

ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«ИНФОРМАТИКА И УПРАВЛЕНИЕ»
РОССИЙСКОЙ АКАДЕМИИ НАУК

на правах рукописи

УДК 519.714

Егорова Евгения Кирилловна

**Редукция количества вхождений переменных
для некоторого класса булевых функций**

специальность 05.13.17 — Теоретические основы информатики

Диссертация на соискание учёной степени
кандидата физико-математических наук

научный руководитель
проф., д. ф.-м. н. Цурков В. И.

Москва 2018

Оглавление

Введение	5
1 История возникновения, становления и развития математической логики, математический аппарат, решаемые задачи	9
1.1 История создания и развития алгебры логики	9
1.2 Математический аппарат для синтеза логических формул	12
1.2.1 Булевы функции и реализующие их формулы	12
1.2.2 Полные системы булевых функций и базисы. Показатели качества	15
1.2.3 Методы минимизации	16
1.2.4 Полиномы Жегалкина	18
1.3 Обзор проблемы нахождения минимальных оценок	19
1.4 Обзор современных проблем	20
1.4.1 Представление полинома Жегалкина в виде гиперграфа	20

2 Оценки качества булевых функций на основе алгебраической декомпозиции	26
2.1 Аналитическое определение оценок сложности	26
2.2 Верхние оценки для класса БФ $F_{n-2}^{(n)}$	28
2.2.1 Показатель качества L_B	29
2.2.2 Показатель качества L_F	30
2.3 Минимизация полученных оценок	30
2.3.1 Показатель качества $L_B \left(F_{n-2}^{(n)}, G_3 \right)$	31
2.3.2 Показатель качества $L_F \left(F_{n-2}^{(n)}, G_3 \right)$	38
2.4 Реализация в классе схем	46
2.4.1 Показатель качества $L_S \left(F_{n-2}^{(n)}, G_1 \right)$	46
2.5 Сравнение результатов	60
2.5.1 Сравнение оценок L_B	60
2.5.2 Сравнение оценок L_S	64
3 Вычислительный метод определения оценки сложности.	65
3.1 Разработка алгоритма	65
3.1.1 Используемые обозначения	66
3.1.2 Алгоритм	70
3.2 Программная реализация	73
3.2.1 Пример работы программы	73
3.2.2 Примеры работы алгоритма	75
3.2.3 Частные случаи	77
Заключение	82

Литература	84
Приложение	93

Введение

Современные достижения в области науки и техники основываются на использовании прямо и косвенным образом различных информационных систем (интеллектуальных, кибернетических, поисковых аналитических автоматизированных и др.).

В таких системах нуждается образование, здравоохранение, космос, экономика, энергетика, робототехника, транспорт, промышленность и др. Для всех таких систем можно сказать, что они обладают некоторыми общими проблемами. Центральное место в них занимают устройства логического управления (управляющие сигналы двоичные) и обработки информации, входящие в состав специализированных и универсальных ЭВМ и продолжающие совершенствоваться.

Булевы функции нашли широкое применение в целом ряде различных областей. Сложность используемых на практике функций увеличивается, в связи с усложнением решаемых задач. Увеличивается число используемых переменных, и количество логических операций которое необходимо выполнить для получения значения функции. Но не только функции становятся сложнее, более сложными становятся и ЭВМ, на которых выполняется их вычисление. В связи с совершенствованием ЭВМ, появлением и широким распространением многопроцессорных и многоядерных систем, появилась необходимость в создании алгоритмов и программ, которые смогут использовать новые аппаратные возможности. В алгоритмах, которые дадут существенное преимущество по сравнению с классическими, на новых ЭВМ, при вычислении больших функций.

Алгебра логики имеет большое значение в основаниях математики. Стро-

гое, математически точное построение логических исчислений, решение проблемы дедукции, аксиоматические системы и доказательство теорем. В то же время быстрое развитие вычислительной техники способствует расширению как круга задач, решаемых с помощью алгебры логики, так и методов, применяемых для их решения. Это в первую очередь относится к задачам искусственного интеллекта, решение которых немыслимо без привлечения методов алгебры логики.

Стоит отметить, что на практике множество элементарных логических операций является обязательной частью набора инструкций всех современных микропроцессоров и соответственно входит в языки программирования. Это является одним из важнейших практических приложений методов алгебры логики, изучаемых в современной информатике.

