Теорема 1. Если вершина $v(S_0)$ графа преемников онд-автомата R разрешима, то для любой сильносвязной реализации автомата R, находящейся в одном из состояний $s \in S_0$, возможен простой условный установочный эксперимент.

Конструктивное доказательство теоремы, которое мы опускаем, позволяет предложить следующий метод построения простого условного эксперимента, идентифицирующего реализацию A онд-автомата R. Построить граф преемников автомата R и вычислить его разрешимые вершины. Если вершина $v(S)$ разрешима, то провести над A сначала простой условный установочный эксперимент, а затем эксперимент по идентификации автомата A при известном его начальном состоянии.

Предложенный метод реализован на языке C++ и апробирован на случайно сгенерированных автоматах. Компьютерные эксперименты показали, что для подавляющего большинства случайно сгенерированных онд-автоматов вершина $v(S)$ графа преемников разрешима, а следовательно, метод применим для практического использования.

ЛИТЕРАТУРА

1. *Гилл А.* Введение в теорию конечных автоматов. М.: Наука, 1966. 272 с.
2. *Тренькаев В. Н.* Реализация шифра Закревского на основе перестраиваемого автомата // Прикладная дискретная математика. 2010. № 3. С. 69–77.
3. *Жуковская А. О., Тренькаев В. Н.* О простых условных экспериментах идентификации обратимых автоматов некоторого класса // Прикладная дискретная математика. Приложение. 2016. № 9. С. 115.

УДК 003.26, 004.021, 519.725.2 DOI 10.17223/2226308X/10/56

ПРИМЕНЕНИЕ РЁБЕРНОГО ЛОКАЛЬНОГО ДОПОЛНЕНИЯ В СТРУКТУРНОМ АНАЛИЗЕ КРИПТОСИСТЕМЫ МАК-ЭЛИСА

А. А. Соколова

Предлагается алгоритм для нахождения и перечисления классов эквивалентности циклических кодов с помощью графов и операции рёберного локального дополнения. Удалось увеличить максимальное количество вершин для обрабатываемого графа с 10 до 17. Построена полная классификация циклических кодов длины 19. Кроме того, реализован алгоритм для определения эквивалентности двух кодов, один из которых циклический. На персональном компьютере достигнута возможность за приемлемое время определять эквивалентность кодов длины 19.

Ключевые слова: *двоичные линейные коды, классификация, графы, рёберное локальное дополнение, криптосистема Мак-Элиса.*

В криптосистеме Мак-Элиса одному и тому же открытому ключу могут соответствовать несколько секретных ключей, следовательно, они могут быть разбиты на классы эквивалентности. Возможность подобрать эквивалентный ключ напрямую влияет на стойкость данного метода шифрования, так как ключ для расшифрования не уникален.

Имеющиеся исследования в данной области рассматривают ограниченный набор кодов с тривиальной группой автоморфизмов и неприменимы в структурном анализе криптосистемы Мак-Элиса. Возникает проблема поиска альтернативы предложенным методам, применимой к циклическим кодам. Вопрос изучения классов эквивалентности двоичных линейных кодов является основным в данной работе.

С помощью простых преобразований двоичный линейный код можно представить в виде двудольного графа. Введённая в [1] операция ELC (Edge Local Complemetation, рёберное локальное дополнение) над ним позволяет рассматривать различные эквивалентные коды, а орбита двудольного графа под ELC является полным классом эквивалентности для кода, отвечающего данному графу. Этот способ подходит для дальнейших исследований, поскольку в целом не зависит от структуры кода. Данный метод представления и обработки графов позволяет по-новому представить классы эквивалентности двоичных кодов и классифицировать все ELC-орбиты кодов различной длины.

