

نگارال روشی برای تولید فرکتال‌ها

پدرام جعفری

چکیده. در این تحقیق به معرفی نظریه نگارال می‌پردازیم که در ابتدا برای مدل سازی رفتار گیاهان چند سلولی طراحی شد، اما پس از آن با طراحی مفسرهای گرافیکی، تبدیل به یک ابزار موفق برای شبیه سازی فرآیند رشد گیاهان و درختان شده است. در ابتدا با معرفی گونه‌های مختلف از گرامرهای آن، به بررسی ساختار و جایگاه آن در نظریه زبان‌های رسمی می‌پردازیم. سپس با معرفی مفسر گرافیکی لاک‌پشتی می‌توانیم رشته‌های خروجی از گرامرهای نگارال را به دستوراتی کم حجم برای تولید منحنی‌های فرکتالی و تصویرهایی تقریباً طبیعی از گیاهان و درختان تبدیل کنیم. به‌طور کلی مهم‌ترین ویژگی نظریه نگارال، اجرای قوانین به‌صورت موازی و هم‌زمان بر روی متغیرها است که باعث ایجاد کاربردهای وسیع در زیست‌شناسی تکوینی و حوزه‌های مختلف نظریه علوم رایانه شده است.

۱. مقدمه

آریستید لیندنمایر^۱ مجارستانی را به‌عنوان پدر نظریه Lindenmayer systems و یا به‌صورت مختصر L-systems می‌نامند که لغت نگارال، معادل فارسی برگزیده شده برای آن است. او که به‌عنوان استاد زیست‌شناسی و گیاه‌شناسی در دانشگاه اوترخت^۲ بر روی الگوی رشد گونه‌های مختلف گیاهان و جلبک‌ها، به‌خصوص خزه بالارو^۳ پژوهش می‌کرد، نگارال را در سال ۱۹۶۸ برای توصیف رشد گیاهان چند سلولی و رابطه همسایگی بین سلول‌های آن‌ها طراحی و معرفی کرد [۳، ۴]. در آن زمان تمرکز نظریه فقط بر روی توپولوژی گیاهان چند سلولی بود و جنبه‌های هندسی از برد آن فراتر بود. اما پس از مدتی چندین مفسر گرافیکی^۴ برای تبدیل نگارال به یک ابزار کارا برای مدل سازی گیاهان و درختان ارائه شد. در این مقاله از مفسر لاک‌پشتی^۵ استفاده خواهیم کرد که در [۸] براساس هندسه لاک‌پشتی^۶ طراحی و معرفی شده است.

نگارال امروزه بجز در شبیه سازی رشد سلول‌های گیاهی که هدف اصلی طراحی آن بوده، در گرافیک رایانه‌ای نیز برای تولید شکل‌های فرکتالی^۷ به‌مانند گیاهان کاربرد دارد [۹، ۱۱]. در حقیقت اختراع لیندنمایر به یکی از نظریه‌های زیبایی بین رشته‌ای تبدیل شده است: به‌طوری‌که پیشرفت در این نظریه، ایده‌ها و نتایج جدیدی را در دیگر حوزه‌های علوم به‌وجود آورده است. به‌عنوان مثال می‌توان به کتاب [۱۲] اشاره کرد که به تاثیرات نظریه نگارال بر نظریه علوم رایانه، گرافیک رایانه‌ای و زیست‌شناسی تکوینی^۸ پرداخته است. تعداد مقالاتی که فقط به جنبه تئوری این نظریه پرداخته‌اند بالغ بر یک هزار می‌باشند که در [۱۰] کسری از مقالات مهم در این حوزه معرفی شده است. همچنین این منبع به معرفی تعدادی از مقالات در زمینه

عبارات و کلمات کلیدی. نگارال، فرکتال، مدل سازی گیاهان، نظریه زبان‌های رسمی، مفسر گرافیکی لاک‌پشتی.

