

НЕСКОЛЬКО НОВЫХ БИОПОДОБНЫХ L-СИСТЕМ

Калмыков А.В., Калмыков Л.В., Кешелава А.В.

(Пушино, Московской области)

С использованием принципов генетических алгоритмов и логического проектирования создано 12 новых биоподобных L-систем (систем Линденмаера). В работе приведены как графические объекты, так и логические формулы этих L-систем (аксиомы, подстановки, углы поворотов и число итераций перезаписи). Обсуждается важность данного типа моделирования для решения проблем биологического морфогенеза и использования биологических идей в информатике. Авторы классифицируют данное направление исследований как «искусственную жизнь» (artificial life).

ABOUT NEW BIOSIMILAR L-SYSTEMS (LINDENMAER'S SYSTEMS).

Kalmykov A.V., Kalmykov L.V., Keshelava A.V.

(Pushchino, Moscow Region)

With use of principles of genetic algorithms and logical designing it is created 12 new biosimilar L-systems (Lindenmaer's systems). In the paper are presented both graphic objects, and logic formulas of these L-systems (an axiom, substitution, corners of turns and number of iterations of rewriting). Importance of the given type of modelling for the decision of problems biological morphogenesis and uses of biological ideas in computer science are discussed. Authors classify the given direction of researches as "artificial life"

Введение.

L-системы иначе называются переписываемыми алгоритмами генерации математических структур. Данный тип алгоритмов был предложен в 1905 году математиком von Koch (генерация кривой, огибающей снежинку). В конце пятидесятих годов

Хомский использовал эти алгоритмы для описания формальных грамматик, а в 1968 году биолог Аристид Линденмаер (по имени которого и были названы L-системы) использовал генерацию структур на основе последовательно (итерационно) перезаписываемых правил для моделирования морфогенеза растений. В 1975 году подобные структуры были названы Бенуа Мандельбротом «фракталами». L-системы, получающиеся на каждом из последовательных этапов генерации масштабируются к исходно заданным размерным ограничениям («чувствительны к окружающей среде»). Главные свойства графических объектов L-структур - симметрия и *фрактальность* (самоподобие). Красота (сложная симметрия) и фрактальность лежат, по-видимому, в основе мира живого. Логические формулы генерации L-систем могут рассматриваться как аналог *генома*. Новые структуры были получены нами в основном в результате перекомбинирования данных логических формул генерации геномов. Использовались принципы генетических алгоритмов (кроссинговера). Применялось также прямое логическое проектирование новых L-систем и комбинированные подходы.

Актуальность создания новых биоподобных L-систем связана не только с поисками решения *проблем биологического морфогенеза*, но и с возможностями *использования биологических идей в информатике*. L-системы могут быть отнесены к «*искусственной жизни*» (www.alife.org). Эта область воспроизводит с помощью компьютеров характерные свойства живого. Создаваемые сегодня системы искусственного интеллекта осваивают последнюю пока еще «чисто человеческую» функцию – сравнительную эстетическую оценку случайно генерируемых решений. Создание все новых и новых, все более красивых и все более биологически подобных L-систем является вкладом в освоение компьютерами (пока еще с помощью человека) мира эстетики, лежащего в основе принятия человеком наиболее сложных решений.

Целью работы было создание новых биоподобных L-систем с использованием принципов генетических алгоритмов.

Оборудование и метод

Использовались компьютеры уровня Pentium I (166 MHz) и бесплатная свободно распространяемая (с www.rechka.ru) Java-программа моделирования L-систем - «MKokh». В основе рабо-

ты программы лежат итерационные подстановочные модификации исходной формулы так называемой черепашьей графики. Черепашья графика - это такой способ рисования линий на экране компьютера. Он состоит в том, что программист как бы управляет движением как бы черепашки. Черепашка, ползая по экрану, оставляет за собой след. При этом цель программиста – управлять черепашкой так, чтобы черепашка нарисовала нужную линию. Команды управления черепашкой просты: сделать шаг вперёд (обозначается F), повернуть направо (обозначается +), повернуть налево (обозначается -), сделать шаг вперёд без перерисовки (прыжок, обозначается V). Вот из этих команд и составляется сценарий построения линии – строка команд. Величина одного шага и угол одного поворота при движении черепашки всегда остаются постоянными и задаются предварительно.

Например, F++F++F это равносторонний треугольник, если угол поворота равен $\pi/3$ (т.е. $180^\circ/3$).

