

Министерство образования и науки Российской Федерации  
(МИНОБРНАУКИ РОССИИ)  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)  
Институт прикладной математики и компьютерных наук  
Кафедра защиты информации и криптографии

## **КУРСОВАЯ РАБОТА**

БИБЛИОТЕКА ДЛЯ РАБОТЫ С БУЛЕВЫМИ ФУНКЦИЯМИ ДЛЯ ЯЗЫКА  
ПРОГРАММИРОВАНИЯ LYAPAS

Муругов Михаил Алексеевич

Руководитель  
канд. физ.-мат. наук, доцент  
\_\_\_\_\_ И.А.Панкратова  
« \_\_\_\_\_ » \_\_\_\_\_ 201 \_\_\_\_ г.

Студент группы № 1155  
\_\_\_\_\_ М.А.Муругов

Томск 2018

## ОГЛАВЛЕНИЕ

Введение.....	2
1 Описание алгоритмов на математическом языке.....	3
1.1 Принадлежность булевой функции к классу $T^0$	
1.2 Принадлежность булевой функции к классу $T^1$	
1.3 Принадлежность булевой функции к классу монотонных булевых функций	
1.4 Преобразование Мёбиуса булевой функции	
1.5 Принадлежность булевой функции к классу линейных булевых функций	
1.6 Отражение вектора значений булевой функции	
1.7 Принадлежность булевой функции к классу самодвойственных булевых функций	
2 Идеи программных реализаций	
2.1 Принадлежность булевой функции к классу $T^0$	
2.2 Принадлежность булевой функции к классу $T^1$	
2.3 Принадлежность булевой функции к классу монотонных булевых функций	
2.4 Преобразование Мёбиуса булевой функции	
2.5 Принадлежность булевой функции к классу линейных булевых функций	
2.6 Отражение вектора значений булевой функции	
2.7 Принадлежность булевой функции к классу самодвойственных булевых функций	
3 Экспериментальные данные	
4 Заключение	
Список использованных источников и литературы	
Приложения	

## ВВЕДЕНИЕ

Целью этой курсовой работы было написание библиотеки для работы с булевыми функциями (определение принадлежности к замкнутым классам, различные преобразования и т.д.) для языка программирования LYaPAS. В дальнейшем планируется, что эта библиотека будет использоваться для реализации криптографических алгоритмов и прочих нужд.

Для криптографии булевы функции важны т.к. они, в частности, используются в качестве комбинирующих и фильтрующих функций при построении поточных шифров; для блочных шифров они используются в качестве функций блоков замены и т.д.

В нынешнее время язык LYaPAS уже выигрывает по скорости на некоторых алгоритмах, но всё ещё требует доработок и улучшений, в следствие чего и был выбран.

В качестве базиса в языке уже реализованы побитовые операции для булевых векторов любой длины, а также для 32-х битных векторов функция подсчёта веса и генерация псевдослучайного вектора. Всё это используется в настоящей работе для реализации более сложных вещей относительно булевых функций.

## ОПИСАНИЕ АЛГОРИТМОВ НА МАТЕМАТИЧЕСКОМ ЯЗЫКЕ

### *Принадлежность булевой функции к классу $T^0$*

**Определение.** Булева функция сохраняет константу 0 (принадлежит классу  $T^0$ ), если на наборе из всех нулей функция принимает значение нуль.

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  принадлежит классу  $T^0$ ?”

Шаг 1) Если  $f(0, 0, \dots, 0) = 0$ , то ответ “Да”

Иначе ответ “Нет”

### *Принадлежность булевой функции к классу $T^1$*

**Определение.** Булева функция сохраняет константу 1 (принадлежит классу  $T^1$ ), если на наборе из всех единиц функция принимает значение единица.

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  принадлежит классу  $T^1$ ?”

Шаг 1) Если  $f(1, 1, \dots, 1) = 1$ , то ответ “Да”

Иначе ответ “Нет”

### *Принадлежность булевой функции к классу монотонных булевых функций*

**Определение.** Булева функция  $f(x_1, \dots, x_n)$  называется монотонной (принадлежит классу  $M$ ), если для любой пары наборов  $\alpha$  и  $\beta$  таких, что  $\beta \succeq \alpha$ , выполняется условие  $f(\beta) \geq f(\alpha)$ .

