Generalized Random Context Picture Grammars: The State of the Art

Sigrid Ewert and Max Rabkin

School of Computer Science, University of the Witwatersrand, Johannesburg
sigrid.ewert@wits.ac.za, max.rabkin@students.wits.ac.za

Abstract. Generalized random context picture grammars (grcpgs) are a method of syntactic picture generation. The terminals are subsets of the Euclidean plane and the replacement of variables involves the building of functions that will eventually be applied to terminals. Context is used to permit or forbid production rules.

Iterated function systems (IFSs) and their generalization, mutually recursive function systems (MRFSs), are among the best-known methods for constructing fractals. In earlier work it was shown that any picture sequence generated by an IFS or MRFS can be generated by a grcpg. Moreover, it was shown that grcpgs can generate a wider range of pictures than IFSs or MRFSs.

In this essay we give a summary of the above mentioned results. We then consider language-restricted iterated function systems (LRIFSs), a method of picture generation where a language controls which functions of an IFS are applied. We first show that LRIFSs are more powerful than IFSs. Then we show that any picture produced by an LRIFS where the restricting language is regular, can be approximated by a grcpg.

1 Introduction

A method of syntactic picture generation, using random context picture grammars (rcpgs), was described and studied elsewhere [6–9]. A summary of results can be found in [5]. In [10], Ewert and van der Walt introduced the notion of a generalized random context picture grammar (grcpg). These grammars use production rules to compose functions from some finite set of functions. These functions are then applied to terminals, which are subsets of the Euclidean plane, to create a picture. Context is used to permit or forbid production rules.

An iterated function system (IFS) is an iterative method for constructing fractals from a finite set of contractive maps defined on a complete metric space. The sequence of pictures generated by an IFS converges to a unique limit. The method was developed principally by Barnsley and co-workers, who obtained impressively life-like images both of nature scenes and the human face [1, 2]. Ewert and van der Walt [10] showed that any picture sequence generated by an IFS can also be generated by a grcpg that uses forbidding context only. Moreover, since grcpgs use context to control the sequence in which functions are applied, they can generate a wider range of fractals or, more generally, pictures than IFSs.

H. Bordihn, M. Kutrib, and B. Truthe (Eds.): Dassow Festschrift 2012, LNCS 7300, pp. 56–74, 2012.

Mutually recursive function systems (MRFSs), called hierarchical iterated function systems by Peitgen and co-workers [13], are powerful methods of mathematical picture generation. MRFSs are a generalization of IFSs, and consist of networks or hierarchies of IFSs. Kruger and Ewert [12] generalized the above mentioned result for IFSs to show that for every MRFS, an equivalent grcpg can be constructed. They also showed that grcpgs are more general than MRFSs, in the sense that grcpgs can be constructed that generate sets of pictures that cannot be generated by any MRFS.

Language-restricted iterated function systems (LRIFSs) [15] are a generalization of IFSs, and consist of an IFS and a language that controls which functions of the IFS are applied. In this essay we first show that LRIFSs are more powerful than IFSs. Then we show that any picture produced by an LRIFS where the restricting language is regular, can be approximated by a grcpg.

The remainder of this paper is structured as follows. In Sect. 2, we review published results about the relationship between grcpgs and IFSs, and MRFSs, respectively. In Sect. 3 we focus on LRIFSs and in particular show that any picture produced by an LRIFS where the restricting language is regular, can be approximated by a grcpg. Future work is recommended in Sect. 4.

2 Previously Published Results

In this section we give the definitions of grcpgs, IFSs and MRFSs. Then we state the most important results about the relationship between grcpgs and IFSs, and grcpgs and MRFSs.

2.1 Generalized Random Context Picture Grammars

We define a generalized random context picture grammar and illustrate the main concepts with an example, the iteration sequence of the Sierpiński gasket.

Definition 1. *Let S be any set. Then $\wp(S)$ denotes the power set of S.*

Definition 2. *A generalized random context picture grammar $G = (V_N, V_T, V_F, P, (S, \epsilon))$ has a finite alphabet V of labels, consisting of disjoint subsets V_N of variables, V_T of terminals and V_F of function identifiers. The productions, finite in number, are of the form $A \to \{(A_1, \rho_1), (A_2, \rho_2), \ldots, (A_t, \rho_t)\}\ (\mathfrak{P}; \mathfrak{F})$, where $A \in V_N$, $A_1, \ldots, A_t \in V_N \cup V_T$, $\rho_1, \ldots, \rho_t \in V_F^*$ and $\mathfrak{P}, \mathfrak{F} \subseteq V_N$. Finally, there is an initial configuration (S, ϵ), where $S \in V_N$ and ϵ denotes the empty string.*

Definition 3. *A pictorial form Π is a finite set $\{(B_1, \varphi_1), (B_2, \varphi_2), \ldots, (B_s, \varphi_s)\}$, where $B_1, \ldots, B_s \in V_N \cup V_T$ and $\varphi_1, \ldots, \varphi_s \in V_F^*$. We denote the set $\{B_1, \ldots, B_s\}$ by $l(\Pi)$.*

Definition 4. *For a grcpg G and pictorial forms Π and Γ we write $\Pi \Longrightarrow_G \Gamma$ if there is a production $A \to \{(A_1, \rho_1), (A_2, \rho_2), \ldots, (A_t, \rho_t)\}\ (\mathfrak{P}; \mathfrak{F})$ in G, Π*

contains an element (A,φ), $l(\Pi \setminus \{(A,\varphi)\}) \supseteq \mathfrak{P}$ *and* $l(\Pi \setminus \{(A,\varphi)\}) \cap \mathfrak{F} = \emptyset$, *and* $\Gamma = (\Pi \setminus \{(A,\varphi)\}) \cup \{(A_1,\varphi\rho_1),(A_2,\varphi\rho_2),\ldots,(A_t,\varphi\rho_t)\}$. *As usual,* $\Longrightarrow_{\mathrm{G}}^{*}$ *denotes the reflexive transitive closure of* $\Longrightarrow_{\mathrm{G}}$.

Definition 5. *A picture is a pictorial form* Π *with* $l(\Pi) \subseteq V_{\mathrm{T}}$.

Definition 6. *The gallery* $\mathcal{G}(G)$ *generated by a grcpg* G *is the set of pictures* Π *such that* $\{(S,\epsilon)\} \Longrightarrow_{\mathrm{G}}^{*} \Pi$.

Definition 7. *The gallery of a grcpg* G *is rendered by specifying functions* $\Psi_{\mathrm{G}} : V_{\mathrm{T}} \to \wp\left(\mathbb{R}^2\right)$ *and* $\Upsilon_{\mathrm{G}} : V_{\mathrm{F}} \to \mathrm{F}\left(\mathbb{R}^2\right)$, *where* $\mathrm{F}\left(\mathbb{R}^2\right) = \left\{g \mid g : \mathbb{R}^2 \to \mathbb{R}^2\right\}$. *This yields a representation of a picture* $\Pi = \{(B_1,\varphi_1),(B_2,\varphi_2),\ldots,(B_s,\varphi_s)\}$ *in* $\mathbb{R}^2$ *by*

$$r(\Pi) = \bigcup_{i=1}^{s} \Upsilon_{\mathrm{G}}(\varphi_i)\left(\Psi_{\mathrm{G}}(B_i)\right) \quad ,$$

where Υ_{G} *has been extended to* V_{F}^{*} *in the obvious manner,* $\Upsilon_{\mathrm{G}}(\epsilon)$ *representing the identity function* id.

Definition 8. *If every production in* G *has* $\mathfrak{P} = \emptyset$, *we call* G *a generalized random forbidding context picture grammar (grFcpg).*

Note 1. It should be clear that we can also use (S,id) as initial configuration without that affecting the rendered gallery.