А вот данные для построения «куста» (формула взята с www.rechka.ru).

axiom = F

newF = -F+F+[+F-F]-[-F+F+F]

Угол поворота = $\pi / 8$

При последовательном увеличении числа шагов, выполняемых программой с использованием этого стиля генерации L-систем, получаются развивающиеся структуры (результат каждый раз автоматически масштабируется к заданному формату рамки):



Итерационные модификации исходной формулы черепашьей графики реализуются как последовательные замены элементарных элементов исходной формулы-аксиомы на определенные заданные структуры (так называемые «новые элементарные

элементы»), написанные на том же языке. В частности:
new F – логическая формула графики, которая подставляется вместо F на каждом следующем шаге работы программы.
new B – логическая формула графики, которая подставляется вместо B.

Надо заметить, что программа MКоkh может нарисовать график сразу, а может осуществлять его пошаговое выполнение – тогда новый элемент (например, очередная ветка «растущего» куста) появляется на экране только после очередного нажатия кнопки. Это удобно для иллюстрации последовательных стадий процесса (например, роста и ветвления растения).

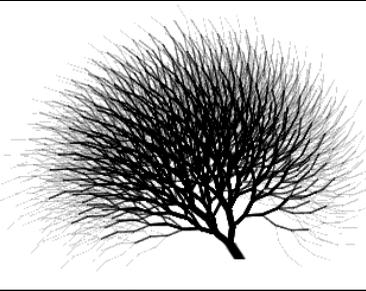
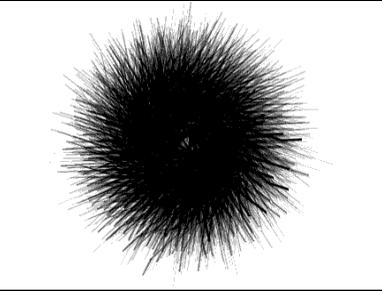
L-системы можно создавать логическим проектированием по определенному задуманному плану. Предварительно проводится расчет углов и проектируются формулы подстановки фигур в элементы формулы-аксиомы. Такой принцип работы приводит к более или менее запланированному результату, который в дальнейшем редактируется для получения наиболее совершенной системы. Пример такого подхода – работа В.О. Бариновой и А.В. Калмыкова (Баринова и Калмыков, 2002). Второй метод создания новых L-систем можно назвать методом *генетических алгоритмов*. Этот метод был заимствован у живых организмов. Он включает в себя в основном механизмы кроссинговера. В качестве «хромосом» (генотипа) L-систем выступают записи формул алгоритмов, на основании которых происходит генерация графического объекта.

Полученные результаты

Из прототипа «Дерево», взятого из стандартных образцов программы MКоkh приемами генетических алгоритмов нами получены последовательно: Дерево, Морской ёж, Ветка 1 можжевельника казацкого, Ветка 2 можжевельника казацкого. Интересно, что превращение «Дерева» в «морского ежа» происходит при весьма незначительном изменении генома – удаляется один знак – в подстановочной формуле.

Таблица 1. Последовательность 1-4 графических объектов L-систем, полученных с использованием принципов генетических алгоритмов

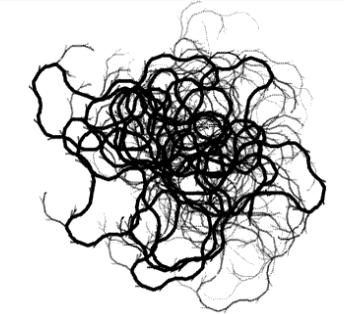
1 Дерево (прототип взят с www.techka.ru)	2 Морской ёж (автор А. В. Калмыков)
--	-------------------------------------

			
Axiom:	F	Axiom:	F
New F:	-F[-F+F-F]+[+F-F-F]	New F:	F[-F+F-F]+[+F-F-F]
XY угол pi:	1\9	XY угол pi:	1\9
Число шагов:	5	Число шагов:	5
3 «Ветка 1 можжевельника казацкого» (автор А. В. Калмыков)		4 «Ветка 2 можжевельника казацкого» (автор А. В. Калмыков)	
			
Axiom:	F-F[-F+F-F]+[+F-F-F]	Axiom:	-[F-F[-F+F-F]+[+F-F-F]]+[F-F[-F+F-F]+[+F-F-F]]
New F:	-F[-F+F-F]+[+F-F-F]-F[-F+F-F]+[+F-F-F]-F[-F+F-F]+[+F-F-F]	New F:	-F[-F+F-F]+[+F-F-F]-F[-F+F-F]+[+F-F-F]-F[-F+F-F]+[+F-F-F]
XY угол pi:	1\9	XY угол pi:	1\9
Число шагов:	3	Число шагов:	3