В диссертации рассматриваются алгоритмы вычисления булевой функции в базисах $G_1 = \{\vee, \wedge, \neg\}$ и $G_3 = \{\wedge, \oplus, 1\}$, что позволяет говорить о вычислении произвольной булевой функции, т. к. любая булева функция может быть представлена в виде полинома Жегалкина, т. е. в базисе G_3 .

Особое внимание уделяется анализу представления булевой функции в виде полинома Жегалкина, а точнее анализу показателей качества соответствующих функций. Поиск минимальных оценок исследуемых показателей качества позволяет провести редукцию количества вхождений переменных в булевую функцию, принадлежащую определённому классу симметрических полиномов Жегалкина.

Диссертация состоит из введения, трёх глав, заключения, списка литературы и приложения.

В начале первой главы освещается история развития алгебры логики.

Вторая часть первой главы посвящена математическому аппарату, на который опираются дальнейшие исследования. Даётся представление о базисах, формулах и схемах. В частности описаны оптимизирующие логико-комбинаторные преобразования, включающие в себя следующие механизмы: удаление фиктивных переменных, эквивалентные преобразования булевых функций, их преобразования между базисами. Эти механизмы представляют качественные способы реализации булевых формул для более простого нахождения различных показателей качества. Завершается глава историческим обзором основных проблем минимизации булевых функций и их представлений. Также рассматриваются современные проблемы представления полинома Жегалкина.

Во второй главе рассматриваются методы, позволяющие оценить сложность булевых формул с помощью показателей качества. При этом рассматриваются различные варианты оптимизации формул для того, чтобы получить более точные оценки. Кроме того рассматривается представление булевых формул в классе схем и получение оценок уже для этого класса. В завершении наглядно сравниваются полученные оценки.

В последней главе представлены вопросы автоматизации редукции количества переменных и поиска оценок качества булевых формул. Был подробно приведён алгоритм, на основе которого был построен программный комплекс автоматизированной оценки сложности. А также были отработаны основные особенности поведения алгоритма на функциях с малым числом переменных. В качестве языка программирования, на котором реализованы алгоритмы, был выбран язык C#. Данный выбор обусловлен следующими преимуществами: хорошая поддержка многопоточности, парал-

лельных вычислений, сетевого программирования, высокая скорость выполнения, высокая скорость разработки. В последнем разделе приведены примеры работы программы.

Глава 1

**История возникновения, становления и развития
математической логики, математический аппарат,
решаемые задачи**

1.1 История создания и развития алгебры логики

Логика (др. греч *λογική* — «наука о правильном мышлении», «искусство рассуждения» от *λόγος* — «речь», «рассуждение», «мысль») — это наука правильно рассуждать, наука о формах, методах и законах человеческого мышления.

Главная задача логики состоит в том, чтобы выявить, какие способы рассуждений правильные, а какие нет. Т.е. определить, как прийти к выводу из предпосылок (правильное рассуждение) и получить истинное знание о предмете размышления.

Основоположником логики считают древнегреческого мыслителя Аристотеля (384 - 322 гг. до н. э.). В своём труде «Первая аналитика» он рассматривает правила вывода одних высказываний из других. Именно Аристотель подверг анализу человеческое мышление, такие его формы, как по-

нятие, суждение, умозаключение, и рассмотрел мышление со стороны его строения и структуры. Так возникла формальная логика – наука, изучающая логические операции и правила человеческого мышления, имеющая дело не с содержанием понятий, а только с их формой. Формальная логика занимается выводом нового знания на основе ранее известного, применяя законы и правила мышления.

В течение долгого времени логика оставалась скорее философским инструментом. Но в XVII веке появилась идея разделения логики и естественного языка. Готфрид Вильгельм Лейбниц (1646 - 1716 гг.) в работе «Искусство комбинаторики» предложил использовать математическую символику в логике и при построении логических исчислений, тем самым заложив основы символической логики.

Но логика, разработанная Аристотелем преобладала до середины XIX века, когда началась алгебраизация логики. Идеи применения символического метода к логике впервые высказаны Джорджем Булем (1815 - 1864 гг.) в статье «Математический анализ логики» (1847). В 1854 году он опубликовал трактат «Исследование законов мышления, на которых основываются математические теории логики и вероятности». Независимо от него исследованием логики занимался Огастес де Морган (1806 - 1871 гг.) в своей работе «Формальная логика», опубликованной в 1847 году, он изложил элементы логики высказываний и основы алгебры отношений.