Note 2. For the sake of convenience, we write a production $A \to \{(A_1,\epsilon)\}\ (\mathfrak{P};\mathfrak{F})$ as $A \to A_1\ (\mathfrak{P};\mathfrak{F})$. Moreover, if $\mathfrak{P} = \mathfrak{F} = \emptyset$ in a production $A \to \{(A_1,\rho_1),(A_2,\rho_2),\ldots,(A_t,\rho_t)\}\ (\mathfrak{P};\mathfrak{F})$, then we write $A \to \{(A_1,\rho_1),(A_2,\rho_2),\ldots,(A_t,\rho_t)\}$.

We illustrate these concepts with an example.

Example 1. We generate the typical iteration sequence of the Sierpiński gasket with the grcpg $G_{\mathrm{gasket}} = (\{S,T,U,F\},\{b\},\{g_{\mathrm{lb}},g_{\mathrm{rb}},g_{\mathrm{t}}\},P,(S,\epsilon))$, where P is the set:

$$S \to \{(T,g_{\mathrm{lb}}),(T,g_{\mathrm{rb}}),(T,g_{\mathrm{t}})\}\ (\emptyset;\{U\}) \tag{1}$$

$$T \to U\ (\emptyset;\{S,F\}) \mid \tag{2}$$

$$F\ (\emptyset;\{S,U,F\}) \mid \tag{3}$$

$$b\ (\{F\};\emptyset) \tag{4}$$

$$U \to S\ (\emptyset;\{T\}) \tag{5}$$

$$F \to b\ (\emptyset;\{T\}) \tag{6}$$

We give the derivation of a picture Π in $\mathcal{G}(G_{\mathrm{gasket}})$ in detail.

$$\begin{aligned}
&\{(S,\epsilon)\} \\
\Longrightarrow_{\mathrm{G}}\ &\{(T,g_{\mathrm{lb}}),(T,g_{\mathrm{rb}}),(T,g_{\mathrm{t}})\} && \text{(rule 1)}\\
\Longrightarrow^*_{\mathrm{G}}\ &\{(U,g_{\mathrm{lb}}),(U,g_{\mathrm{rb}}),(U,g_{\mathrm{t}})\} && \text{(thrice rule 2)}\\
\Longrightarrow^*_{\mathrm{G}}\ &\{(S,g_{\mathrm{lb}}),(S,g_{\mathrm{rb}}),(S,g_{\mathrm{t}})\} && \text{(thrice rule 5)}\\
\Longrightarrow^*_{\mathrm{G}}\ &\{(T,g_{\mathrm{lb}}g_{\mathrm{lb}}),(T,g_{\mathrm{lb}}g_{\mathrm{rb}}),(T,g_{\mathrm{lb}}g_{\mathrm{t}})\}\cup\\
&\{(T,g_{\mathrm{rb}}g_{\mathrm{lb}}),(T,g_{\mathrm{rb}}g_{\mathrm{rb}}),(T,g_{\mathrm{rb}}g_{\mathrm{t}})\}\cup\\
&\{(T,g_{\mathrm{t}}g_{\mathrm{lb}}),(T,g_{\mathrm{t}}g_{\mathrm{rb}}),(T,g_{\mathrm{t}}g_{\mathrm{t}})\} && \text{(thrice rule 1)}\\
\Longrightarrow_{\mathrm{G}}\ &\{(T,g_{\mathrm{lb}}g_{\mathrm{lb}}),(T,g_{\mathrm{lb}}g_{\mathrm{rb}}),(T,g_{\mathrm{lb}}g_{\mathrm{t}})\}\cup\\
&\{(T,g_{\mathrm{rb}}g_{\mathrm{lb}}),(F,g_{\mathrm{rb}}g_{\mathrm{rb}}),(T,g_{\mathrm{rb}}g_{\mathrm{t}})\}\cup\\
&\{(T,g_{\mathrm{t}}g_{\mathrm{lb}}),(T,g_{\mathrm{t}}g_{\mathrm{rb}}),(T,g_{\mathrm{t}}g_{\mathrm{t}})\} && \text{(rule 3)}\\
\Longrightarrow^*_{\mathrm{G}}\ &\{(b,g_{\mathrm{lb}}g_{\mathrm{lb}}),(b,g_{\mathrm{lb}}g_{\mathrm{rb}}),(b,g_{\mathrm{lb}}g_{\mathrm{t}})\}\cup\\
&\{(b,g_{\mathrm{rb}}g_{\mathrm{lb}}),(F,g_{\mathrm{rb}}g_{\mathrm{rb}}),(b,g_{\mathrm{rb}}g_{\mathrm{t}})\}\cup\\
&\{(b,g_{\mathrm{t}}g_{\mathrm{lb}}),(b,g_{\mathrm{t}}g_{\mathrm{rb}}),(b,g_{\mathrm{t}}g_{\mathrm{t}})\} && \text{(repeated application of rule 4)}\\
\Longrightarrow_{\mathrm{G}}\ &\{(b,g_{\mathrm{lb}}g_{\mathrm{lb}}),(b,g_{\mathrm{lb}}g_{\mathrm{rb}}),(b,g_{\mathrm{lb}}g_{\mathrm{t}})\}\cup\\
&\{(b,g_{\mathrm{rb}}g_{\mathrm{lb}}),(b,g_{\mathrm{rb}}g_{\mathrm{rb}}),(b,g_{\mathrm{rb}}g_{\mathrm{t}})\}\cup\\
&\{(b,g_{\mathrm{t}}g_{\mathrm{lb}}),(b,g_{\mathrm{t}}g_{\mathrm{rb}}),(b,g_{\mathrm{t}}g_{\mathrm{t}})\} && \text{(rule 6)}
\end{aligned}$$

Let $\varUpsilon_{\mathrm{G}}(g_{\mathrm{lb}}) = (x,y) \to \left(\frac{x}{2},\frac{y}{2}\right)$, $\varUpsilon_{\mathrm{G}}(g_{\mathrm{rb}}) = (x,y) \to \left(\frac{x}{2}+\frac{1}{2},\frac{y}{2}\right)$ and $\varUpsilon_{\mathrm{G}}(g_{\mathrm{t}}) = (x,y) \to \left(\frac{x}{2}+\frac{1}{4},\frac{y}{2}+\frac{\sqrt{3}}{4}\right)$.

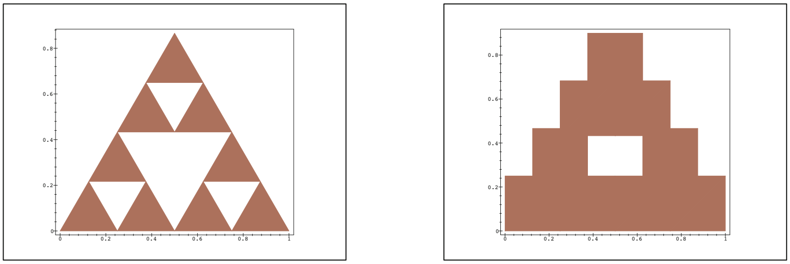
Then $r(\varPi) = \bigcup_{i=1}^{9} \varUpsilon_{\mathrm{G}}(\varphi_i)(\varPsi_{\mathrm{G}}(b))$, where $\varUpsilon_{\mathrm{G}}(\varphi_1) = (x,y) \to \left(\frac{1}{2}\times\frac{x}{2},\frac{1}{2}\times\frac{y}{2}\right)$, $\varUpsilon_{\mathrm{G}}(\varphi_2) = (x,y) \to \left(\frac{1}{2}\left(\frac{x}{2}+\frac{1}{2}\right),\frac{1}{2}\times\frac{y}{2}\right)$, $\varUpsilon_{\mathrm{G}}(\varphi_3) = (x,y) \to \left(\frac{1}{2}\left(\frac{x}{2}+\frac{1}{4}\right),\frac{1}{2}\left(\frac{y}{2}+\frac{\sqrt{3}}{4}\right)\right)$, ….