دبیر تخصصی رابط: سعیداعظم

تاریخ دریافت: ۱۳۹۴/۱۱/۰۶ تاریخ پذیرش: ۱۳۹۷/۰۱/۱۵

<http://dx.doi.org/10.22108/msci.2018.47307.1160>

¹ Aristid Lindenmayer ² Utrecht University ³ Anabaena ⁴ Graphical Interpreter ⁵ Turtle Interpreter ⁶ Turtle Geometry ⁷ Fractal

⁸ Developmental Biology

کاربرد نظریه نگارال در زیست شناسی نیز پرداخته است. ازجمله کاربردهای جالب و غیر منتظره آن می‌توان به استفاده از آن برای تولید ساختارهای موسیقی اشاره نمود [۷].

واژه فرکتال در سال ۱۹۷۵ توسط ریاضی‌دان فرانسوی به نام بنوا ماندلبرو^۹ وارد دنیای ریاضیات شد. این لغت که از کلمه‌ای لاتین به معنای «شکسته شده» برگرفته شده، برای اشیایی به کار می‌رود که در توصیف هندسه اقلیدسی زیادی نامنظم هستند. برای مثال کوه‌ها، درختان، ابرها و یا خطوط ساحلی را نمی‌توان توسط هندسه اقلیدسی توصیف کرد. فرکتال‌ها به‌طور کلی تعریف دقیق ریاضی ندارند اما بیشتر آن‌ها خصوصیات مشابهی دارند، از جمله این‌که دارای الگوهای تکراری هستند، یعنی با هر مقیاسی که شی مورد نظر را بزرگ نمایی کنیم، یک سری الگوهای تکراری را در خودشان نشان می‌دهند. اگر این الگوهای تکراری در هر مقیاسی یکسان باشند، آن شی را خود متشابه^{۱۰} گویند (برای مطالعه بیشتر منابع [۱، ۲] و یا منبع فنی‌تر [۶] پیشنهاد می‌شود).

اما دلیل انجام این پژوهش برای نگارنده، وجود دو تابلوی بزرگ موزاییک شده بر روی ضلع غربی دانشکده ریاضی دانشگاه اصفهان است که هیچ‌گونه توضیح و یا ارجاع به منبعی برای‌شان وجود ندارد (شکل ۱). اما مفهوم این دو تابلو به‌صورت تصادفی در حین گشت زنی در یک کتاب‌خانه با مشاهده کتاب [۱۱] برای نگارنده مشخص شد!



شکل ۱: یکی از دو تابلوی موزاییک شده‌ی دانشکده ریاضی

۲. ساختار تشکیل دهنده‌ی نگارال

گرامرهای مختلفی که در دروس نظریه محاسبه^{۱۱} و یا نظریه زبان‌ها و ماشین‌ها^{۱۲} می‌آموزیم ویژگی‌های مشترکی دارند از جمله این‌که: تمام‌شان بر اساس یک الفبای خروجی Σ (ترمینال‌ها^{۱۳})، مجموعه‌ای از متغیرها، یک متغیر آغازین^{۱۴} و مجموعه‌ای از قوانین جای‌گزینی^{۱۵} هستند. که هر قانون یک متغیر را به یک رشته از متغیرها و ترمینال‌ها تبدیل می‌کند. تفاوت ترمینال‌ها با متغیرها در این است که یک رشته زمانی به‌عنوان یک رشته‌ی خروجی پذیرفته می‌شود که تمامی متغیرهای آن به ترمینال‌ها تبدیل شده باشند. پس زبان یک گرامر^{۱۶} را مجموعه رشته‌های خروجی، از اجرای قوانین گرامر بر روی متغیر

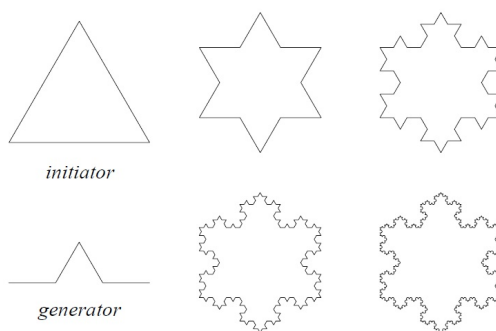
^۹Benoît Mandelbrot ^{۱۰}Self-similar ^{۱۱}Theory of Computation ^{۱۲}Formal Languages and Automata ^{۱۳}Terminals ^{۱۴}Start variable

^{۱۵}Substitution rules ^{۱۶}Language of a grammar

آغازین آن گرامر می‌دانند. این مجموعه تعاریف را به‌عنوان مفهومی از سیستم‌های بازنویسی^{۱۷} در نظر می‌گیرند (در صورت نیاز به اطلاعات بیشتر مرجع [۵] پیشنهاد می‌شود).