Семейство L-систем «Лоза» (Таблица 2) возникло как результат свободной игры с формулами без опоры на конкретный прототип. Однако идея использования переменных X и Y для получения двумерных L-систем была заимствована из формул «Кривая Серпинского» и «Ковер Серпинского», «Кривая Гильберта» и «Кривая Госпера», входящих в состав стандартных образцов программы «МКокh». Переменные X и Y, а также их

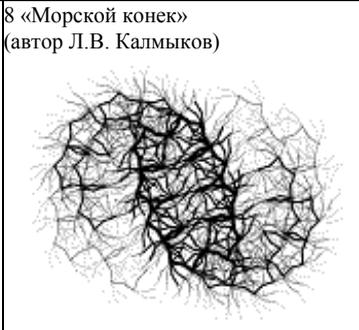
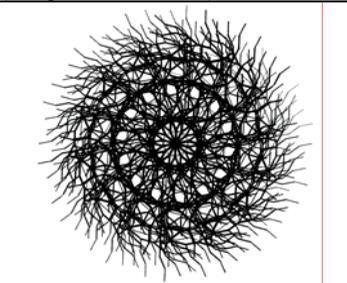
подстановки “New X” и “New Y” зарезервированы в программе для создания трехмерных L-систем. Оказывается, их можно за-действовать и в двумерном моделировании. Отличие между «Клубком лианы» и «Веточкой лианы» заключается в отсутст-вии у «Клубка лианы» внешних скобок в формуле “New Y”.

Таблица 2. Семейство L-систем «Лоза»

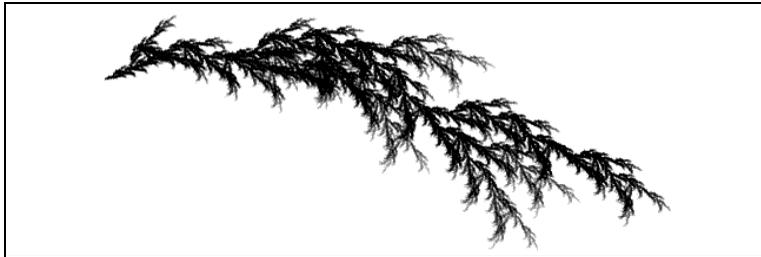
5 «Веточка лианы» (автор А. В. Калмыков)		6 «Клубок лианы» (автор А. В. Калмыков)	
			
Axiom:	FYX	FYX	
New F:	FFFXFYFX-[-FFFXFYFX]	FFFXFYFX-[-FFFXFYFX]	
New X:	Y[[XY]+X]+F[+FX]+XF	Y[[XY]+X]+F[+FX]+XF	
New Y:	[FFF[[YX]+Y]+F[+FY]+F]	FFF[[YX]+Y]+F[+FY]+F	
XY угол pi:	1/9	1/9	
Количество шагов:	4	4	

Из прототипа «Венок», взятого из стандартных образцов программы MKokh приемами генетических алгоритмов получены последовательно: Морской конек, Симметризация морского конька, Коралл, Еловая ветка (см. Табл. 3). Преобразование из «Венок» в «Морской конек» осуществлено путем удвоения «хромосомы»-алгоритма подстановочной формулы **New F**. Следующие два преобразования заключались во вставках поворотов вправо («+») внутрь подстановочной формулы. И последняя структура данной последовательности генетических преобразований – «Еловая ветка» возникла как результат четырехкратных вставок подстановочных алгоритмов в аксиому, ряда точечных мутаций и изменения величины угла поворота XY.

Таблица 3. Основные этапы создания L-системы «Еловая ветка»

7 «Венок» (образец, прилагаемый к программе MKokh)		8 «Морской конек» (автор Л.В. Калмыков)	
			
Аxiом:	F	F	
New F:	$F+F-[F-F+[F]]+[F-F+[F]]$	$F+F-[F-F+[F]]+[F-F+[F]]----$ $F+F-[F-F+[F]]+[F-F+[F]]----$	
XY угол pi:	1\7	1\7	
Число шагов:	4	3	
9 «Симметризация Морского конька» (автор Л.В. Калмыков)		10 «Коралл» (автор Л.В. Калмыков)	
			
Аxiом:	F	F	
New F:	$++F+F-[F-F+[F]]+[F-F+[F]]----F+F-[F-F+[F]]+++[F-F+[F]]----$	$++F+F-[F-F+++[F]]+++[F-F+[F]]----F+F-[F-F+[F]]+++[F-F+[F]]----$	
XY угол pi:	1\7	1\7	
Число шагов:	4	4	

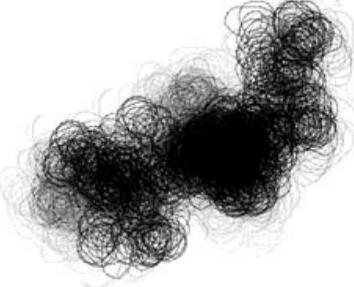
11 «Еловая ветка» (автор Л.В. Калмыков)
3 шаг перезаписи. Масштаб увеличен.