Идеи Буля послужили началу попыток создания логических вычислительных машин. В 1866 году Уильям Стенли Джевонс создал «Логическое фортепиано» — машина, позволяющая механически выводить заключения из поставленных посылок. Также его «Элементарные уроки логики» стали

наиболее читаемым учебником по логике на английском языке.

Дальнейшим развитием алгебры отношений в частности и символической логики в целом занимался Чарльз Сандерс Пирс (1839 - 1914 гг.). Он же в 1880 году в первой главе «Простейшей математики» дал определение «Булиановская алгебра».

Совершенствованием символической логики занимался и Джонн Венн (1834 - 1923 гг.). Важнейшей его работой в данной области является «Символическая логика» (1894). Основной задачей он считал создание языка для записи процесса логического исчисления. В настоящее время его именем названы диаграммы, используемые наглядного отображения логических выражений и решения логических задач.

Идеи Буля получили своё продолжение в трудах Эрнста Шрёдера (1841 - 1902 гг.). Он разработал полную систему аксиом логического исчисления — того, что впоследствии станет булевой алгеброй. Ему же принадлежат термины «логическое исчисление» и «исчисление высказываний». Первый труд «Операционная схема логического исчисления» вышел в 1877 году. Итог своей работы Шрёдер подвёл трёхтомником «Алгебра логики», вышедшем с 1890 по 1905 гг.

В это же время в конце XIX — начале XX веков алгебра логики стала инструментом для аксиоматической формализации различных основополагающих разделов математики, таких как арифметика, геометрия и математический анализ. Работающий в конце 1880-х над аксиоматикой арифметики Джузеппе Пеано (1858 - 1932 гг.) создал удобную систему обозначений, закрепившуюся в современной алгебре логики.

В начале XX века Пауль Эренфест (1880 - 1933 гг.) указал на возмож-

ность применения аппарата булевой алгебры в телефонной связи для описания переключательных цепей. В 1938-1940 г. почти одновременно появились работы советского ученого Шестакова, американского ученого Шеннона и японских ученых Накасимы и Хакадзавы о применении алгебры логики в цифровой технике. Аппарат алгебры логики играет важную роль в развитии современной микропроцессорной техники: он используется в проектировании аппаратных вычислительных средств, в разработке всех языков программирования и в конструировании дискретных устройств автоматики.

1.2 Математический аппарат для синтеза логических формул

1.2.1 Булевые функции и реализующие их формулы

Булевой функцией, или функцией алгебры логики $f(x_1, \dots, x_n)$ ($f^{(n)}$) от n переменных называется функция, аргументы и значения которой определены на множестве $\{0, 1\}$. Для задания такой функции достаточно указать, какое значение функции соответствует каждому из наборов значений переменных функции. Легко подсчитать, что число функций от n переменных соответствует количеству наборов и равняется 2^n . Стандартным способом задания булевых функций является их запись в виде таблицы, где в первых n столбцах выписаны значения аргументов функции, а в последнем — значения функции при соответствующих значениях аргументов.

x_1	\dots	x_{n-1}	x_n	$f(x_1, \dots, x_{n-1}, x_n)$
0	\dots	0	0	$f(0, \dots, 0, 0)$
0	\dots	0	1	$f(0, \dots, 0, 1)$
0	\dots	1	0	$f(0, \dots, 1, 0)$
\dots	\dots			\dots
1	\dots	1	1	$f(1, \dots, 1, 1)$

Определение. Булева функция $f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$ *существенно* зависит от аргумента x_i , если существуют такие значения $\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n$ переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$, что

$$f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) \neq f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n).$$

В этом случае переменная x_i называется *существенной*. Если для на всех наборах значений переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$

$$f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n),$$

то переменная x_i называется *фиктивной* или *несущественной*.

Пусть функция $f(x_1, \dots, x_n)$ несущественно зависит от переменной x_i . После удаления из таблицы истинности всех строк вида $x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n$ и столбца для аргумента x_i , полученная таблица будет определять функцию $g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ такую, что

$$g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n).$$

В таком случае говорится, что функция g получена из функции f *yda-*

удалением фиктивной переменной x_i . Также можно сказать, что функция f получена из функции g *добавлением фиктивной переменной* x_i .

Определение. Булевы функции f и g называются *равными*, если они могут быть получены одна из другой удалением и/или добавлением фиктивных переменных.