اما نگارال را باید توسط سیستم‌های بازنویسی موازی^{۱۸} تعریف کرد. زمانی یک مدل برای توصیف رشد ساختارهای چندسلولی مناسب است که در آن عمل رشد به‌صورت هم‌زمان^{۱۹} در هر واحد زمانی، برای تمام سلول‌های آن رخ بدهد. زیرا هر یک از این زیر واحدهای سلولی ساختار ژنتیکی یکسانی دارند و زمان نیز به‌صورت یکسان برای تمامی سلول‌ها گذر می‌کند، پس در شرایط استاندارد نمی‌توانیم قواعد رشد را به‌صورت غیر هم‌زمان^{۲۰} بر روی یکی از سلول‌ها و سپس دیگری اعمال کنیم.

برای درک بهتر می‌توانیم به شی گرافیکی برف دانه کخ^{۲۱} اشاره کنیم که به‌وسیله قوانین بازنویسی موازی تولید می‌شود. مندلیو ساخت آن را در کتاب [۶] به‌گونه زیر توصیف کرده است: برای ساخت آن از دو شکل آغازگر^{۲۲} و مولد^{۲۳} استفاده می‌کنیم (شکل ۲). مولد یک خط شکسته جهت‌دار است که از ۴ قسمت هم‌اندازه به طول متغیر t تشکیل شده است. با شروع بر روی شکل آغازگر، در هر مرحله به ازای هر خط راست موجود، یک مولد هم‌جهت را به‌گونه‌ای جای‌گزین آن می‌کنیم که نقاط انتهایی‌شان بر روی هم قرار گیرند. هر مرحله زمانی به اتمام می‌رسد که تمام خطوط راست در مرحله قبلی، با یک مولد هم‌اندازه و هم‌جهت خود جای‌گزین شده باشند.



شکل ۲: برف دانه کخ.

برای معرفی‌ای کوتاه بر مهم‌ترین گونه‌های گرامرهای نگارال، از ساده‌ترین آن‌ها شروع می‌کنیم: گرامر نگارالی مستقل از متن^{۲۴} یا $\circ - \text{نگارال}$ ($\circ L$) است که برای اجرای هر قانون آن تنها به یک نماد در سمت چپ قانون نیاز داشته باشیم. به عبارت دیگر قوانین‌شان از فرم زیر است:

$$(۱) \quad a \rightarrow u$$

که $a \in \Sigma$ و $u \in \Sigma^*$. زمانی که می‌خواهیم یک رشته بازنویسی شود، یک چنین قانونی را باید برای تمام نمادهای رشته اجرا کنیم تا رشته جدید حاصل شود. همچنین علامت صفر در آن اشاره به تعاملات یک طرفه‌ی آن دارد^{۲۵}. درواقع این‌گونه قوانین مناسب سلول‌هایی هستند که فقط از والد خود ارث می‌برند، یعنی اطلاعات از سلول والد به فرزندان‌اش انتقال می‌یابد. درمقابل اگر در مدل‌مان نیاز به تبادل اطلاعات بین سلول‌های همسایه نیز داشته باشیم (مثلا جریان مواد معدنی و یا هورمون‌ها)، در آن صورت گرامر نگارال حساس به متن^{۲۶} مناسب کار ما خواهد بود. این گرامر را در حالت کلی برای k

