Результаты и их обсуждение. *Полугруппой* называется множество, с заданной бинарной алгебраической ассоциативной операцией. Если в полугруппе имеется единичный элемент, то она называется *моноидом.*

Для доказательства основного результата мы используем следующую лемму.

Лемма. *Если* $\mathfrak{F}$ *и* $\mathfrak{H}$ *– радикальные гомоморфы, то их произведение* $\mathfrak{F} \lozenge \mathfrak{H}$ *– радикальный гомоморф.*

Основным результатом работы является доказанная

Теорема. *Пусть* $\mathcal{F} = \{\mathcal{F}_i | i \in I\}$ *– множество всех радикальных гомоморфов. Тогда справедливы следующие утверждения: алгебра* $< F, \lozenge >$ *является полугруппой и моноидом.*

Заключение. Найдены новые подалгебры алгебры классов Фиттинга. Доказано, что множество радикальных гомоморфов является полугруппой и моноидом.

1. Doerk, K. Finite solvable groups / K. Doerk, T. Hawkes. – Berlin–New York : Walter de Gruyter, 1992. – 891 p.
2. Bryce, R.A., Cossey J. Finite formations of finite solvable groups / R.A. Bryce, J. Cossey // Math. Z. – 1972. – Bd. 127. – S. 217-223.

ПРИМЕНЕНИЕ ИЕРАРХИЧЕСКОЙ АГЛОМЕРАТИВНОЙ КЛАСТЕРИЗАЦИИ ДЛЯ АНАЛИЗА РЕЗУЛЬТАТОВ СЛУЧАЙНОЙ ГЕНЕРАЦИИ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ

С.А. Ермоченко, Л.В. Иванова, К.В. Прохоров
Витебск, ВГУ имени П.М. Машерова

В настоящей работе рассматривается задача генерации случайных математических функций одного аргумента, которые являются суперпозицией элементарных математических функций со случайно сгенерированными коэффициентами. Необходимость в генерировании подобных математических объектов возникает в случаях, когда необходимо создать большое количество заданий для проверки знаний по определённой теме из области математики, физики или информатики, в том числе и для систем автоматизированного тестирования. Например, для проверки сформированности навыка вычислять производную функции, необходимо подготовить набор уникальных функций, для каждой из которых в процессе генерации определяется также и правильное решение, с которым потом преподаватель или автоматизированная система сравнивает полученный обучающимся результат.

В данной работе рассматриваются лишь подходы к генерации математических функций в общем виде, без привязки к конкретным математическим или другим практическим задачам.

Цель данной работы – разработать алгоритм генерации математических функций и определить различные критерии сравнения этих функций

между собой, которые позволят параметризировать алгоритм генерации для получения функций, обладающих требуемыми свойствами.

Актуальность работы продиктована всё возрастающим интересом к системам автоматизированной проверки знаний. Применение таких систем позволяет повысить объективность оценки знаний за счёт исключения субъективного оценивания работы обучаемого преподавателем. Но системы автоматизированного тестирования требуют от преподавателя скрупулёзной работы по разработке тестовых заданий, которая, как правило, занимает достаточно много времени. Системы, позволяющие генерировать подобные тестовые задания случайным образом позволят существенно сэкономить время преподавателя.

Материалы и методы. Данное исследование базируется на понятии элементарной математической функции [1] и идея суперпозиции таких функций. Для сравнения генерируемых функций между собой и выявления общих свойств получаемых функций используется метод иерархической агломеративной кластеризации [2]. Также применяется метод объектно-ориентированного анализа, моделирования и программирования [3], метод нисходящего проектирования программного обеспечения, а также общенаучный метод анализа и обобщения.

Результаты и их обсуждение. Одним из основных требований к генератору является представление математической функции одной переменной в программном коде в таком виде, который обеспечивал бы в дальнейшем удобную обработку этой функции внутри программы, манипулирование такими объектами, их хранение и передачу между программными модулями.

Для программной реализации генератора решено было использовать принципы объектно-ориентированного программирования и язык программирования Java [4]. ООП хорошо зарекомендовало себя в крупных проектах, над которыми работают большие коллективы разработчиков. Оно обеспечивает гибкость сопровождения программного кода и позволяет удобно реализовать принцип повторного использования. Java – это язык программирования, который реализует концепции ООП в полной мере и позволяет эффективно применять принципы ООП на практике.

Для решения задачи внутреннего представления математических функций в программном коде использован шаблон проектирования «Декоратор» («Decorator») [3], для чего описывается абстрактный класс ElementaryFunction, реализующий стандартный интерфейс Function<Double, Double>. В данном абстрактном классе, в соответствии с шаблоном проектирования Decorator, описывается поле – ссылка на объект класса, реализующего интерфейс Function<Double, Double>, которая является аргументом элементарной функции, а также абстрактный защищённый метод calc(), принимающий единственный параметр типа double, и возвращающий результат типа double. Подклассы класса

ElementaryFunction должны будут описывать различные элементарные функции, реализуя этот метод. При этом с помощью стандартного метода интерфейса Function.apply() реализуется суперпозиция двух функций.