Определение. Булева функция $f(x_1, \dots, x_n)$ называется *симметрической* если для любой подстановки $\begin{pmatrix} 1 & \dots & n \\ i_1 & \dots & i_n \end{pmatrix}$ выполняется равенство

$$f(x_1, \dots, x_n) = f(x_{i_1}, \dots, x_{i_n}).$$

Определение. Понятие *формулы, реализующей булеву функцию* задаётся по индукции. Пусть задано некоторое множество функций

$$G = \{f_1(x_1, \dots, x_{n_1}), f_2(x_1, \dots, x_{n_2}), \dots, f_s(x_1, \dots, x_{n_k}), \dots\}$$

1. Каждая функция $f_i(x_1, \dots, x_{n_1})$ из G называется формулой над G .
2. Если A_{j_1}, \dots, A_{j_m} — переменные, или формулы над G , тогда $F = f_i(A_{j_1}, \dots, A_{j_m})$ — *формула над G* . A_{j_1}, \dots, A_{j_m} называются *подформулами* формулы F .

Формулы, реализующие равные функции называются *эквивалентными*.

Функция f , представляемая формулой F называется *суперпозицией* над множеством G . Последовательное применение суперпозиций базисных функций и определяет строение формулы и можно сказать является формулой.

Всего существует $2^{2^2} = 16$ функций от двух переменных. В таблице

приведены вектора значений бинарных функций, и соответствующие им формулы.

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	$x_1 \cdot x_2$	$\overline{x_1} \rightarrow \overline{x_2}$	x_1	$\overline{x_2} \rightarrow x_1$	x_2	$x_1 \oplus x_2$	$x_1 \vee x_2$	$x_1 \downarrow x_2$	$x_1 \sim x_2$	$\overline{x_2}$	$x_2 \rightarrow x_1$	$\overline{x_1}$	$x_1 \rightarrow x_2$	$x_1 \mid x_2$	1	

1.2.2 Полные системы булевых функций и базисы. Показатели качества

Определение. Система булевых функций $G = \{f_1, \dots, f_n\}$ называется *полной*, если любая функция алгебры логики выражается формулой над G .

Определение. Система булевых функций $G = \{f_1, \dots, f_n\}$ называется *базисом*, если она полна, но при удалении любой из функций данной системы, она перестаёт быть полной.

Но в общем случае базисом называют любую полную систему функций. Так базисом называют $G_1 = \{\vee, \wedge, \neg\}$

Функции алгебры логики могут представляться в виде формул над различными базисами. И даже в одном и том же базисе иметь различные представления. В связи с этим возникает проблема сравнения и выбора наиболее подходящей реализации. Для этого вводятся оценки, характеризующие

сложность реализации в виде формулы, удовлетворяющие аксиомам неотрицательности, монотонности, выпуклости и инвариантности [55].

Таких характеристик можно выделить достаточно много. Наиболее часто встречающиеся из них:

- $L_B(F)$ — число символов переменных, встречающихся в записи формулы F .
- $L_K(F)$ — число элементарных конъюнкций, входящих в формулу F .
- $L_F(F)$ — число базисных функций в формуле F .
- $Dep(F)$ — глубина формулы F (максимальный уровень вложенности суперпозиции базисных функций)
- $L_S(f, G)$ — число функциональных (логических) элементов в схеме S ;

Показатель качества реализации функции f в базисе G определяется как минимум из всех реализаций функции в виде формулы в данном базисе:

$$L(f, G) = \min_i (L_i(F_i)).$$

1.2.3 Методы минимизации

Междусобой показатели качества имеют сложные связи. Один из главных показателей — L_B . Уменьшение показателя L_B влечет уменьшение — L_F и L_S . От показателя L_B разным образом зависят остальные из перечисленных выше показателей. При этом минимизация одного показателя

качества может приводить к уменьшению или к возрастанию другого показателя (других). Например, минимизация L_B приводит к минимизации L_F и может приводить к уменьшению, или оставлять без изменения, или увеличивать глубину $Dep(F)$ суперпозиционной формулы F . В таких случаях рекомендуется устанавливать приоритет показателей.

Первым этапом минимизации булевой функции исключаются фиктивные переменные, не влияющие на значения функции. Основной этап минимизации происходит выражением булевой функции через суперпозицию более простых функций. При этом происходит редукция количества вхождений переменных скобочной суперпозиционной формулы.