Алгоритм определения принадлежности булевой функции к классу монотонных булевых функций приведён в [1].

### Преобразование Мёбиуса булевой функции

**Определение.** Положительной конъюнкцией называется элементарная конъюнкция, не содержащая инверсий переменных. Договоримся обозначать положительную конъюнкцию через  $K^+$ .

**///Определение АНФ взять из “Булевы функции в криптографии”! (не нашёл)**

**Определение.** Полиномом Жегалкина, или алгебраической нормальной формой (АНФ), булевой функции  $f(x_1, \dots, x_n)$  называется дизъюнкция с исключением различных положительных конъюнкций переменных из множества  $X = \{x_1, \dots, x_n\}$ , то есть формула вида  $P = K_1^+ \oplus \dots \oplus K_p^+$ , задающая функцию  $f(x_1, \dots, x_n)$ .

**Определение.** Преобразованием Мёбиуса называется функция  $\mu: P_2(n) \rightarrow P_2(n)$ , где  $P_2(n)$  – множество всех булевых функций от  $n$  переменных. С помощью преобразования Мёбиуса решается задача построения АНФ булевой функции, и вычислить его значения для функции  $f(x)$  можно по формуле  $g(a) = \bigoplus_{a \succeq x} f(x)$ , где  $g = \mu(f)$ .

Преобразование Мёбиуса связано с полиномом Жегалкина следующим образом: значение  $\mu(f)$  на наборе аргументов говорит о том, есть ли положительная конъюнкция аргументов со значением 1 из этого набора в АНФ функции  $f$  (1 – положительная конъюнкция есть, 0 – положительной конъюнкции нет). Набор аргументов  $(0, \dots, 0)$  соответствует константе 1.

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход:  $g = \mu(f)$

Шаг 1)  $g := f$

Шаг 2) Для всех  $a = (a_1, \dots, a_n)$ :

Шаг 2.1)  $g(a) = \bigoplus_{x \preceq a} f(x)$

### ***Принадлежность булевой функции к классу линейных булевых функций***

**Определение.** *Длиной* булева вектора назовем количество его компонент, а *весом* вектора – количество компонент, равных единице

Длину булева вектора  $a$  в дальнейшем будем обозначать  $l(a)$ . Запись  $l(f)$ , где  $f$  – булева функция, будет обозначать длину вектора её значений.

Вес булева вектора  $a$  в дальнейшем будем обозначать  $w(a)$ . Запись  $w(f)$ , где  $f$  – булева функция, будет обозначать вес вектора её значений.

**Определение.** *Длиной полинома Жегалкина* назовем количество конъюнкций в полиноме, а его *степенью* – наибольший из рангов конъюнкций, входящих в полином.

**Определение.** Полином Жегалкина называется *линейным*, если его степень не превышает единицы.

**Определение.** Булева функция называется *линейной* (*принадлежит классу  $L$* ), если ее полином Жегалкина линеен.

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  – линейна?”

Шаг 1)  $g := \mu(f)$

Шаг 2) Если полином Жегалкина, построенный по коэффициентам  $g$  линеен, то ответ “Да”

Иначе ответ “Нет”

### ***Отражение вектора значений булевой функции***

**Определение.** *Отражением* вектора значений булевой функции является обмен значениями на противоположных наборах аргументов.

В дальнейшем отражение вектора значений булевой функции  $f$  будем обозначать  $f^R$

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход:  $f^R(x_1, \dots, x_n)$

Шаг 1) Для всех  $a = (a_1, \dots, a_n)$  таких, что  $a_1 = 0$ :

Шаг 1.1)  $f(a) \leftrightarrow f(\bar{a})$

### ***Принадлежность булевой функции к классу самодвойственных булевых функций***

**Определение.** Булева функция  $f(x_1, \dots, x_n)$  называется *двойственной булевой функции*  $g(x_1, \dots, x_n)$ , если она получена из  $g(x_1, \dots, x_n)$  инверсией всех аргументов и самой функции, то есть  $f(x_1, \dots, x_n) = \overline{g(\overline{x_1}, \dots, \overline{x_n})}$ .