Аналогично через шаблон проектирования «Компоновщик» («Composite») [3] реализуется сумма, разность, произведение и частное двух функций.

Сам генератор случайных функций реализуется через шаблон проектирования «Строитель» («Builder») [3], который позволяет для каждой элементарной функции случайным образом генерировать коэффициенты (если они есть). Генерация коэффициентов для различных элементарных функций, или даже для одной и той же элементарной функции в разных ситуациях могут применяться разные правила генерации. Например, в одном случае для упрощения полученной функции необходимо генерировать коэффициенты квадратного трёхчлена так, чтобы они были целыми числами в определённом диапазоне, как правило на отрезке [-10, 10]. А в другом случае для этой же элементарной функции допустимо использовать вещественные числа с заданием двух знаков после запятой. А, например, для показательной функции параметр может принимать значение из строго определённого множества {2, *e*, 3, 5, 7, 10}. Такую вариативность как раз и обеспечивает шаблон «Строитель», позволяющий для каждого генератора определить свой подкласс, реализующий некоторый вариант алгоритма генерации.

Для сравнения получаемых функций между собой, классификации полученных функций по различным признакам применялись методы иерархической агломеративной кластеризации. Основным вопросом для проведения кластеризации является критерии сравнения двух функций. В данной работе применялись критерии, позволяющие оценить сложность функции, такие как уровень вложенности элементарных функций; интегрированный показатель сложности, основанный на субъективном ранжировании элементарных функции по степени сложности работы с ними у обучающихся; область определения функции; сложность перечисленных показателей для производной данной функции.

Для оценки получаемых функции применялась генерация достаточно большого числа функций (до 500 функций). Результат оценивался с помощью построения дендрограмм. Библиотека построения дендрограмм разрабатывалась на языке программирования JavaScript и в отличие от имеющихся стандартных инструментов математических пакетов позволяет более удобно работать с большим числом объектом, входящим в кластер.

Анализ сгенерированных функций позволил определить, какие параметры разработанных генераторов позволяют получать функцию с заданным характеристиками.

Заключение. Таким образом, в результате проделанной работы разработана система генерации математических функций, которая позволяет гибко модифицировать систему под требования различных задач, косвенно

оценивать сложность сгенерированной функции, выполнять различные манипуляции с полученными функциями.

Также одним из полученных результатов является разработанная библиотека языка программирования JavaScript по визуализации результатов иерархического агломеративного кластерного анализа.

Работа выполнена в рамках НИР «Методы искусственного интеллекта для оптимизации образовательного процесса, №ГР 20210790» задания «Информационные технологии повышения качества образовательного процесса» ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства».

1. Выгодский, М. Я. Справочник по высшей математике. – Москва: АСТ, 2019. – 703 с.

2. Жамбю, М. Иерархический кластер-анализ и соответствия. – Москва: Финансы и статистика, 1988. – 345 с.

3. Гамма, Э. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р Хелм, Р. Джонсон, Д. Влиссидес. – Санкт-Петербург: Питер, 2020. – 368 с.

4. Gosling, J. The Java® Language Specification. Java SE 17 Edition [Electronic Resource] / J. Gosling, B. Joy, G. Steele etc. – Oracle Inc., 2021. – Mode access: https://docs.oracle.com/javase/specs/jls/se17/html/index.html. – Date access: 23.01.2022

О КОМПОЗИЦИОННЫХ ФОРМАЦИЯХ СО СТОУНОВОЙ РЕШЕТКОЙ ПОДФОРМАЦИЙ

А.П. Мехович, А.Ю. Столяренко
Витебск, ВГУ имени П.М. Машерова

Результат 1986 года А.Н. Скибы (О локальных формациях длины 5 // Арифметическое и подгрупповое строение конечных групп) получил развитие в различных направлениях. В частности, в работе А.Н. Скибы и Н.Н. Воробьева (Алгебра и дискретная математика, 2007) описаны стоуновы решетки n-кратно локальных и тотально локальных классов Фиттинга, в работе Н.Н. Воробьева (Весці НАН Беларусі, 2008) описаны стоуновы решетки n-кратно насыщенных и тотально насыщенных формаций.

Сказанное позволяет утверждать, что исследование стоуновых решеток является актуальной задачей.

Цель настоящей работы – нахождение условий, при которых решетка n-кратно ω-композиционных формаций является стоуновой.

Материал и методы. Используется терминология и методы исследования конечных групп и их классов, а также теории решеток.

Результаты и их обсуждение. В данной работе рассматриваются только конечные группы. Используется стандартная терминология теории групп и их классов, а также теории решеток. Все необходимые обозначения и термины можно найти в [1–3].

Для произвольной решетки $<L, \wedge, \vee>$ с нулем элемент a^* называется псевдодополнением элемента a, если a^* – наибольший элемент в решетке