Определение. Представление булевой функции $y = f(X)$ в виде суперпозиции функций f_1, f_2, \dots, f_k и f_0 , называется её *декомпозицией*, причем

$$y = f_0(X_0 \cup Z), \quad z_1 = f_1(X_1), \dots, z_k = f_k(X_k),$$

где f_0 — внешняя функция суперпозиции, $k \geq 1$, $Z = \{z_1, \dots, z_k\}$ и $X_i \neq \emptyset$, $i = 1, \dots, k$.

Декомпозиция функции преследует две цели: декомпозиция как представление функции в виде суперпозиции базисных функций и декомпозиция как представление функции в виде суперпозиции меньших размерностей.

Одним из способов минимизации может служить проведение эквивалентных преобразований над формулой, и переход к новому базису.

1.2.4 Полиномы Жегалкина

Определение. Элементарной конъюнкцией ранга p называется конъюнкция p переменных вида $x_1^{\sigma_1} \cdot \dots \cdot x_n^{\sigma_n}$. Монотонной элементарной конъюнкцией называется положительно определённая элементарная конъюнкция $x_1 \cdot \dots \cdot x_n$. Константа 1 является элементарной конъюнкцией ранга 0.

Определение. Полиномом Жегалкина или алгебраической нормальной формой называется сумма по модулю 2 попарно различных монотонных элементарных конъюнкций. Пустой полином, т. е. не содержащий ни одной конъюнкции выражается константой 0. Канонический вид полинома Жегалкина:

$$\sum_{i_1, \dots, i_k} \alpha_{i_1 \dots i_k} x_{i_1} \dots x_{i_k},$$

где $\alpha_{i_1 \dots i_k} \in \{0, 1\}$ — двоичный коэффициент разложения в полином Жегалкина. Максимальный ранг элементарной конъюнкции с неравным нулю коэффициентом называется степенью полинома.

Теорема Жегалкина. Каждая булева функция может быть выражена при помощи полинома по модулю 2 (полинома Жегалкина) и притом единственным образом.

Пусть произвольная булева функция $f^{(n)}$ задается полиномом Жегалкина $F^{(n)} = K_1 \oplus \dots \oplus K_i \oplus \dots \oplus K_m$, $i = \overline{1, m}$, где K_i — монотонная элементарная конъюнкция ранга r_i , причем $r_1 \geq \dots \geq r_i \geq \dots \geq r_m$, $i = \overline{1, m}$. В каноническом полиноме Жегалкина его элементарные конъюнкции расположены в соответствии с порядком “ \geq ”, определенным для их рангов. Вектор $\mathbf{r} = (r_1, \dots, r_i, \dots, r_m)$ задает строение полинома Жегалкина $F^{(n)}$. Также для полинома определяется вектор $\mathbf{p} = (p_1, \dots, p_j, \dots, p_n)$ повторя-

емости переменных из множества X в формуле $F^{(n)}$, т. е. переменная x_j , $j = \overline{1, n}$, повторяется p_j раз в формуле $F^{(n)}$.

Функциональное уравнение декомпозиции полинома Жегалкина имеет вид

$$F^{(n)} = \left(\left(x_j F_0^{(n-1)} \right) \oplus F_1^{(n-1)} \right), \quad (1.2.1)$$

где нижние индексы 0 и 1 – номера соответствующих остаточных функций, рассматриваемых на одном множестве $X' = X \setminus \{x_j\}$.

1.3 Обзор проблемы нахождения минимальных оценок

Минимизация формул, реализующих функции, по тем или иным показателям качества диктуется необходимостью совершенствования аппарата алгебры логики, находящего практическое применение в таких областях, как схемотехника, теория управления, криптография и других.

Вопросы конструирования булевых формул, минимальных по своим показателям возникли одновременно с началом применения логических функций для проектирования вычислительных и управляющих устройств. Трудоемкость получения оптимального по какому-либо показателю решения является является сложностью алгоритма переборного характера [2, 3, 4, 32, 33, 56, 57]. Это привело к отказу от стандартных подходов постановки задачи и ее решения. Основным стал асимптотический подход, основывающийся на функции Шеннона [53]. Используя его, О.Б. Лупанов [32, 33] первым разработал метод и получил асимптотическую оценку числа элементов схемы, реализующей булеву функцию.