**Определение.** Булева функция  $f(x_1, \dots, x_n)$  *самодвойственна (принадлежит классу  $S$ )*, если она равна двойственной себе функции, то есть  $f(x_1, \dots, x_n) = \overline{f(\overline{x_1}, \dots, \overline{x_n})}$ .

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  – самодвойственна?”

Шаг 1) Для всех векторов  $a = (a_1, \dots, a_n)$  таких, что  $a_1 = 0$ :

Шаг 1.1) Если  $f(a) \neq \overline{f(\overline{a})}$ , то ответ “Нет”

Шаг 2) Ответ “Да”

## ИДЕИ ПРОГРАММНЫХ РЕАЛИЗАЦИЙ

Перед изложением дальнейшего материала необходимо кое-что обозначить:

Во-первых, булевы функции в языке LYaPAS представляются векторами их значений.

Во-вторых, вектора значений булевых функций хранятся в логических комплексах  $L$ , каждый элемент которого занимает в памяти 4 байта (32 бита). Таким образом, т.к.  $l(f(x_1, \dots, x_n)) = 2^n$ , то функция до 5 аргументов включительно помещается в один элемент комплекса. От 6 в 2 элемента, от 7 в 4 и т.д. Количество элементов комплекса, необходимых для хранения функции от  $n$  аргументов можно вычислить по формуле  $Q = \left\lceil \frac{2^n + 31}{32} \right\rceil$ .

В-третьих, значения булевой функции в памяти хранятся в привычном нам порядке (младшие биты справа, старшие биты слева), при этом нулевой бит нулевого элемента комплекса соответствует значению  $f(0, \dots, 0)$ , следующий за ним  $f(0, \dots, 0, 1)$  и т.д.

### *Принадлежность к классу $T^0$*

Проверка булевой функции на принадлежность к классу  $T^0$  тривиальна. Необходимо просто посмотреть на первый бит вектора её значений. Если этот бит равен нулю, то функция сохраняет константу 0.

### *Принадлежность к классу $T^1$*

Для проверки принадлежности булевой функции к классу  $T^1$  необходимо посмотреть на старший бит вектора её значений. Если этот бит равен 1, то функция сохраняет константу 1. Но проверка булевой функции на принадлежность к классу  $T^1$  немного сложнее, чем к классу  $T^0$ , т.к. у функций, зависящих от  $n \leq 5$  аргументов старший бит вектора значений находится в нулевом элементе комплекса и его сначала необходимо найти. В общем же случае найти старший бит вектора значений функции можно по следующим правилам:  $i = \left\lceil \frac{2^n + 31}{32} \right\rceil - 1$ ,  $j = 2^n \pmod{32}$ , где  $i$  – индекс элемента комплекса, а  $j$  – номер бита в элементе с индексом  $i$ .



### ***Принадлежность булевой функции к классу монотонных булевых функций***

Т.к. булева функция помещена в «блоки» по 32 бита, то перед стартом рекурсии можно проверить её на монотонность вплоть до 5 компоненты следующими действиями:

$L1i < 1 \ \& \ AAAAAAAAAh \Rightarrow a$	***Проверяем на монотонность на наборах,
$a \ \& \ L1i \oplus a \mapsto 2$	***соседних по пятой компоненте
$L1i < 2 \ \& \ CCCCCCCCCh \Rightarrow a$	***Проверяем на монотонность на наборах,
$a \ \& \ L1i \oplus a \mapsto 2$	***соседних по четвёртой компоненте
$L1i < 4 \ \& \ F0F0F0F0h \Rightarrow a$	
$a \ \& \ L1i \oplus a \mapsto 2$	***...
$L1i < 8 \ \& \ FF00FF00h \Rightarrow a$	
$a \ \& \ L1i \oplus a \mapsto 2$	
$L1i < 16 \Rightarrow a$	***Проверяем на монотонность на наборах,
$a \ \& \ L1i \oplus a \mapsto 2$	***соседних по первой компоненте

После того, как каждый элемент комплекса проверен таким образом и немонотонность не обнаружена, то запускается рекурсивная функция, которая проверяет на монотонность по следующим компонентам.