Axiom: F++F+F-[F+F[F]]-++[F-F+[F]]--F+F-++[F-F+[F]]++[F-F+[F]]--F+F+F-
 [F+F[F]]-[F-F+[F]]--F+F-++[F-F+[F]]-[F-F+[F]]----
 New F: +F+F-[F+F[F]]-++[F-F+[F]]---F+F+[F-F+[F]]+++[F-F+[F]]----
 XY угол pi: 1/9
 Количество шагов: 3

Приведем характеристики еще четырех L-систем созданных нами. Это «Куст», «Колос», «Раковина» и «Моток волос». Эти структуры были получены нами сочетанием как случайных, так и целенаправленно проектируемых модификаций алгоритмов их генерации.

12 «Куст» автор А.В. Кешелова		13 «Колос» автор – Калмыков Л.В.	
Axiom:	[---F+FF+F-F][----F+FF-F] [+++F-FF-F][+F[---F] [+++F]]+F[F[---F][+++F]]-F	Axiom:	FFFF
New F:	[F[-F-F]][-F-F]+F[+F+F-F]-F	New F:	F[F-F[F-- F]][F+F[F+++F]] [F-F[F]][F+F[F]]-
Угол поворота:	1\14 π	Угол поворота:	36/24
Число шагов:	2	Число шагов:	3

14 «Раковина» (автор Л.В. Калмыков)		15 «Моток волос» (автор Л.В. Калмыков)	
			
Axiom:	FF++XX---YYFF++ +XX---YYFF---XX+++YY	Axiom:	FFYYXX
New F:	[FXFYF-F+++F-FFX][--]	New F:	FFXY
New X:	[YY++XX+++FFFXYF-F+ +F-FFXYF-F++F-F]	New X:	Y[[XY]+X]+F[+FX] +XF+F+F+X
New Y:	FFXF++FFXF	New Y:	FF[-][[YX]+Y]+F[+FY] +F+F+Y
Угол поворота:	1/12 π	Угол поворота:	1/8 π
Число шагов:	3	Число шагов:	5

Обсуждение

С точки зрения руководителя нашей школьной Лаборатории компьютерной бионики к.б.н. В.Л. Калмыкова живые организмы можно представить себе как векторные графические трехмерные анимации, нарисованные молекулами. При этом аналогом генома служат математические формулы, на основе которых происходит генерация этих векторных графических систем. С другой стороны и компьютеры могут быть объектами изучения биологов т.к. и те и другие относятся к специфическому миру информационных процессов и к миру сложных систем одновременно. Если бы на Марсе были найдены компьютеры, можно было бы уверенно утверждать — это след живых существ. Современное научное направление «искусственная жизнь» (www.alife.org) рассматривает живое независимо от типа элементной базы его реализации. В качестве аналога фенотипов нами рассматриваются генерируемые графические объекты. Саму программу можно рассматривать как некий организм,

осуществляющий все подстановки, задаваемые геномом. Все это очень похоже на наш собственный организм – наш генотип является специфической записью выполнения ряда подстановок, на основе которых формируется фенотип. Один тип подстановок продуцирует рибонуклеиновые кислоты, другой тип подстановок продуцирует белки, «дирижирующие» многими биохимическими процессами в организме. При этом сам организм выполняет функции программы. Эта аналогия позволяет нам создавать новые типы L-систем с помощью таких же модификаций «генома» (совокупности формул сценария создания L-системы), которые реализуются в мире живого – т.е. с помощью кроссинговера, дупликаций, инверсий, включений, выпадений, разрывов и точечных модификаций алгоритмов их генерации. В современных компьютерных науках совокупность этих приемов носит название «генетических алгоритмов». Эти же приемы модификации генома активно используются и в современных гено-инженерных технологиях. Интересна и обратная аналогия – живые организмы могут рассматриваться как графические объекты, нарисованные молекулами. Кроме механизма кроссинговера, метод генетических алгоритмов включает весь набор возможных способов модификации генома: точечные мутации (замена единичных букв алфавита, возникновение разрывов), выпадения (делекции) или вставки отдельных сегментов генетической записи, в том числе и чужеродной для данного генома (взятой из другой хромосомы, из другого вида организмов или из формулы другой L-системы), инверсии (перевороты отдельных сегментов), мультипликации (удвоения, утроения,...) отдельных сегментов или целых хромосом. В отличие от логического проектирования L-системы, при котором она строится предопределенным образом (как результат проектирования), использование генетических алгоритмов зачастую приводит к неожиданным результатам. Разработчик (или природа) отбирает из полученных результатов лучшие и может использовать их для дальнейшего генетического скрещивания. В данной работе были использованы оба подхода создания новых L-систем, но предпочтение отдавалось методу генетических алгоритмов. Это предпочтение связано с тем, что в случае сложных L-систем, предсказание всех возможных вариантов их дальнейшего со-