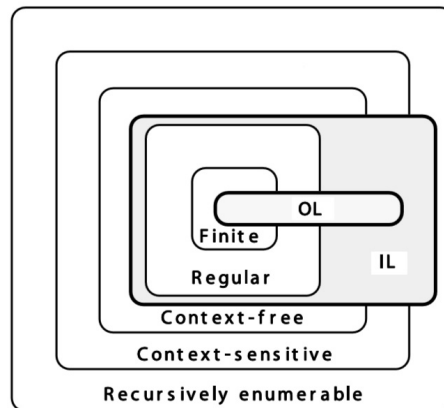
¹⁷ Rewriting systems ¹⁸ Parallel ¹⁹ Synchronous ²⁰ Asynchronous ²¹ Koch snowflake ²² Initiator ²³ Generator ²⁴ Context-free L-systems ²⁵ L-systems with zero-sided interactions ²⁶ Context-sensitive L-systems

و l متغیر، (k, l) -نگارال و یا l -نگارال (lL) می‌نامند و قوانین آن به فرم زیر می‌باشند [۱۰، ۱۱]:

$$(2) \quad a_k < a > a_l \rightarrow u$$

که $a_l \in \Sigma^l$ و $a_k \in \Sigma^k$ ، $u \in \Sigma^*$ ، $a \in \Sigma$ (یعنی در سمت چپ a رشته‌ای به طول k و در سمت راست آن رشته‌ای به طول l باید باشد تا a به رشته‌ی u تبدیل شود). حال اگر برای هر نماد موجود دقیقاً یک قانون تعریف شده باشد، آن نگارال را معین^{۲۷} می‌نامند و درمقابل اگر برای یک نمادی چندین قانون متفاوت تعریف شده باشد و هرکدام را با یک احتمال مشخص در هر تکرار اجرا کنند، نگارال مورد نظر را تصادفی^{۲۸} می‌نامند. درواقع این نام‌گذاری‌ها بر اساس تعریف‌های مشابه در نظریه زبان‌های رسمی انتخاب شده‌اند.

می‌توان گفت که تفاوت اصلی گرامرهای نگارال با دیگر گرامرها به‌مانند گرامرهای مستقل از متن^{۲۹}، در روش به‌کارگیری قوانین است. در گرامرهای مستقل از متن قوانین به‌صورت دنباله‌ای و پشت سرهم بر روی متغیرها اجرا می‌شوند، در حالی که گرامرهای نگارال قوانین را به‌صورت موازی و هم‌زمان بر روی تمام متغیرهای یک رشته اجرا می‌کنند. این ویژگی تاثیر اساسی بر روی خواص گرامرهای نگارال گذاشته است. برای مثال، وجود دارد زبان‌هایی که توسط گرامر نگارال مستقل از متن ساخته می‌شوند ولی نمی‌توان آن‌ها را توسط گرامرهای مستقل از متن ساخت (شکل ۳) [۱۰].



شکل ۳: سلسله مراتب زبان‌های رسمی.

۳. مثال‌هایی از نگارال مستقل از متن

مثال ۱.۳. خزّه بالارو

variables : A, B

constants : none

axiom : A

rules : $(A \rightarrow AB), (B \rightarrow A)$

متغیری که اصل (axiom) در نظر گرفته می‌شود در حقیقت همان متغیر آغازین است و منظور از ثابت‌ها (constants) همان ترمینال‌ها است. که در اجراهای اول تا هفتم خواهیم داشت:

$n = \circ : A$

$n = \backslash : AB$

²⁷ Deterministic ²⁸ Stochastic ²⁹ Context-free grammars

$$n = \mathfrak{Y} : ABA$$

$$n = 3 : AB AAB$$

$$n = 4 : ABABABA$$

$$n = \mathfrak{5} : ABABABABABAB$$

$$n = \mathfrak{S} : ABAABABAABAABAABAABA$$

$$n = \Upsilon : ABAABABAABAABABAABABAABABAABABAAB$$

اگر طول هر رشته را در نظر بگیریم، دنباله اعداد فیوناچی را خواهیم داشت (بدون در نظر گرفتن اولین یک، به خاطر انتخاب اصل مان).



شکل ۴: خزہ بالارو.