В работах Ю. И. Журавлева, С. В. Яблонского и Д. А. Поспелова предложено ограничить трудоемкость синтеза и искать алгоритмы, приводящие к схемам наилучшего качества в ограниченном классе. В этих же работах показано, что в классе алгоритмов, заметно отличающихся по трудоемкости от переборных, далеко не всегда удается получить решение, сколько-нибудь близкое к оптимальному [26, 27, 56].

Интересны и вопросы как и насколько может измениться сложность при переходе из одного базиса в другой [2, 3]. Приводятся примеры, когда одна функция в некотором базисе имеет меньшие значения показателей сложности по сравнению с другим базисом, другая же функция, наоборот.

Как было показано в дальнейшем, поиск оптимального решения стоит осуществлять в отдельных счётных классах булевых функций [7, 8].

Тем не менее поиски способа уменьшения асимптотической сложности продолжаются до сих пор. Так в работах [58, 59, 60] оценивается мультиплексивная сложность функции для базиса Жегалкина. До сих пор интересны вопросы связи между различными функционалами качества [61]. Также описываются различные способы применения полиномов Жегалкина для построения логических устройств [40, 62]

1.4 Обзор современных проблем

1.4.1 Представление полинома Жегалкина в виде гиперграфа

Гиперграфом H называется пара $H = (V, E)$, где V — множество элементов, называемых узлами или вершинами, а E — множество непустых

подмножеств V , называемых гиперребрами или просто ребрами гиперграфа [5, 29].

В то время как ребра графа являются парами вершин, гиперребра — произвольными наборами вершин и поэтому могут содержать произвольное количество этих самых вершин [9].

Тем не менее, часто желательно изучать гиперграфы, где все гиперребра имеют одинаковую мощность. Такие гиперграфы называются комплексами [9].

Рассмотрим представление полинома Жегалкина, в виде гиперграфа.

Пусть дана произвольная булева функция, задаваемая полиномом Жегалкина $F^{(n)} = K_1 \oplus \dots \oplus K_i \oplus \dots \oplus K_m$, $i = \overline{1, m}$, где K_i — монотонная элементарная конъюнкция (\exists К) ранга r_i , причем $r_1 \geq \dots \geq r_i \geq \dots \geq r_m$, $i = \overline{1, m}$. Тогда можно провести следующие связи:

1. вершины v_i гиперграфа H соответствуют переменным x_i булевой функции, где $i = \overline{1, n}$;
2. гиперребра e_j гиперграфа H соответствуют конъюнкциям K_j булевой функции, где $i = \overline{1, m}$.

Отдельно стоит рассмотреть конъюнкцию с нулевым рангом, то есть тождественную 1. В этом случае гиперребро e_\emptyset должно быть пустым, что не противоречит понятию гиперграфа, так как гиперребро может содержать **любое** количество вершин.

Для нахождения значения булевой функции нужно ввести понятия булевого значения вершины и булевого веса гиперребра.

Булевое значение вершины $bd(v_i)$ задаётся следующим образом:

$$bw(v_i) = \begin{cases} 0, & x_i \text{ — ложь;} \\ 1, & x_i \text{ — истина.} \end{cases}$$

Булевый вес гиперребра $bw(e_j)$ можно задать, как произведение булевых значений, входящих в гиперребро вершин:

$$bw(e_j) = \prod_k bd(v_k), \quad \forall v_k \in e_j.$$

Для пустого гиперребра булевый вес $bw(e_\emptyset)$ положим равным 1.

Значение функции $F^{(n)}$ можно определить следующим образом:

$$F^{(n)} = \begin{cases} \text{ложь,} & \left(\sum_{j=1}^m e_j\right) \bmod 2 = 0; \\ \text{истина,} & \left(\sum_{j=1}^m e_j\right) \bmod 2 = 1; \end{cases}$$

Отдельно стоит отметить представление в виде гиперграфов симметричных полиномов Жегалкина. Они отличаются от обычных полиномов Жегалкина, тем что все конъюнкции внутри полинома имеют одинаковый ранг. Таким образом гиперграф в виде которого симметричный полином Жегалкина можно представить содержит гиперребра с одной и той же раз мерностью. Отсюда следует, что такой гиперграф является k -комплексом, где k на единицу меньше ранга конъюнкции полинома Жегалкина. Следовательно мы можем рассмотреть представление полинома Жегалкина в более узкий класс гиперграфов — k -комплексы.