Let $\varPsi_{\mathrm{G}}(b)$ be the dark triangle with vertices $\left\{(0,0),(1,0),\left(\frac{1}{2},\frac{\sqrt{3}}{2}\right)\right\}$. Then $r(\varPi)$ represents the picture in Fig. 1(a). Alternatively, let $\varPsi_{\mathrm{G}}(b)$ be the dark square determined by the vertices $\{(0,0),(1,0),(1,1)\}$. Then $r(\varPi)$ represents Fig. 1(b).

2.2 Iterated Function Systems

Iterated function systems are among the best-known methods for constructing fractals. An extensive treatment of IFSs can be found in [11]. In this section we review results that show that grcpgs are more powerful than IFSs.

Definition 9. *An iterated function system* $\{X,\mathcal{F}\}$ *or* $\{X; f_1, f_2, \ldots, f_t\}$, $t>0$, *is a pair consisting of a complete metric space* X *together with a finite set of contractive maps* $f_i : X \to X$, $1 \le i \le t$.



(a) $\Psi_{\mathrm{G}}(\{b\})$ is a dark triangle

(b) $\Psi_{\mathrm{G}}(\{b\})$ is a dark square

Fig. 1. Two pictures in the iteration sequence of the Sierpiński gasket

Let $\mathcal{H}(X)$ be the set of all nonempty compact subsets of X. For $E \in \mathcal{H}(X)$, let $F(E) = f_1(E) \cup f_2(E) \cup \ldots \cup f_t(E)$. By repeated application of F to E, we obtain a sequence in $\mathcal{H}(X)$, $E_0 = E, E_1 = F(E_0), E_2 = F(E_1), \ldots$.

The sequence $E_0, E_1, E_2, \ldots$ converges to a unique limit $\mathcal{E}$, called the attractor of the IFS, which is independent of the choice of starting set E_0, but completely determined by the choice of the maps f_i.

This sequence can be generated by a grFcpg, as was shown in [10]. We state the full result here—in Theorem 1—since the proof gives the translation from a given IFS to a grFcpg.

Theorem 1. *Let $\{X, \mathcal{F}\}$ be an IFS. Then there is a grFcpg G such that for every $l \geq 1$, G generates the set $\left\{\left(a, \varphi_1^l\right), \left(a, \varphi_2^l\right), \ldots, \left(a, \varphi_{t^l}^l\right)\right\}$, where the φ_i^l are all t^l possible sequences of length l of the $f_j \in \mathcal{F}$.*

Proof. Let $G = (\{S, I, T, U, F\}, \{a\}, \{f_1, f_2, \ldots, f_t\}, P, (S, \epsilon))$, where P is the set:

$$
\begin{aligned}
S &\to \{(I, f_1), (I, f_2), \ldots, (I, f_t)\} \\
I &\to \{(T, f_1), (T, f_2), \ldots, (T, f_t)\}\ (\emptyset; \{F, U\})\ \mid \\
&\quad\ F\ (\emptyset; \{T, U\}) \\
T &\to U\ (\emptyset; \{I\}) \\
U &\to I\ (\emptyset; \{T\}) \\
F &\to a\ (\emptyset; \{I\})
\end{aligned}
$$

□

Example 2. We obtain the iteration sequence of the Sierpiński gasket with the IFS $\left\{\mathbb{R}^2; g_{\mathrm{lb}}, g_{\mathrm{rb}}, g_{\mathrm{t}}\right\}$, where $g_{\mathrm{lb}} : (x, y) \to \left(\frac{x}{2}, \frac{y}{2}\right)$, $g_{\mathrm{rb}} : (x, y) \to \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right)$ and $g_{\mathrm{t}} : (x, y) \to \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{\sqrt{3}}{4}\right)$.

For any $E \in \mathcal{H}(\mathbb{R}^2)$, $F(E) = g_{\mathrm{lb}}(E) \cup g_{\mathrm{rb}}(E) \cup g_{\mathrm{t}}(E)$. Let $E_0 = E$. Then $E_1 = F(E_0) = g_{\mathrm{lb}}(E_0) \cup g_{\mathrm{rb}}(E_0) \cup g_{\mathrm{t}}(E_0)$, $E_2 = F(E_1) = g_{\mathrm{lb}} g_{\mathrm{lb}}(E_0) \cup g_{\mathrm{lb}} g_{\mathrm{rb}}(E_0) \cup g_{\mathrm{lb}} g_{\mathrm{t}}(E_0) \cup g_{\mathrm{rb}} g_{\mathrm{lb}}(E_0) \cup g_{\mathrm{rb}} g_{\mathrm{rb}}(E_0) \cup g_{\mathrm{rb}} g_{\mathrm{t}}(E_0) \cup g_{\mathrm{t}} g_{\mathrm{lb}}(E_0) \cup g_{\mathrm{t}} g_{\mathrm{rb}}(E_0) \cup g_{\mathrm{t}} g_{\mathrm{t}}(E_0), \ldots$. When we choose E_0 to be a dark triangle, respectively, a dark square, E_2 is represented by Fig. 1(a) and Fig. 1(b), respectively.

To this IFS corresponds the grFcpg $G = (\{S, I, T, U, F\}, \{a\}, \{g_{\mathrm{lb}}, g_{\mathrm{rb}}, g_{\mathrm{t}}\}, P, (S, \epsilon))$, where P is the set:

$$
\begin{aligned}
& S \to \{(I, g_{\mathrm{lb}}), (I, g_{\mathrm{rb}}), (I, g_{\mathrm{t}})\} \\
& I \to \{(T, g_{\mathrm{lb}}), (T, g_{\mathrm{rb}}), (T, g_{\mathrm{t}})\} \ (\emptyset; \{F, U\}) \ | \\
& \qquad F \ (\emptyset; \{T, U\}) \\
& T \to U \ (\emptyset; \{I\}) \\
& U \to I \ (\emptyset; \{T\}) \\
& F \to a \ (\emptyset; \{I\})
\end{aligned}
$$

G generates the pictorial forms $\{(a, g_{\mathrm{lb}}), (a, g_{\mathrm{rb}}), (a, g_{\mathrm{t}})\}$, $\{(a, g_{\mathrm{lb}} g_{\mathrm{lb}}), (a, g_{\mathrm{lb}} g_{\mathrm{rb}}), (a, g_{\mathrm{lb}} g_{\mathrm{t}})\} \cup \{(a, g_{\mathrm{rb}} g_{\mathrm{lb}}), (a, g_{\mathrm{rb}} g_{\mathrm{rb}}), (a, g_{\mathrm{rb}} g_{\mathrm{t}})\} \cup \{(a, g_{\mathrm{t}} g_{\mathrm{lb}}), (a, g_{\mathrm{t}} g_{\mathrm{rb}}), (a, g_{\mathrm{t}} g_{\mathrm{t}})\}, \ldots$.

In [10] it was also shown that there exists a set of pictures that can be generated by a grcpg, but that is not the sequence converging to the attractor of any IFS. Since grcpgs use context to control the sequence in which functions are applied, they can generate a wider range of pictures than IFSs. An example of such a picture set is $\mathcal{G}_{\mathrm{trail}}$, which is described below. $\mathcal{G}_{\mathrm{trail}}$ cannot be generated by a grFcpg, as becomes clear when inspecting the proof in [8], and therefore also not by an IFS.