در این مثال اگر به جای قانون $A \rightarrow AB$ از قانون $A \rightarrow BA$ استفاده کنیم، از نظر اندازه نتیجه یکسان خواهد بود ولی مکان Aها و Bها تغییر خواهد کرد، در حقیقت رشته‌ها معکوس خواهند شد (رشته‌ها از لحاظ هندسی یکسان می‌مانند ولی از لحاظ توپولوژی تغییر خواهند کرد).

variables : A, B

constants : none

$$axiom : A$$
$$rules : (A \rightarrow ABA), (B \rightarrow BBB)$$

مثال ۲.۳. مجموعه کانتور^{۳۰}

اگر A را به معنای رسم رو به جلو و B را به معنای حرکت به جلو بدون رسم در نظر بگیریم، حاصل کار مجموعه معروف فرکتالی کانتور بر روی اعداد حقیقی خواهد بود.

$$n = \circ : A$$

$$n = \backslash : ABA$$

$$n = \mathfrak{Y} : ABABBBABA$$

$$n = \mathfrak{V} : ABABBBABABBBBBBBBBBABABBBABA$$

³⁰ Cantor set



شکل ۵: مجموعه کانتور

۱.۳. مفسر لاک پشتی. برای مدل سازی گیاهان پیچیده تر، به یک مفسر گرافیکی پیشرفته تر برای نگارال نیازمندیم. یکی از این مفسرها بر اساس ایده هندسه لاک پشتی ارائه شده است: یک لاک پشت بر روی یک کاغذ قرار دارد و ما به آن دستورات مشخصی می دهیم. دم لاک پشت کثیف است و در صورت تماس با کاغذ از خودش رد باقی می گذارد. دستورات ما براساس نمادهای زیر است: F به معنای حرکت به اندازه یک واحد به سمت مستقیم با جای گذاری مسیر حرکت و f به معنای حرکت به اندازه یک واحد به سمت مستقیم بدون جای گذاری اثر است. $+$ به معنای چرخش به سمت چپ به اندازه زاویه α و $-$ به معنای چرخش به سمت راست به اندازه زاویه α است. [به معنای گذاشتن 31 حالت فعلی 32 (ویژگی هایی از جمله موقعیت، زاویه، رنگ و پهنای خطوط) به درون پشته 33 و [به معنای خارج کردن 34 بالاترین حالت از پشته و تبدیل آن به حالت فعلی لاک پشت است. در اجرای [و [هیچ خطی رسم نمی شود اگرچه ممکن است موقعیت لاک پشت تغییر کند.

variables : F

constants : $+$ $-$

axiom : F

rules : $(F \rightarrow F + F - F - F + F)$

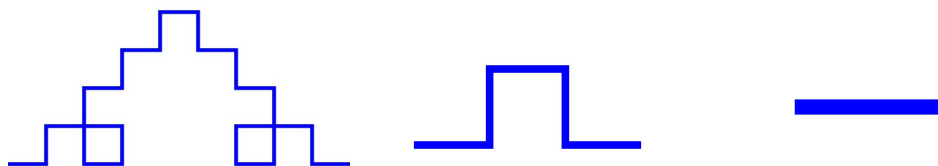
angle : 90°

مثال ۳.۳. منحنی کخ 35

$n = \circ : F$

$n = \backslash : F + F - F - F + F$

$n = \vee : F + F - F - F + F + F + F - F - F + F - F + F - F - F + F - F + F + F - F - F + F$

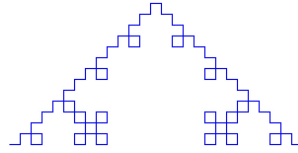


$n = 2,$

$n = 1,$

شکل ۶: $n = 0$

³¹ Push ³² Current State ³³ Stack ³⁴ Pop ³⁵ Koch curve



شکل ۷: منحنی کخ برای $n = 3$.

مثال ۴.۳. منحنی هیلبرت^{۳۶}

منحنی هیلبرت از جمله منحنی‌های فضا-پرکنی است که توسط یک منحنی یک بعدی کل فضای دو بعدی را دربرمی‌گیرند. دقت شود که وقتی از مفسر لاک‌پشتی استفاده می‌کنیم، هنگام رسم فقط دستورات F ، $+$ و $-$ را پیاده سازی می‌کنیم و متغیرهای A و B روی رسم ما تاثیر مستقیمی نخواهند گذاشت.

