



ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

**Кафедра
«Криптология и кибербезопасность»**

ОТЧЕТ о лабораторной работе №2

Выполнил студент
группы Б20-505
Соколов Александр Дмитриевич

Москва – 2023

1 Исследование функций S7 и S9

1.1 Получить следующие характеристики разрядных функций

- таблицу истинности;
- АНФ;
- вес;
- число мономов для многочлена Жегалкина каждой функции;
- число мономов во всех многочленах Жегалкина разрядных функциях (т.е. мощность объединения множеств мономов многочленов Жегалкина разрядных функций);
- коэффициенты Фурье и Адамара-Уолша, соответственно список линейных аналогов и соответствующих вероятностей совпадения разрядной функции с линейной (аффинной);
- действительный многочлен, его степень и число мономов;
- число мономов во всех действительных многочленах (т.е. мощность объединения множеств мономов действительных многочленов разрядных функций);
- действительный многочлен, его степень и число мономов, а также частные производные в точке $(0.5, \dots, 0.5)$ по каждой переменной. Просто генерируем вектор длины 2^n с помощью генератора случайных чисел

Так как все пункты уже были расписаны в лабораторной работе номер 1, то дополню только результатами анализа функций S7, S9

S7, S9

Из того, что функция F7[{0, 1, 1, 0, 1,, 0}] дала результат {0, 1, 0, 1, 1, 1, 0} (что совпадает с результатом из документа)

А также, что полиномы Жегалкина разрядных функций совпадают с функциями из документа

Я сделал вывод, что все преобразования из упакованных в целые числа значений S7 в вектор функцию были сделаны верно.

Аналогично для S9.

1.2 Получение коэффициентов Адамара-Уолша вектор-функций S7, S9 и их статистику (значения коэффициентов и сколько раз они встретились)

Добавлена функция Counted, которая считает количество появлений каждого элемента.

2 Для линейного автомата, определяемого матрицей A и вектором B

построить эквивалентный регистр сдвига (записать рекуррентный закон, характеристический многочлен);

Пусть имеется линейный автомат, заданный

- Невырожденной матрицей линейного оператора A размера $n \times n$
- Линейной функцией выхода, представленной в виде вектора B размера n
- Начальным состоянием S размера n

Выходная последовательность имеет вид:

$$(\vec{B} * (A^i * \vec{S})), i = 0..$$

Производящая функция данной последовательности будет

$$\sum_{i=0}^{\infty} (\vec{B} * (A^i * \vec{S})) * x^i = \vec{B} * \left(\sum_{i=0}^{\infty} (Ax)^i \right) * \vec{S} = \vec{B} * (E - Ax)^{-1} * \vec{S} = \frac{\vec{B} * M(x) * \vec{S}}{\det(E - Ax)}$$

Где

- $M(x)$ - матрица алгебраических дополнений, матрицы $E - Ax$
- E - единичная матрица
- $\det(E - Ax)$ - характеристический многочлен последовательности

Таким образом мы получим в знаменателе многочлен степени n со свободным членом 1 (так как матрица невырожденная и мы работаем в поле по модулю 2), а в числителе многочлен степени не больше $n - 1$. То есть рациональную производящую функцию некоторой рекуррентной последовательности.

Однако в зависимости от начального состояния, если знаменатель факторизуется в данном поле, то возможно понижение линейной сложности последовательности, в чем мы убедимся позже.

Обобщенный рекуррентный закон получаем из характеристического многочлена:

$$Q(x) = 1 + q_1 * x + q_2 * x^2 + \dots + q_d * x^d$$

Рекуррентный закон будет:

$$f(m + d) + q_1 * f(m + d - 1) + q_2 * f(m + d - 2) + \dots + q_d * f(m) = 0$$

В блокноте представлены примеры нахождения обоих параметров для матрицы 4×4 , а также проверка что выходные последовательности совпадают, с помощью функций `getn(A, B, C, n)` и `LFSRgetn(rel, n)`

определить, является ли характеристический многочлен приводимым

Вызываем функцию `Factor[poly, Modulus -> 2];`

найти цикловую структуру состояний автомата (длины циклов и число циклов каждой длины).

Самый простой способ - просто проитерироваться по всем состояниям и искать совпадения, однако это было долго, поэтому я использовал следующий алгоритм:

- Во-первых, так как в общем случае считать $\vec{B} * (A - Ex)^{-1} * \vec{S}$ очень долго, можно посчитать $P(x)$ (числитель производящей функции) напрямую:

$$p_0 = f(0)$$

$$p_1 = f(1) + q_1 * f(2)$$

...

$$p_{d-1} = f(d-1) + q_1 * f(d-2) + \dots + q_{d-1} * f(0)$$

- Во-вторых для каждого состояния я находил характеристический многочлен, факторизовал его и находил предположительные периоды которые возможны:

для характеристического многочлена $f(x) = g_1(x)^{e_1} * \dots * g_m(x)^{e_m}$ предположительные длины периодов будут делителями числа $LCM(2^{(e_i-1)*d_i} * (2^{d_i} - 1))$, где d_i - степень многочлена $g_i(x)$.

Для этого была реализована функция *orderpoly*

- Так как для каждого характеристического многочлена период будет одинаковый (вся последовательность определена состоянием), то при переборе это помогает снизить время проверки длины последовательности.
- Так как возведение матрицы в степень по модулю не поддерживается в моей версии Mathematica, реализована функция быстрого возведения в степень $Mrow(A, n, k)$
- Так же все степени матрицы, которые в итоге понадобятся, посчитаны заранее.

3 Найти все функции де Брёйна от 4 переменных. Для каждой функции привести

- цикл вершин графа;
- таблицу (истинности);
- многочлен Жегалкина (и указать число одночленов);
- коэффициенты Адамара-Уолша.

Для поиска функций де Брёйна от n переменных, составляем ориентированный граф из 2^{n-1} вершин, в котором каждая вершина соответствует бинарному вектору длины $n - 1$.

$v_i \rightarrow v_j$ если в бинарном представлении $\langle i \rangle_{n-1} = (a_1, a_2, \dots, a_{n-1})$, а $\langle j \rangle_{n-1} = (a_2, a_3, \dots, a_{n-1}, b)$, с $b = 0, 1$

В этом графе мы ищем Эйлеров цикл, тем самым строя Гамильтонов цикл для вершин (a_1, \dots, a_{n-1}, b) . И на основе этого цикла строим булеву функцию от n переменных.

Так как алгоритм поиска Эйлерова цикла в общем случае случаен, с помощью него нельзя с уверенностью найти ВСЕ присутствующие Эйлеровы циклы, поэтому был реализован рекурсивный алгоритм на основе полного перебора.

Чтобы проверить, что были найдены действительно все циклы(функции), нужно сравнить число найденных циклов с числом $\frac{2^{2^{n-1}}}{2^n}$

В нашем случае при $n = 4$, число функций будет $\frac{2^{2^3}}{2^4} = 16$