Двоичный линейный $[n, k]$-код C соответствует $(k, n-k)$-двудольному графу G на n вершинах с матрицей смежности определённого вида. Применение любой последовательности ELC-операций к графу G, соответствующему коду C, преобразует его в граф, код которого эквивалентен C. Пусть коды C и C' эквивалентны, их порождающие матрицы соответственно $\mathcal{C}$ и $\mathcal{C}'$, G и G' — двудольные графы, отвечающие $\mathcal{C}$ и $\mathcal{C}'$. Тогда G' изоморфен графу, полученному последовательностью ELC-операций над G.

Для исследования реализованы две программы на языке C. Одна из них строит по длине порождающего полинома циклического кода все возможные графы, проводит различное количество ELC-преобразований, находит изоморфные графы и подсчитывает количество классов эквивалентности (число орбит).

Вторая программа принимает на вход длину двух кодов (первый — циклический) и их значения и с помощью ELC определяет, являются ли эти два кода эквивалентными.

Выбор языка программирования обусловлен тем, что для нахождения изоморфных графов использована программа nauty [6], написанная на C.

Поскольку в классической теории кодирования имеют применение только двудольные графы, непосредственным предметом изучения являются только они. В качестве входного набора данных генерируются все возможные порождающие многочлены для указанной пользователем длины.

Любому двоичному линейному коду и соответствующей ему порождающей матрице можно сопоставить некоторый двудольный граф и его матрицу смежности и обратно. Поэтому вместо графов удобнее создавать порождающие матрицы кодов и работать с ними.

В таблице приведены количество классов эквивалентности и затраченное на обработку время для многочленов различной длины.

Длина многочл.	Длина кода	Классы эквивал.	Время работы
3	7	1	$<$ 0,01 с
4	9	1	$<$ 0,01 с
5	11	3	$<$ 0,01с
6	13	12	$<$ 0,01с
7	15	31	7 с
8	17	68	3663 с
9	19	168	707691 с (8 дней)

Вычисления проводились на процессоре Intel Core i7 2.50 GHz.

Помимо нахождения количества орбит, решается проблема поиска всех эквивалентных циклических кодов. При переборе кодов каждому из них сопоставляется орбита, после чего начинает обрабатываться следующий код и проверяется его присутствие в ранее найденных орбитах. Так как операция ELC над циклическими кодами замкнута, то есть не выводит за рамки циклических кодов, можно сказать, что все орбиты

будут состоять только из них и, следовательно, находя все орбиты, мы находим все циклические коды, эквивалентные данному.

В работе [1] на суперкомпьютере подсчитаны классы эквивалентности для графов с количеством вершин не более 12. На пользовательском компьютере за адекватное время удалось реализовать вычисления для 10 вершин. В данной работе только на пользовательском компьютере удалось корректно работать с графами с 17 вершинами (порождающий многочлен длины 9, код длины 19).

Более важной задачей является сравнение двух кодов, один из которых циклический, на эквивалентность. Реализована программа, решающая эту задачу. Сравнение происходит следующим образом: для данного циклического графа строится орбита, после чего в ней ищется второй граф, приведённый к требуемому виду. В случае обнаружения можно утверждать, что графы эквивалентны, иначе — обратное. На персональном компьютере за приемлемое время определяется эквивалентность или её отсутствие для кодов длины 19 (граф с 37 вершинами).

В ходе работы рассмотрена и изучена операция рёберного локального дополнения и эквивалентность двудольных графов и, как следствие, двоичных линейных кодов. Для циклических кодов улучшен достигнутый ранее результат скорости и качества вычислений их классов эквивалентности, а также реализован эффективный алгоритм сравнения на эквивалентность двух кодов, один из которых циклический.

ЛИТЕРАТУРА

1. *Danielsen L. E. and Parker M. G.* Edge local complementation and equivalence of binary linear codes // Des. Codes Cryptogr. 2008. No. 49. P. 161–170.
2. Nauty and Traces User's Guide (Version 2.5). `http://users.cecs.anu.edu.au/\textasciitildebdm/nauty/nug25.pdf`