вершенствования затруднительно из-за слишком большого объема переборов. Применение генетических алгоритмов позволяет идти большими шагами в пространстве возможных структурных переборов и получать самые неожиданные результаты. Наилучшие полученные результаты можно «скрещивать» друг с другом, ускоряя достижение искомого результата. Этот метод можно использовать даже не зная формального языка построения L-систем. Однако, использование логического проектирования L-систем иногда необходимо для окончательной доводки полученных результатов.

Заключение

С использованием принципов генетических алгоритмов и логического проектирования создано 12 новых биоподобных L-систем (систем Линденмаера). В работе приведены как графические объекты, так и логические формулы этих L-систем (аксиомы, подстановки, углы поворотов и число итераций перезаписи). Обсуждается важность данного типа моделирования для решения *проблем биологического морфогенеза и использования биологических идей в информатике*. Авторы классифицируют данное направление исследований как *«искусственную жизнь» (artificial life)*.

Планы дальнейших исследований

Генетические алгоритмы и отбор получаемых результатов производился нами «вручную», т.е. без участия специальной программы автоматического модифицирования «генома» и программы автоматического отбора получающихся результатов. Главным направлением *дальнейших исследований* планируется попытка найти возможность программным образом автоматизировать как генерацию новых L-систем на основе генетических алгоритмов, так и отбор лучших вариантов на основе формализованных эстетических решений. Если автоматизация генетических алгоритмов является в принципе уже решенной задачей, и существует множество программ в этой области, то формализация и автоматизация эстетического выбора среди случайно сгенерированных решений является нерешенной задачей, стоящей перед компьютерными науками. Возможно, в решении последней задачи окажутся полезными методы функционального программирования, использующие аппарат теории категорий и тео-

рии групп (т.е. формальных теорий красоты), а также нейрональные сети, уже применяемые сегодня для построения целевых функций отбора случайных решений. Нейрональная сеть может использоваться как многомерные суммирующие векторные весы, которые сводят многочисленные оценочные данные (вектора) к одному ответу «Да» в том случае, если векторная сумма достигает определенной величины. Дело за количественными критериями математической оценки эстетической ценности структур.

Благодарности: авторы благодарят своего научного руководителя к.б.н. Вячеслава Львовича Калмыкова за всестороннюю помощь в создании данной работы.

Литература.

1. Барина В.О. и Калмыков А.В. «Фракталы. Использование фрактальной графики на уроках биологии», Биология, N27-28; 2002 г., Изд. Первое сентября
<http://bio.1september.ru/2002/28/14.htm>
2. Kalmykov V.L. (1997) "The Abstract Theory of Evolution of the Living", in Lecture Notes in Computer Science, Springer Verlag, v.1305, pp. 43-51, <http://www.stormloader.com/theory/atel.htm>,
<http://www.iteb.serpukhov.su/gentl/atel.htm>
3. Lindenmayer A (1968). Mathematical models for cellular interaction in development I. Filaments with one-sided inputs. Journal of Theoretical Biology 18:280-289
4. Пайтген Х.-О., Рихтер П.Х. Красота фракталов. М.: Мир, 1993. (1986 - оригинал) 176 с. Prusinkiewicz, P. and Hammel, M. (1995) Visual Models of Morphogenesis: A Guided Tour - <http://www.cpsc.ucalgary.ca/projects/bmv/vmm/intro.html>, Department of Computer Science, University of Calgary
5. Федер Е. Фракталы. Пер. с англ.-М.: Мир,1991.-254с. (Jens Feder, Plenum Press, NewYork, 1988)