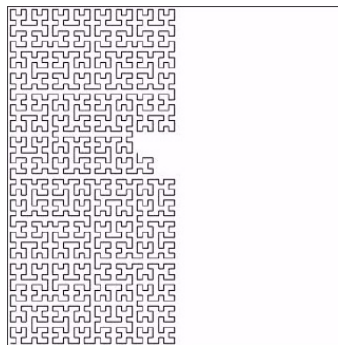
variables : A, B

constants : $F + -$

axiom : A

rules : $(A \rightarrow +BF - AFA - FB+), (B \rightarrow -AF + BFB + FA-)$

angle : 90°



شکل ۸: منحنی هیلبرت برای $n = 6$.

مثال ۵.۳. مثلث سیرپینسکی^{۳۷}

variables : F_1, F_2

constants : $+$ $-$

axiom : F_1

rules : $(F_1 \rightarrow F_2 - F_1 - F_2), (F_2 \rightarrow F_1 + F_2 + F_1)$

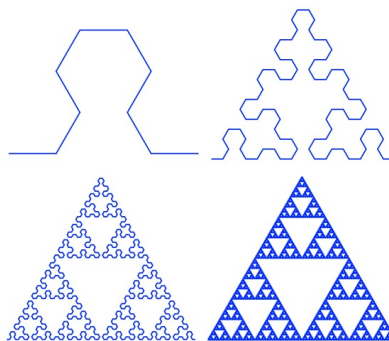
angle : 60°

مثال ۶.۳. گیاه فرکتالی

variables : X, F

constants : $+$ $-$ $[]$

³⁶ Hilbert curve ³⁷ Sierpinski triangle



شکل ۹: مثلث سیرپینسکی برای $n = 2, n = 4, n = 6, n = 8$.

axiom : X

rules : $(X \rightarrow F - [[X] + X] + F[+FX] - X), (F \rightarrow FF)$

angle : 25°

در این مثال - به معنای «چرخش به سمت چپ به اندازه 25° درجه» و + به معنای «چرخش به سمت راست به اندازه 25° درجه» است. برای X هیچ رسمی را انجام نمی‌دهیم و فقط برای کنترل دستورات است. [برای ذخیره کردن موقعیت و زاویه فعلی در پشته است که با دستور] بازایی می‌شود (شکل ۱۰ را ببینید).



شکل ۱۰: گیاه فرکتالی برای $n = 6$.

۴. نگارال تصادفی

برای تولید مجموعه‌ای از گیاهان، اگر آن‌ها را با یک نگارال معین تولید کنیم حاصل کار منظره‌ای منظم و مصنوعی خواهد بود. پس باید سیستمی طراحی شود که در آن هر گونه با همون‌اش متفاوت باشد ولی در خصوصیات عام گونه‌ی

خودشان مشترک باشند. تنوع مورد نظرمان می‌تواند از طریق تصادفی کردن تفسیر لاک‌پشتی بدست بیاید و یا این‌که خود گرامرهای نگارال را تصادفی کنیم. تصادفی کردن تفسیر به‌تنهایی تاثیر محدودی دارد. با این‌که ویژگی‌های هندسی گیاه از جمله طول و زاویه شاخه‌ها تغییر خواهد کرد، توپولوژی گیاه بدون تغییر باقی می‌ماند. درمقابل به‌کار بردن قوانین تصادفی (احتمالی) می‌تواند هم هندسه و هم توپولوژی گیاه را تحت تاثیر قرار دهد. برای مثال می‌توانیم قانون $0 \rightarrow 1[0]$ را به قانون احتمالی $0 \rightarrow 1[0] (P = 0/5)$ تغییر بدهیم. دقت شود که اگر برای احتمال باقی‌مانده متغیری قانونی تعریف نشده باشد، به‌صورت پیش فرض قانون یک‌ای که به خودش برمی‌گردد برای آن در نظر گرفته می‌شود. یعنی در مثال ذکر شده به‌صورت پیش فرض برای $0 \rightarrow 0 (P = 0/5)$ خواهیم داشت: (برای توضیحات بیشتر مرجع [۱۰] پیشنهاد می‌شود).

