq m$ for all $m\geq m_0$ for some $m_0\in\mathbb{N}$,*

$$\lim_{m\to\infty}\frac{f_G(m)}{p(m)}=0\,.$$

Proof. We consider the $\langle 1,0\rangle$L system G_9 of Example 11.37. In Example 11.37, we have shown that $L(G_9)$ is infinite. Therefore f_{G_9} cannot be bounded by a constant.

Considering (11.12) and (11.13) we see that at least m derivation steps are necessary in order to get a length extension of a word of length m by one. Thus we need at least $1+2+3+\cdots+m$ steps in order to obtain a word of length $m+1$. Therefore we get

$$f_{G_9}(\frac{m(m+1)}{2})\leq m+1$$

or

$$f_{G_9}(m)\leq -\frac{1}{2}+\sqrt{\frac{1+8m}{4}}\leq\sqrt{2m}\,.$$

Therefore we get

$$\lim_{m\to\infty} \frac{f_{G_9}}{p(m)} \le \lim_{m\to\infty} \frac{\sqrt{2m}}{m} = 0\,.$$

By Theorem 11.32, f_{G_9} grows slower than any unbounded growth function of a D0L system. Hence f_{G_9} is not a growth function of a D0L system. □

We have seen that the growth of a DIL system can be smaller than any polynomial. However, for PDIL systems, there is a lower bound. Any unbounded growth function of a PDIL system is bounded from below by the logarithm function. More precisely, we have the following theorem.

Theorem 11.47 *For any PDIL system $G = (V, \S, P, \omega)$ with $r = \#(V)$ and an unbounded growth function f_G,*

$$\lim_{n\to\infty} \frac{f_G(n)}{\log_r(n)} \ge 1.$$

Proof. Let $f : \mathbb{N} \to \mathbb{N}$ such that $f(t)$ gives the length of the tth non-empty word over V in the lexicographic order. Since we have r^n words of length n, we have

$$r + r^2 + r^3 + \cdots + r^{n-1} = \frac{r^n - r}{r-1}$$

words with a length at most $n-1$. This implies $f(n) = \lfloor \log_r(n(r-1)+r) \rfloor$.

Let $\omega = w_0 \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \ldots$ be the unique derivation with respect to G. Since G is propagating, we have $|w_n| \le |w_{n+1}|$ for $n \ge 0$. Thus $f_G(n) \ge f(n)$. Therefore we get

$$\lim_{n\to\infty} \frac{f_G(n)}{\log_r(n)} \ge \lim_{n\to\infty} \frac{f(n)}{\log_r(n)} = \lim_{n\to\infty} \frac{\lfloor \log_r(n(r-1)+r) \rfloor}{\log_r(n)} \ge 1.$$

□

Finally we discuss the decidability status of some decision problems for IL systems.

Obviously, the membership problem is undecidable for IL systems. This can be seen as follows. Let $H = (N, T, P, S)$ be an arbitrary phrase structure grammar and w a word over T. Then we can construct an IL system G such that $L(G) \cap T^* = L(H)$. Thus $w \in L(H)$ if and only if $w \in L(G)$. Because the membership problem for arbitrary phrase structure grammars is undecidable, the membership problem for IL systems is undecidable, too.

We now improve this statement and show that the decision problems which are of interest for IL systems are already undecidable for deterministic IL systems.

Theorem 11.48 *i) The membership problem for deterministic IL systems is undecidable.*
ii) The finiteness problem for deterministic IL systems is undecidable.
iii) The equivalence problem for deterministic IL systems is undecidable.

Proof. i) Let $\mathcal{M} = (X, Z, z_0, Q, \delta)$ be a deterministic Turing machine and w a word over X. We construct the deterministic $\langle 2, 2\rangle$L system $G = (V \cup Z \cup \{*\}, P, z_0 w')$ with

$$w' = \begin{cases} w & \text{if } w \ne \lambda \\ * & \text{if } w = \lambda \end{cases},$$