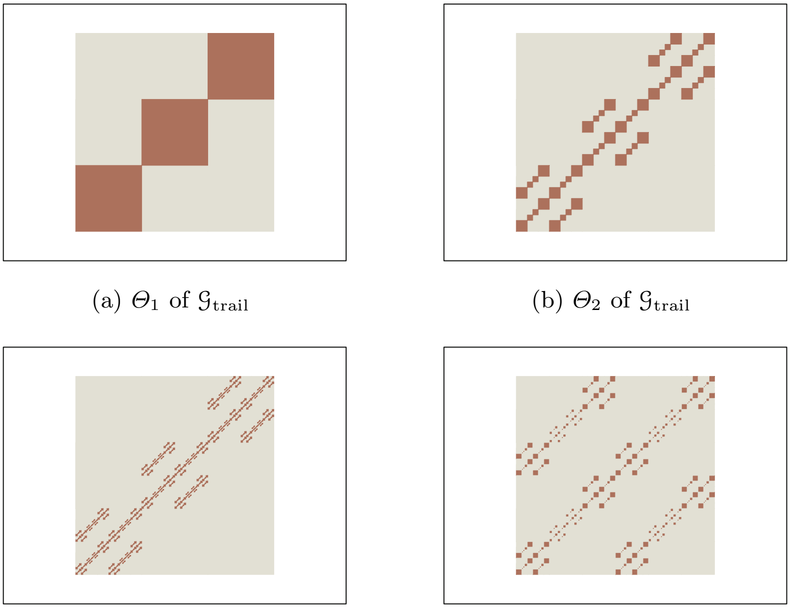
$\mathcal{G}_{\mathrm{trail}} = \{\Theta_1, \Theta_2, \ldots\}$, where Θ_1, Θ_2 and Θ_3 are shown in Fig. 2(a), Fig. 2(b) and Fig. 2(c), respectively. For the sake of clarity, an enlargement of the lower lefthand ninth of Θ_3 is given in Fig. 2(d).

For $i = 2, 3, \ldots$, Θ_i is obtained by dividing each dark square in Θ_{i-1} into four and placing a copy of Θ_1, modified so that it has exactly $i + 2$ dark squares, all on the diagonal, into each quarter.

The modification of Θ_1 is effected in its middle dark square only and proceeds in detail as follows: The square is divided into four and the newly-created lower lefthand quarter coloured dark. The newly-created upper righthand quarter is again divided into four and its lower lefthand quarter coloured dark. This successive quartering of the upper righthand square is repeated until a total of $i - 1$ dark squares have been created, then the upper righthand square is also coloured dark. The new dark squares thus get successively smaller, except for the last two, which are of equal size.

2.3 Mutually Recursive Function Systems

Mutually recursive function systems, called hierarchical iterated function systems by Peitgen and co-workers [13], are a generalization of IFSs, and consist of



(a) Θ_1 of $\mathcal{G}_{\text{trail}}$

(b) Θ_2 of $\mathcal{G}_{\text{trail}}$

(c) Θ_3 of $\mathcal{G}_{\text{trail}}$

(d) Bottom lefthand ninth of Θ_3 enlarged

Fig. 2. Pictures of $\mathcal{G}_{\text{trail}}$

networks or hierarchies of IFSs. Mutually recursive function systems were developed to study wider ranges of fractal-like images that do not exhibit such high degrees of self-similarity as IFSs [13]. In this section we review results that show that grcpgs are more powerful than MRFSs.

There are a number of slight variations in the definitions of MRFSs that can be found in the literature. Here we use the definition used by Drewes [4].

Definition 10. *Let $n \in \mathbb{N}_+$. Then $I = (M, c)$ is an MRFS of rank n such that*

- *M is an $n \times n$ matrix $(m_{i,j})$ with*
  - *$m_{i,j} = f^1_{i,j}, \ldots, f^{t_{i,j}}_{i,j}$, $t_{i,j} \in \mathbb{N}$, and*
  - *$\forall i, j \in [n]$ and $k \in [t_{i,j}]$, $f^k_{i,j} : \mathbb{R}^2 \to \mathbb{R}^2$.*
- *$c = (c_1, \ldots, c_n)$ is a vector where each c_i is a possibly empty compact subset of $\mathbb{R}^2$. These sets are called condensation sets.*
- *For each i such that c_i is empty, $\exists j$ such that $t_{i,j} > 0$.*

Mutually recursive function systems generate pictures through application of the extended Hutchinson operator.

Definition 11. *Given an MRFS $I = (M, c)$, the Hutchinson operator $H_I : \left(\wp\left(\mathbb{R}^2\right)\right)^n \to \left(\wp\left(\mathbb{R}^2\right)\right)^n$ is defined as follows: for $v = (v_1, \ldots, v_n) \in \left(\wp\left(\mathbb{R}^2\right)\right)^n$, $H_I(v) = (v'_1, \ldots, v'_n)$, where $v'_i = c_i \cup \bigcup_{j \in [n]} H_{m_{i,j}}(v_j)$ for $i \in [n]$.*

Now, given an MRFS $I = (M, c)$ and a vector of initial pictures $u = (u_1, \ldots, u_n)$, with u_i a compact, possibly empty subset of $\mathbb{R}^2$, the sequence of pictures generated by I is $S_I(u, 1), S_I(u, 2), \ldots$, where $S_I(u, i) = H_I^i(u)$ [1] (the first component of the ith iteration of the Hutchinson operator). The picture language obtained from I is the set $L(I, u) = \{S_I(u, i) | i \in \mathbb{N}_+\}$.

Example 3. The MRFS I_S of rank 3 generates pictures which consist of a Sierpiński triangle with a "shadow" consisting of an inverse Sierpiński triangle. Figure 3 shows the first four pictures in $L(I_S, a)$ where $a = (a_1, a_2, a_3)$, a_1 is the empty set, a_2 is the filled-in triangle with vertices $(0, 0), (5, \sqrt{75})$ and$(-5, \sqrt{75})$ and a_3 is the filled-in triangle with vertices $(-5, 0), (5, 0)$ and $(0, \sqrt{75})$.

$$
\begin{aligned}
&I_S = (M, c) \\
&\text{where :} \\
&M = \begin{pmatrix} \epsilon & f_7 & f_4, f_5, f_6 \\ \epsilon & \text{id}, f_1, f_2, f_3 & \epsilon \\ \epsilon & \epsilon & f_4, f_5, f_6 \end{pmatrix} \\
&\text{and for } i \in [3],\ c_i = \emptyset,
\end{aligned}
$$

with the functions defined as follows :

$$
\begin{aligned}
\text{id}(x, y) &= (x, y) \\
f_1(x, y) &= \left(\frac{x}{2} + 5, \frac{y}{2}\right) \\
f_2(x, y) &= \left(\frac{x}{2} - 5, \frac{y}{2}\right) \\
f_3(x, y) &= \left(\frac{x}{2}, \frac{y}{2} + \sqrt{75}\right) \\
f_4(x, y) &= \left(\frac{x}{2}, \frac{y}{2} + \frac{\sqrt{75}}{2}\right) \\
f_5(x, y) &= \left(\frac{x}{2} - 2.5, \frac{y}{2}\right) \\
f_6(x, y) &= \left(\frac{x}{2} + 2.5, \frac{y}{2}\right) \\
f_7(x, y) &= \left(\frac{x}{2} + y \tan \frac{\Pi}{8}, -\frac{y}{4}\right)
\end{aligned}
$$

Kruger and Ewert [12] showed that for every MRFS, an equivalent grcpg can be constructed. We state the result here in full—in Theorem 2—since the proof gives the translation from a given MRFS to a grFcpg.

Theorem 2. *An MRFS $I = (M, c)$, of degree n with a vector of initial pictures $a = (a_1, \ldots, a_n)$, can be translated into a grcpg G_I.*

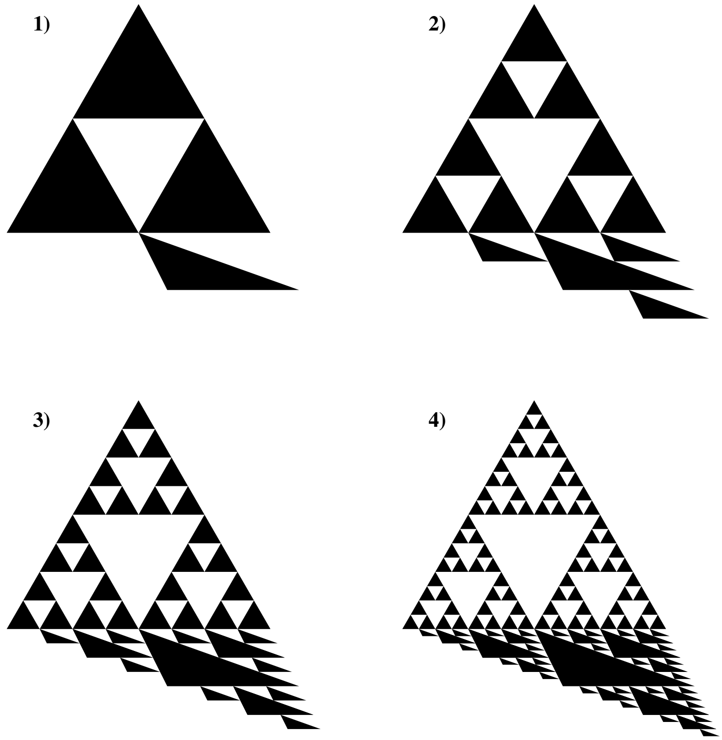


Fig. 3. Four pictures generated by the MRFS I_S

Proof.