### ***Преобразование Мёбиуса булевой функции***

Как следует из способа, изложенного в [2], преобразование Мёбиуса рекурсивно реализуется по следующему алгоритму:

Шаг 1) Разбиваем вектор значений булевой функции на младшую и старшую часть  $f^0$  и  $f^1$  соответственно

Шаг 2)  $f^1 := f^0 \oplus f^1$

Шаг 3) Если  $l(f) = 2$ , то выход

Иначе выполнить эту процедуру для  $f^0$  и  $f^1$

### *Принадлежность булевой функции к классу линейных булевых функций*

Алгоритм на проверку принадлежности булевой функции довольно прост. Необходимо выполнить преобразование Мёбиуса для этой функции и посмотреть, есть ли хотя бы одна единица на наборе аргументов с более, чем одной единицей.

Алгоритм:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  – линейна?”

Шаг 1)  $g := \mu(f)$

Шаг 2)  $g(0, \dots, 0) := 0$

Шаг 3) Для всех  $a = (a_1, \dots, a_n)$  таких, что  $w(a) = 1$ :

Шаг 3.1)  $g(a) := 0$

Шаг 4) Если  $w(g) = 0$ , то ответ “Да”

Иначе ответ “Нет”

### Отражение вектора значений булевой функции

Отражение вектора значений булевой функции программно довольно нетривиально, т.к. нет таких средств, которые позволили бы сделать это за одну операцию. В языке LYaPAS, т.к. вектора значений булевых функций разбиты на «блоки» по 32 бита, преобразование выполняется следующим образом: сначала выполняется отражение каждого «блока» по отдельности, затем первый «блок» меняется местами с последним, второй с предпоследним и т.д.

Также, т.к. вектора значений булевых функций от  $n \leq 5$  переменных включительно помещаются в один элемент логического комплекса L, то после отражения таких векторов их необходимо будет побитово сдвинуть вправо на  $32 - 2^n$  бита, т.к. после отражения младшие биты станут старшими в 32-х битном «блоке».

Функция, выполняющая отражение 32-х битного «блока»:

```
reverseBits(a/b)
***a – входной вектор
***b – отраженный вектор a
a > 1 & 55555555h ⇒ v
a & 55555555h < 1 ∨ v ⇒ b ***Меняем местами чётные и нечётные биты
b > 2 & 33333333h ⇒ v
b & 33333333h < 2 ∨ v ⇒ b ***Меняем местами пары битов
b > 4 & 0f0f0f0fh ⇒ v
b & 0f0f0f0fh < 4 ∨ v ⇒ b ***Меняем местами последовательности из 4-х битов
b > 8 & 00ff00ffh ⇒ v
b & 00ff00ffh < 8 ∨ v ⇒ b ***Меняем местами байты
b > 16 ⇒ v
b < 16 ∨ v ⇒ b ***Меняем местами 2-х байтовые слова
**
```

### Принадлежность булевой функции к классу самодвойственных булевых функций

Программно проверка на принадлежность булевой функции к классу самодвойственных булевых функций реализована согласно следующему алгоритму:

Вход:  $f(x_1, \dots, x_n)$  – булева функция

Выход: “ $f$  – самодвойственна?”

Шаг 1) Разбиваем вектор значений булевой функции на младшую и старшую часть  $f^0$  и  $f^1$  соответственно

Шаг 2) Если  $f^0 = \overline{(f^1)^R}$ , то ответ “Да”

Иначе ответ “Нет”

$\overline{(f^1)^R}$  – этим самым действием мы сопоставляем значения функции на противоположных наборах аргументов, а затем проверяем условие  $f(x_1, \dots, x_n) = \overline{f(\overline{x_1}, \dots, \overline{x_n})}$ .

## **ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ**

**//TODO**

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ**

1. Быкова С.В. Учебно-методический комплекс «Булевы функции». Томск 2006.
2. Панкратова И.А. Учебное пособие «Булевы функции в криптографии». Томск 2014.