and

$$\left.\begin{array}{l} (\$\$, z, ax) \to z'* \\ (y'y, z, ax) \to z'y \\ (y'x', y, za) \to \lambda \\ (y'z, a, xx') \to a' \end{array}\right\} \quad \text{for} \quad \left\{\begin{array}{l} \delta(z,a) = (z', a', L), z \in Z \setminus Q,\ a \in X \cup \{*\}, \\ y \in X \cup \{*\},\ x, x', y' \in X \cup \{*, \$\}, \end{array}\right.$$

$$\left.\begin{array}{l} (x'y, z, ax) \to z' \\ (yz, a, xx') \to a' \end{array}\right\} \quad \text{for} \quad \left\{\begin{array}{l} \delta(z,a) = (z', a', N), z \in Z \setminus Q, a \in X \cup \{*\}, \\ x, x', y \in X \cup \{*, \$\}, \end{array}\right.$$

$$\left.\begin{array}{l} (y'y, z, ax') \to a'z' \\ (yz, a, \$\$) \to * \\ (yz, a, xx') \to \lambda \end{array}\right\} \quad \text{for} \quad \left\{\begin{array}{l} \delta(z,a) = (z', a', R), z \in Z \setminus Q,\ a \in X \cup \{*\}, \\ x \in V \cup *,\ x', y \in X \cup \{*, \$\}, \end{array}\right.$$

$$\left.\begin{array}{l} (\$\$, q\$\$) \to \lambda \\ (yq, a, xx') \to \lambda \\ (qb, a, xx') \to \lambda \\ (x'x, a, qy) \to \lambda \\ (x'x, a, bq) \to \lambda \end{array}\right\} \quad \text{for} \quad q \in Q,\ a, b \in X \cup \{*\}, x, x'y, \in X \cup \{*, \$\}$$

(we mention that not all combinations of letter from $V \cup \{*\}$ and \$ given by the conditions can occur, e.g. the right marker can not be followed by a symbol of $V \cup \{*\}$) and $(xy, a, x'y') \to a$ in all remaining cases.

Clearly, z_0w' gives a description of the initial configuration (λ, z_0, w) of the Turing machine. Furthermore, it is easy to see that $vzv' \Longrightarrow uz'u'$ in G if and only if $\mathcal{M}$ transforms the configuration (v, z, v') in (u, z', u') as long as $z \in Z \setminus Q$. If q is in Q, then in the word uqu' we cancel the last two letters of u and the first two letters of u' which are in $V \cup \{*\}$. If q has \$ as the right and the left neighbour, then we cancel q. Thus starting from uqu' with $q \in Q$ we derive the empty word. Therefore we generate the empty word if and only if we reach a halting state from Q in $\mathcal{M}$, i.e., if $\mathcal{M}$ stops on the input w. Hence the decidability of membership problem for DIL systems implies the decidability of the halting problem for Turing machine. Now Theorem 3.31 gives the undecidability of membership problem for DIL systems.

ii) Let $\mathcal{M} = (X, Z, z_0, Q, \delta)$ be a deterministic Turing machine and w a word over X. We construct - in analogy to the preceding proof - the deterministic $\langle 2, 2\rangle$L system

$$G' = (X \cup \{*, d, d_1, d_2\},\ P',\ z_0w'ddd_1),$$

where P' consist of the rules of P where we use d like a right marker as long as steps of $\mathcal{M}$ are simulated and d, d_1, d_2 are also cancelled if a halting state is reached and the additional rules

$$(xy, d_1, \$\$) \to d_1d_2 \text{ and } (xd_1, d_2, \$\$) \to d_1$$

(again, we mention only those rules which change the letter under consideration).

If the Turing machine has performed $2n-1$ or $2n$ steps where $n \geq 1$ and the configuration (u, z, v) or (u', z', v') is obtained, then G' derives in $2n-1$ or $2n$ the word $uzvddd_1^nd_2$ or $u'z'v'd_1^{n+1}$. Obviously, if no halting state is reached, then the language $L(G')$ is infinite (since there are infinitely many tails d_1^n or $d_1^nd_2$). If a halting state obtained after a certain finite number of steps (note that $\mathcal{M}$ is deterministic), then after a certain number of steps the empty word is obtained because more letter are cancelled than newly introduced. Thus $L(G')$ is infinite if and only if $\mathcal{M}$ does not halt on w. Thus the finiteness problem for DIL systems is undecidable.

iii) We omit the proof and refer to [31] $\square$