$$\begin{aligned} G_I &= (V_N, V_T, V_F, P, (S, \epsilon)) \text{ where} \\ V_N &= \{S, I_1, \ldots, I_n, T_1, \ldots, T_n, U_1, \ldots, U_n, F_1, \ldots, F_n\} \\ V_T &= \{a_1, \ldots, a_n, c_1, \ldots, c_n\} \\ V_F &= \bigcup_{i,j \in [n]} \{f_{i,j}^1, \ldots, f_{i,j}^{t_{i,j}}\} \end{aligned}$$

and P is the set of productions :

$$\begin{aligned} &S \to \{(I_1, f_{1,1}^1), \ldots, (I_1, f_{1,1}^{t_{1,1}}), \ldots, (I_n, f_{1,n}^1), \ldots, (I_n, f_{1,n}^{t_{1,n}}), c_1\} \\ &I_i \to \{(T_1, f_{i,1}^1), \ldots, (T_1, f_{i,1}^{t_{i,1}}), \ldots, (T_n, f_{i,n}^1), \ldots, (T_n, f_{i,n}^{t_{i,n}}), c_i\} \\ &\quad\ (\emptyset; \{I_1, \ldots, I_{i-1}, F_1, \ldots, F_n, U_1, \ldots, U_n\}) \\ &I_i \to F_i(\emptyset; \{I_1, \ldots, I_{i-1}, T_1, \ldots, T_n, U_1, \ldots, U_n\}) \\ &T_i \to U_i(\emptyset; \{I_1, \ldots, I_n, T_1, \ldots, T_{i-1}\}) \\ &U_i \to I_i(\emptyset; \{T_1, \ldots, T_n, U_1, \ldots, U_{i-1}\}) \\ &F_i \to a_i(\emptyset; \{I_1, \ldots, I_n, F_1, \ldots, F_{i-1}\}) \end{aligned}$$

□

The language of G_I can be rendered in such a way that it is equal to the set of all approximations generated by I.

Example 4. The grcpg G_{shadow} was obtained by translating the MRFS I_S into a grcpg. With the terminals and functions defined as for I_S above, this grammar will generate exactly the same set of pictures as I_S.

$$
\begin{aligned}
G_{\text{shadow}} &= (V_N, V_T, V_F, P, (S, \epsilon)) \text{ where} \\
V_N &= \{S, I_1, I_2, I_3, T_1, T_2, T_3, U_1, U_2, U_3, F_1, F_2, F_3\} \\
V_T &= \{a_1, a_2, a_3\} \\
V_F &= \{\text{id}, f_1, f_2, f_3, f_4, f_5, f_6, f_7\}
\end{aligned}
$$

and P is the set of productions :

$$
\begin{aligned}
&S \to \{(I_2, f_7), (I_3, f_4), (I_3, f_5), (I_3, f_6)\} \\
&I_1 \to \{(T_2, f_7), (T_3, f_4), (T_3, f_5), (T_3, f_6)\}(\emptyset; \{F_1, F_2, F_3, U_1, U_2, U_3\}) \\
&I_1 \to F_1(\emptyset; \{T_1, T_2, T_3, U_1, U_2, U_3\}) \\
&I_2 \to \{(T_2, \text{id}), (T_2, f_1), (T_2, f_2), (T_2, f_3)\}(\emptyset; \{I_1, F_1, F_2, F_3, U_1, U_2, U_3\}) \\
&I_2 \to F_2(\emptyset; \{I_1, T_1, T_2, T_3, U_1, U_2, U_3\}) \\
&I_3 \to \{(T_3, f_4), (T_3, f_5), (T_3, f_6)\}(\emptyset; \{I_1, I_2, F_1, F_2, F_3, U_1, U_2, U_3\}) \\
&I_3 \to F_3(\emptyset; \{I_1, I_2, T_1, T_2, T_3, U_1, U_2, U_3\}) \\
&T_1 \to U_1(\emptyset; \{I_1, I_2, I_3\}) \\
&T_2 \to U_2(\emptyset; \{I_1, I_2, I_3, T_1\}) \\
&T_3 \to U_3(\emptyset; \{I_1, I_2, I_3, T_1, T_2\}) \\
&U_1 \to I_1(\emptyset; \{T_1, T_2, T_3\}) \\
&U_2 \to I_2(\emptyset; \{T_1, T_2, T_3, U_1\}) \\
&U_3 \to I_3(\emptyset; \{T_1, T_2, T_3, U_1, U_2\}) \\
&F_1 \to a_1(\emptyset; \{I_1, I_2, I_3\}) \\
&F_2 \to a_2(\emptyset; \{I_1, I_2, I_3, F_1\}) \\
&F_3 \to a_3(\emptyset; \{I_1, I_2, I_3, F_1, F_2\})
\end{aligned}
$$

In [12], Kruger and Ewert also showed that grcpgs can be constructed that generate sets of pictures that cannot be generated by any MRFS. Such a grcpg is easily obtained by simply modifying the context rules in a grcpg translated from some MRFS, to remove some (or all) of the restrictions that guarantee uniform refinement in the resulting pictures. Another easy way of obtaining such a grcpg is to simply add production rules to a grcpg translation of an MRFS.

Consider the set of all pictures that consist of a Sierpiński triangle with uniform refinement and a "shadow" made of an inverted Sierpiński triangle, also with uniform refinement, but the triangle and the "shadow" need not have the same level of refinement. Thus, this set contains all the pictures in $\mathcal{G}(G_{\text{shadow}})$ as well as pictures such as shown in Fig. 4. This set can be generated by a grFcpg, called G_{ext_1} in [12]. It should be clear that no MRFS can be constructed to generate all the pictures in this set.

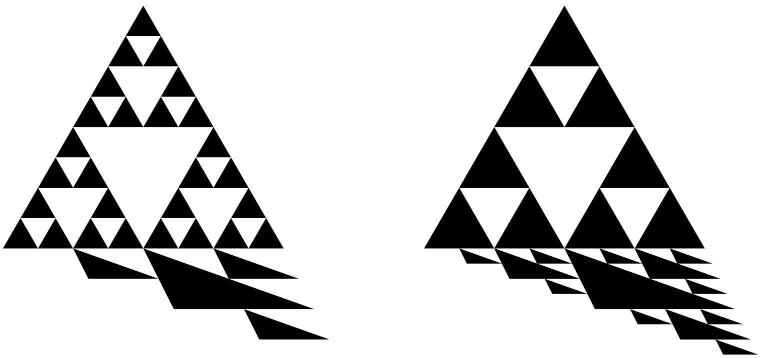


Fig. 4. Two pictures from $\mathcal{G}(G_{\mathrm{ext}_1})$ that are not in $\mathcal{G}(G_{\mathrm{shadow}})$