۵. نتیجه‌گیری

هدف کلی این مقاله، آشنا سازی با این تکنیک مدل سازی و شبیه سازی رشد گیاهان بود که امروزه در گرافیک رایانه‌ای، زیست‌شناسی و در نظریه زبان‌های رسمی بیشترین کاربرد را دارد، اما به اعتقاد نگارنده پتانسیل ایجاد کاربردهای جدیدتر و جالب‌تری را نیز دارا است. به دلیل سنگین شدن بحث، وارد موضوعاتی هم‌چون نگارال حساس به متن، نگارال پارامتری و نیز مدل سازی سه بعدی گیاهان نمی‌شویم. در خاتمه از دوستان عزیز تقاضا دارم به بررسی دو تابلوی دانشکده نیز بپردازند، که در شکل ۱ یکی از آن‌ها که یک نگارال تصادفی است آورده شده است (البته توقعی به رسم دستی آن‌ها برای تکرارهای بالا نیست، چون بیشترشان با برنامه‌های رایانه‌ای پیاده سازی می‌شوند). همچنین به علاقه‌مندان به کسب اطلاعات بیشتر در مورد فرکتال‌ها، جستجوی ساده‌ای را در اینترنت پیشنهاد می‌کنم که شکل‌های مسحور کننده‌ای را به‌همراه خواهد داشت. بر خود لازم می‌دانم از جناب آقای دکتر سلیمی مقدم، جناب آقای دکتر پوریای ولی، سرکار خانم دکتر رضایی و جناب آقای دکتر میرمیران به‌خاطر تشویق و حمایت‌های‌شان برای به ثمر رساندن این تحقیق تشکر و قدردانی کنم.

مراجع

- [۱] م. دلخوش، معرفی فرکتال‌ها و بعدها‌های کسری، نشریه ریاضی و جامعه، ۲ (۱۳۹۴) ۱-۲۳.
- [2] D. P. Feldman, *Chaos and Fractals, An Elementary Introduction*, Oxford University Press, Oxford, 2012.
- [3] A. Lindenmayer, Mathematical models for cellular interaction in development, Parts I and II, *Journal of Theoretical Biology*, **18** (1968) 280-315.
- [4] A. Lindenmayer, Developmental systems without cellular interaction, their languages and grammars, *Journal of Theoretical Biology*, **2** (1971) 455-484.
- [5] P. Linz, *An Introduction to Formal Languages and Automata*, Fourth ed, Jones and Bratlett Publishers, Boston, 2008.
- [6] B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman, San Francisco, 1982.
- [7] S. Manousakis, *Musical L-Systems*, MS. thesis, The Royal Conservatory of The Hague, June 2006.
- [8] P. Prusinkiewicz, Graphical applications of L-systems, In *Proceedings of Graphics Interface '86 — Vision Interface*, **86** (1986) 247-253.
- [9] P. Prusinkiewicz, Applications of L-systems to computer imagery, In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Third International Workshop*, 534-548, Springer-Verlag, Berlin, 1987, Lecture Notes in Computer Science 291.
- [10] P. Prusinkiewicz and J. Hanan, *Lindenmayer Systems, Fractals, and Plants*, Springer-Verlag, Berlin, 1989.
- [11] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990.
- [12] G. Rozenberg and A. Salomaa, *Lindenmayer Systems Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Springer-Verlag, Berlin, 1992.

[13] Examples from: <http://en.wikipedia.org/wiki/L-system>, 2016.

پدرام جعفری قمی

تهران، دانشگاه امیرکبیر، گروه علوم کامپیوتر

p.jafari.q@gmail.com

پدرام جعفری قمی متولد سال ۱۳۷۱ در شهر تهران است. وی در سال ۱۳۹۰ وارد مقطع کارشناسی رشته علوم کامپیوتر دانشگاه اصفهان شد و در سال ۱۳۹۴ وارد مقطع کارشناسی ارشد رشته علوم کامپیوتر گرایش نظریه محاسبه در دانشگاه امیرکبیر شده است.