3 Language-Restricted Iterated Function Systems

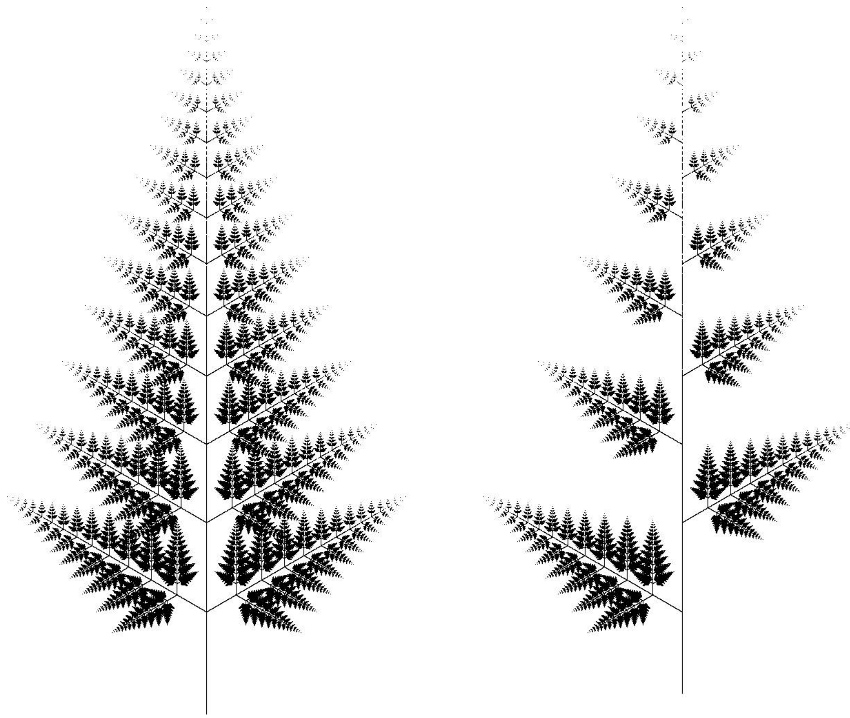
We can modify the picture produced by an IFS by using a language restriction, where a language controls which functions of the IFS are applied at different stages. This method of picture generation, introduced in [14], allows us to create pictures which are self-similar but not self-identical. For example, we can take an IFS which generates a picture of leaves on a stalk—Fig. 5(a)—and restrict it to get leaves on alternating sides—Fig. 5(b)—without changing the leaves themselves.

In this section we prove that the LRIFSs are strictly more powerful than the IFSs, and therefore investigate the relationship between LRIFSs and grcpgs. Although we do not investigate the relationship between LRIFSs and MRFSs, we use different types of approximation sequences for the two systems, so LRIFSs are of independent interest,

Definition 12. *A language-restricted iterated function system (LRIFS) is a tuple* $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ *where* $\mathcal{I} = \{X, \mathcal{F}\}$ *is an IFS, called the underlying IFS of* $\mathcal{I}_L$*, and* $L \subseteq \mathcal{F}^*$*.*

Following [15], we interpret the words of $\mathcal{F}^*$ as functions by *reverse* composition; that is, if $f = f_1 f_2 \ldots f_{n-1} f_n$, where $f_1, f_2, \ldots, f_{n-1}, f_n \in \mathcal{F}$, then $f(\pi) = f_n(f_{n-1}(\ldots f_2(f_1(\pi))\ldots))$. Unlike [15], however, we give the symbol $\circ$ its usual meaning.

The definition of the attractor of an LRIFS is based on the fact that if $\{X, \mathcal{F}\}$ is an IFS and π is a point in its attractor, then the attractor is equal to $\overline{\{f(\pi) \mid f \in \mathcal{F}^*\}}$, where $\overline{\cdot}$ denotes the topological closure.



(a) (b)

Fig. 5. An IFS and a language-restricted variation

Definition 13. *If $\mathcal{I}_L = (X, \mathcal{F}, L)$ is an LRIFS and $\pi \in X$ then the attractor of $\mathcal{I}_L$ at π is*

$$\mathcal{A}_\pi(\mathcal{I}_L) = \overline{\{f(\pi) \mid f \in L\}} \ .$$

Thus, every word of L contributes a single point to the picture. In [15], π is required to be in the attractor of the underlying IFS, but Lemma 1 shows that that is unnecessary.

In all examples in this work and in [15], L is a regular language and $\mathcal{F}$ a set of affine functions (an affine function is a translation composed with linear function). We call such LRIFSs regular and affine, respectively. Affine regular LRIFSs can generate a wide variety of pictures, even with a single underlying IFS. For example, see Fig. 5, already mentioned, and Fig. 6, which shows two fractals described in [15] and a version of the Sierpiński triangle (restricted with the language $(F_1 + F_3 + F_4)^*$ in the notation of that paper). The functions which generate Fig. 5 are

$$f_1 = t\left(0, \frac{1}{8}\right) \circ s\left(\frac{7}{8}\right)$$

$$f_2 = t\left(0, \frac{1}{8}\right) \circ r\,(60) \circ s\left(\frac{1}{3}\right)$$
$$f_3 = t\left(0, \frac{1}{8}\right) \circ r\,(-60) \circ s\left(\frac{1}{3}\right)$$
$$f_4 = \mathrm{proj}_y \circ s\left(\frac{1}{4}\right)$$

where t denotes translation, s scaling, r rotation (in degrees) and proj_y projection onto the y axis. Fig. 5(b) is restricted with the language $(f_1+f_2+f_3+f_4)^*(f_2+f_3f_1)(f_1f_1)^*(f_4f_1^*+\varepsilon)$.

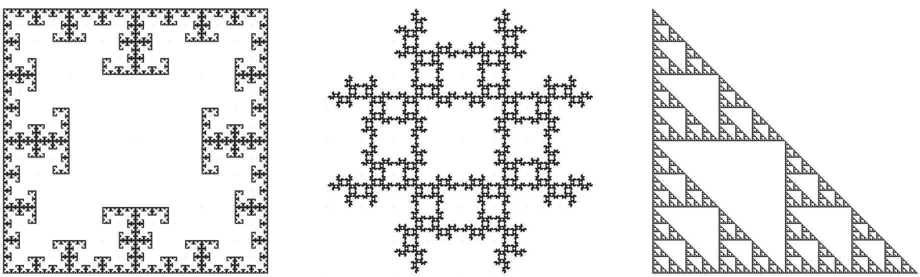


Fig. 6. Attractors of a single IFS restricted by three different languages

Every attractor of an IFS is also an attractor of a regular LRIFS, which can be seen by using the language $L = \mathcal{F}^*$. On the other hand, there are pictures which are the attractor of an LRIFS but not of any IFS (at least when we restrict ourselves to the affine functions). To prove this, we will need some basic facts about the closure properties of LRIFS attractors.

Lemma 1. *If $\mathcal{A}$ is the attractor of an (affine, regular) LRIFS at a point π, then there is another (affine, regular) LRIFS whose attractor is $\mathcal{A}$ at every point.*

Proof. Suppose $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ and $\mathcal{A} = \mathcal{A}_\pi(\mathcal{I}_L)$. Let g be the function which is constantly π, and $\mathcal{J}_{gL} = \{X, \mathcal{F} \cup \{g\}, gL\}$. Then

$$\begin{aligned}\mathcal{A} &= \mathcal{A}_\pi(\mathcal{I}_L)\\ &= \overline{\{f(\pi) \mid f \in L\}}\\ &= \overline{\{(f \circ g)(\rho) \mid f \in L\}}\\ &= \overline{\{f(\rho) \mid f \in gL\}}\\ &= \mathcal{A}_\rho(\mathcal{J}_{gL})\end{aligned}$$

for any ρ. Thus $\mathcal{J}_{gL}$ is the desired LRIFS; furthermore it is regular (resp. affine) if $\mathcal{I}_L$ is. □

Thus the starting point π is essentially arbitrary: if we want to generate a single picture, we can find an LRIFS which generates it from any starting point.

Lemma 2. *If $\mathcal{A}$ is the attractor of an (affine, regular) LRIFS at a point π and $0 < a < 1$, then there is another (affine, regular) LRIFS whose attractor at π is $s(a)(\mathcal{A})$.*

Proof. Suppose $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ and $\mathcal{A} = \mathcal{A}_\pi(\mathcal{I}_L)$. Let $g = s(a)$. Then $\mathcal{J}_{Lg} = \{X, \mathcal{F} \cup \{g\}, Lg\}$ has the desired attractor. □

Thus the class of LRIFS attractors is closed under downscaling. Furthermore, we will now show that they are closed under union.

Lemma 3. *If $\mathcal{A}$ and $\mathcal{A}'$ are attractors of (affine, regular) LRIFSs, then so is $\mathcal{A} \cup \mathcal{A}'$.*

Proof. Let $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ and $\mathcal{I}'_{L'} = \{X, \mathcal{F}', L'\}$ be LRIFSs whose attractors are $\mathcal{A}$ and $\mathcal{A}'$ respectively. By Lemma 1 we can assume, without loss of generality, that they can be generated from the same starting point, π. Then

$$\begin{aligned}
\mathcal{A} \cup \mathcal{A}' &= \mathcal{A}_\pi(\mathcal{I}_L) \cup \mathcal{A}_\pi(\mathcal{I}'_{L'}) \\
&= \overline{\{f(\pi) \mid f \in L\}} \cup \overline{\{f(\pi) \mid f \in L'\}} \\
&= \overline{\{f(\pi) \mid f \in L\} \cup \{f(\pi) \mid f \in L'\}} \\
&= \overline{\{f(\pi) \mid f \in (L \cup L')\}} \\
&= \mathcal{A}_\pi(\mathcal{J}_{L \cup L'})
\end{aligned}$$

where $\mathcal{J} = \{X, \mathcal{F} \cup \mathcal{F}'\}$. □

The previous three lemmas allow us to generate an LRIFS attractor by overlaying the (possibly downscaled) attractors of other LRIFSs (with all three operations preserving affineness and regularity). This contrasts with IFSs, as the following theorem shows by an example.

Theorem 3. *There is an LRIFS $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ with $\mathcal{F}$ a set of affine functions $X \to X$ whose attractor is not the attractor of any IFS $\mathcal{J} = \{X, \mathcal{F}'\}$ with $\mathcal{F}'$ a set of affine functions.*

Proof. Let $\mathcal{A}$ be the Cantor square, suitably scaled down, surrounded by a square, as depicted in Fig. 7. Since the Cantor square and the square are both attractors of IFSs, $\mathcal{A}$ is the attractor of an LRIFS by the above theorems.

Suppose $\mathcal{A}$ is the attractor of an IFS $\{X, \mathcal{F}\}$ where $\mathcal{F}$ is a set of affine functions.

Let $S \subseteq \mathcal{A}$ be the square, and $C = \mathcal{A} \setminus S$ be the Cantor square. Let $f \in \mathcal{F}$. Since S is connected, either $f(S) \subseteq S$ or $f(S) \subseteq C$.

If $f(S) \subseteq S$, then $f(S)$ must be a point or a line segment, since the image of S under an affine map is either a quadrilateral (but $f(S) \neq S$ since f is a contraction), a triangle (but no triangle is a subset of S), a line segment or a point.

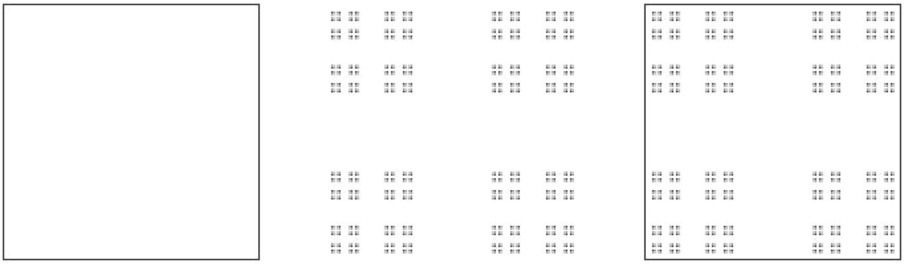


Fig. 7. A square, $\mathcal{S}$, and the Cantor square, $\mathcal{C}$, which are attractors of IFSs, and their union, $\mathcal{A}$

On the other hand, if $f(S) \subseteq C$ then $f(S)$ is a singleton, because C is totally disconnected.

Thus each $f \in \mathcal{F}$ maps S to either a line segment or a singleton. Since C is inside S and each f is affine, $f(\mathcal{A})$ is either a line segment or a singleton for each f. However, $\mathcal{A} = f_1(\mathcal{A}) \cup \ldots \cup f_n(\mathcal{A})$ but $\mathcal{A}$ is not a finite union of line segments and singletons, which is a contradiction. Thus $\mathcal{A}$ is not the attractor of any IFS. □

Since we have proven the LRIFSs are strictly more powerful than IFSs, we wish to extend our main result for IFSs—that they can be generated by a grFcpg—to LRIFSs. We use a different notion of approximation than for IFSs, because we wish to retain the information provided by all the strings in the language, rather than discarding them in better approximations.

Definition 14. *If L is a language, let $L_{\leq n} = \{x \in L : |x| \leq n\}$.*

We use the concept of $L_{\leq n}$ to generate approximations to the attractors of an LRIFS which are uniform, in that they are not closer approximations in one part than another. This is illustrated by Fig. 8, which shows three approximations based on $L_{\leq n}$ for different n, and one based on an arbitrary subset of L.

Theorem 4. *Let $\mathcal{I}_L = \{X, \mathcal{F}, L\}$ be a regular LRIFS, and π a point in X. Then there is a grFcpg G that can be rendered as*

$$\left\{\mathcal{A}_\pi(\mathcal{I}_{L_{\leq n}}) : n \in \mathbb{N}\right\} ,$$

and the functions used in rendering G are exactly those in $\mathcal{F}$ along with the identity.

Proof. Let $M = (Q, \mathcal{F}, q_0, A, \delta)$ be a deterministic finite automaton which recognizes L^r, the reverse language of L. The use of L^r is due to the fact that reverse composition is used in LRIFSs. We simulate all the paths of M by a grcpg G, and control G so that all paths are truncated at the same length.

Let $G = ((Q \times \{0,1\}) \cup \{S, C_0, C_1, C_2\}, \{e, p\}, \mathcal{F} \cup \{\mathrm{id}\}, P, (S, \mathrm{id}))$, where S, C_0, C_1, C_2, e and p are fresh symbols, id is the identity function on X, and P is constructed from N as follows:

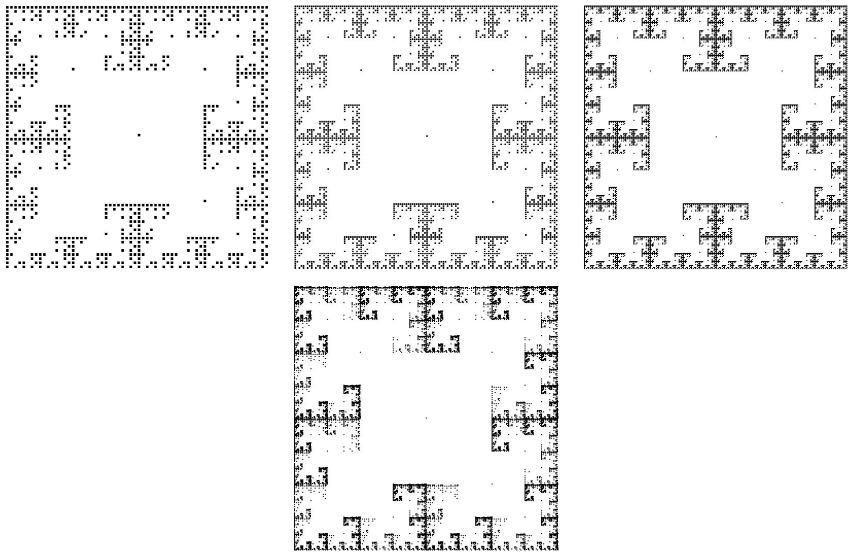


Fig. 8. Three uniform approximations of an LRIFS and a non-uniform approximation

1. $S \to (C_0, \mathrm{id})\ ((q_0, 0), \mathrm{id})$
2. For each accepting state $q \in A$, with edges to $q'_1, \ldots, q'_k \in Q$ labelled with $f_1, \ldots, f_k \in \mathcal{F}$ respectively, add a production

$$(q, 0) \to ((q'_1, 1), f_1)\ \ldots\ ((q'_k, 1), f_k)\ (p, \mathrm{id})\ \ (\emptyset; \{C_1\})$$

3. For each non-accepting state $q \in Q \setminus A$, with edges to $q'_1, \ldots, q'_k \in Q$ labelled with $f_1, \ldots, f_k \in \mathcal{F}$ respectively, add a production

$$(q, 0) \to ((q'_1, 1), f_1)\ \ldots\ ((q'_k, 1), f_k)\ \ (\emptyset; \{C_1\})$$

4. $C_0 \to (C_1, \mathrm{id})\ \ (\emptyset; Q \times \{0\})$
5. For each state $q \in Q$, add a production $(q, 1) \to ((q, 0), \mathrm{id})\ \ (\emptyset; \{C_0, C_2\})$
6. $C_1 \to (C_0, \mathrm{id})\ \ (\emptyset; Q \times \{1\})$
7. $C_0 \to (C_2, \mathrm{id})\ \ (\emptyset; Q \times \{0\})$
8. For each state $q \in Q$, add a production $(q, 1) \to (e, \mathrm{id})\ \ (\emptyset; \{C_0, C_1\})$
9. $C_2 \to (e, \mathrm{id})\ \ (\emptyset; Q \times \{1\})$

Any derivation in this grammar proceeds in phases. First the start symbol is rewritten by production 1. At any point where C_0 and $(q, 0)$ appear in the string, we can only apply productions from 2 and 3, since all other productions are forbidden or cannot be applied, and therefore these productions will be applied to all non-terminals of the form $(q, 0)$. When this is done, there is a single C_0 and all other non-terminals are of the form $(q, 1)$, and there is a choice of productions: 4 or 7.

If we apply production 4, we rewrite every $(q,1)$ into $(q,0)$ by production 5. When this is done we rewrite C_1 into C_0 by production 6 so that another iteration can be applied.

If, instead, we apply production 7, we proceed to delete the $(q,1)$ non-terminals by rewriting them to a symbol (production 8) which will be rendered as the empty set. Once they are all deleted, we delete C_2 by production 9.

Thus all branches are extended in tandem until they terminate. The branches, besides those containing the control symbols C_0, C_1 and C_2, correspond to paths through M up to a certain length, and are labelled by a composition of the symbols along the paths (interspersed with id). Thus if we render the generated gallery by interpreting each function in $\mathcal{F} \cup \{\text{id}\}$ as itself and rendering p by $\{\pi\}$ and e by $\emptyset$, then we obtain $\mathcal{A}_\pi(\mathcal{I}_{L_{\leq n}})$ for each $n \in \mathbb{N}$ (since $L_{\leq n}$ is finite, the closure operation in the definition of the attractor makes no difference). □

An example illustrating the method used in this proof is given in Fig. 9.

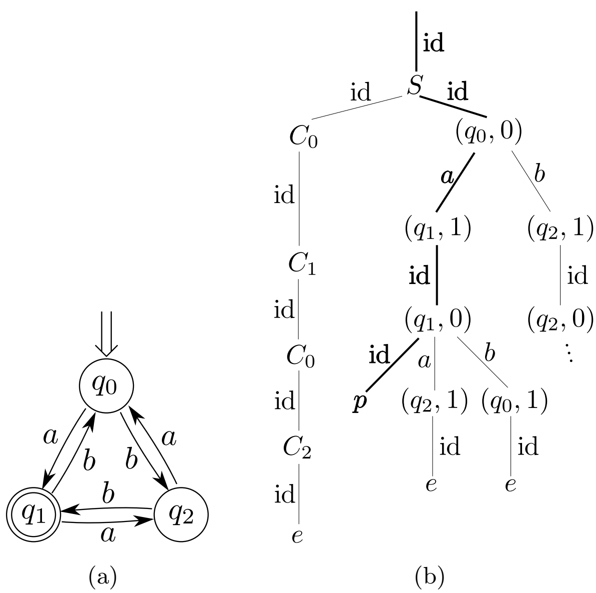


Fig. 9. (a) An automaton for a language L and (b) the derivation tree corresponding to $L_{\leq 1}$, with the highlighted path corresponding to the word $a \in L_{\leq 1}$

4 Future Work

Culik and Dube [3] showed that any uniformly growing, deterministic, context-free Lindenmayer system (D0L-system) can be simulated by an MRFS. As mentioned above, Kruger and Ewert [12] showed that for any MRFS, an equivalent grcpg can be constructed, and that the grcpg can be modified to generate sequences of pictures that cannot be generated by the basis MRFS. Therefore it would be interesting to simulate uniformly growing D0L-systems by grcpgs and then modify the grcpg to generate pictures that cannot be generated by the basis D0L-system, as we have done in this paper for IFSs and MRFS.

Our notion of a uniform approximation to an LRIFS is based exclusively on the length of the strings in the language; it would be interesting to formulate a notion which depends on the area, to obtain an approximation which looks uniform rather than having uniform depth, and determine whether this approximation can also be generated by a grcpg.

We showed that LRIFSs are more powerful than IFSs, but did not investigate their relationship to other extensions of IFSs (in particular MRFSs), and this topic is worthy of investigation.

Acknowledgements. We would like to thank the referees for their helpful comments.

References

1. Barnsley, M.F.: Fractals Everywhere. Academic Press, Boston (1988)
2. Barnsley, M.F., Hurd, L.P.: Fractal Image Compression. Peters, Wellesley (1993)
3. Culik II, K., Dube, S.: L-systems and mutually recursive function systems. Acta Informatica 30, 279–302 (1993)
4. Drewes, F.: Tree-based picture generation. Theoretical Computer Science 246, 1–51 (2000)
5. Ewert, S.: Random context picture grammars: The state of the art. In: Drewes, F., Habel, A., Hoffmann, B., Plump, D. (eds.) Manipulation of Graphs, Algebras and Pictures, pp. 135–147. Hohnholt, Bremen (2009)
6. Ewert, S., van der Walt, A.: Generating pictures using random forbidding context. International Journal of Pattern Recognition and Artificial Intelligence 12(7), 939–950 (1998)
7. Ewert, S., van der Walt, A.: Generating pictures using random permitting context. International Journal of Pattern Recognition and Artificial Intelligence 13(3), 339–355 (1999)
8. Ewert, S., van der Walt, A.: A hierarchy result for random forbidding context picture grammars. International Journal of Pattern Recognition and Artificial Intelligence 13(7), 997–1007 (1999)
9. Ewert, S., van der Walt, A.: Random context picture grammars. Publicationes Mathematicae (Debrecen) 54 (supp.), 763–786 (1999)
10. Ewert, S., van der Walt, A.: Shrink indecomposable fractals. Journal of Universal Computer Science 5(9), 521–531 (1999), `http://www.jucs.org/jucs_5_9`
11. Hoggar, S.G.: Mathematics for Computer Graphics. Cambridge University Press, Cambridge (1992)
12. Kruger, H., Ewert, S.: Translating mutually recursive function systems into generalised random context picture grammars. South African Computer Journal (36), 99–109 (2006)

13. Peitgen, H.O., Jürgens, H., Saupe, D.: Chaos and Fractals. New Frontiers of Science. Springer, New York (1992)
14. Prusinkiewicz, P., Hammel, M.: Automata, languages, and iterated function systems. In: Hart, J.C., Musgrave, F.K. (eds.) Fractal Modeling in 3D Computer Graphics and Imagery, pp. 115–143. ACM SIGGRAPH (1991)
15. Prusinkiewicz, P., Hammel, M.: Escape-time visualization method for language-restricted iterated function systems. In: Proceedings of Graphics Interface 1992, Vancouver, British Columbia, Canada, pp. 213–223 (May 1992)