Определим кратко k -комплексы:

Для множества вершин $U(n)$, где $n \geq 2$, и $k \in \mathbb{Z}$, $0 \leq k \leq n - 1$, через

$$S^{k+1}(n) = \left\{ \{u_{i_1}, \dots, u_{i_{k+1}}\} : u_{i_j} \in U(n), u_{i_p} \neq u_{i_q} \text{ при } p \neq q \right\}$$

обозначим множество всех $(k + 1)$ -элементных подмножеств из $U(n)$. Пара множеств $\{U(n), S^{k+1}\}$, где $S^{k+1} \subseteq S^{k+1}(n)$, которую обозначим

$$G^k = G^k(U(n), S^{k+1} = S^{k+1}(G^k)),$$

называется k -комплексом [35].

Мощность множества M обозначим $|M|$. Очевидно, что

$$|S^{k+1}(n)| = C_n^{k+1}.$$

Также легко заметить, что:

a) Для 0-комплекса $G^0 = G^0(U(n), S^1)$ имеет место равенство $S^1 = S^1(n)$.

б) Пусть $\Gamma^k(n) = \{G^k(U(n), S^{k+1})\}$ — множество всех k -комплексов с вершинами $U(n)$. Применяя терминологию теории графов, будем полагать, что каждый 0-комплекс — это *вполне несвязный* k -комплекс ($n \geq k + 1$) для любого $k \geq 1$: $G^0(U(n), S^1) \in \Gamma^k(n)$ (будем полагать, что $G^0(U(n), S^1) = G^k(U(n), \emptyset)$ при $k \geq 1$).

В топологии элементы из $S^{k+1}(n)$ называются k -мерными симплексами [1]. В частности, вершины из $U(n) = S^1(n)$ называются 0-мерными симплексами. Элементы из $S^2(n)$ в теории графов называются рёбрами.

Для k -комплекса $G^k(U(n), S^{k+1})$ вершины u_{i_1}, \dots, u_{i_m} , где $2 \leq m \leq k +$

1, называются *смежными*, если они принадлежат некоторому симплексу из S^{k+1} . Симплекс $\{u_{i_1}, \dots, u_{i_{k+1}}\}$ из S^{k+1} называется *инцидентным* каждой вершине u_{i_j} , $1 \leq j \leq k + 1$, а каждая из вершин u_{i_j} , $1 \leq j \leq k + 1$, называется *инцидентной* симплексу $\{u_{i_1}, \dots, u_{i_{k+1}}\}$ [37].

Теперь возьмём произвольный k -мерный симплекс $\{u_{i_1}, \dots, u_{i_{k+1}}\}$ из S^{k+1} и $m \in \mathbb{Z}$, $1 \leq m \leq k$. Любое m -элементное подмножество вершин $\{u_{i_1}, \dots, u_{i_m}\}$ из множества $\{u_{i_1}, \dots, u_{i_{k+1}}\}$ является $(m - 1)$ -мерным симплексом. В топологии симплекс $\{u_{i_1}, \dots, u_{i_m}\}$ называется гранью симплекса $\{u_{i_1}, \dots, u_{i_{k+1}}\}$. Для таких комплексов верно следующее:

a) Множество k -комплексов $\Gamma^k(n) = \{G^k(U(n), S^{k+1})\}$ есть подмножество всех k -мерных комплексов с множеством вершин $U(n)$: в отличие от произвольного k -мерного комплекса любой k -комплекс G^k обладает дополнительным свойством: каждый m -мерный симплекс, где $1 \leq m \leq k$, этого комплекса есть грань некоторого k -мерного симплекса k -комплекса G^k (такие комплексы в топологии относятся к классу *полных* комплексов [1]).

Отметим, что определение k -комплекса допускает существование в k -комплексе изолированных вершин (то есть k -комплекс может иметь 0-мерные симплексы, не являющиеся гранями симплексов большей размерности).

б) Любой k -комплекс G^k , не содержащий изолированные вершины, называется локально k -мерным комплексом [1].

Следует отметить, что в случае, например, симметрического полинома Жегалкина третьей степени мы можем представить его в виде полного 2-комплекса [21].

Таким же образом можно приставить симметрический полином Жегалкина второй степени в виде полного ненаправленного графа.

Глава 2

Оценки качества булевых функций на основе алгебраической декомпозиции

Математическое моделирование в научном и техническом мире все больше становится компьютерным моделированием с постоянно расширяющимися интеллектуальными свойствами технических (аппаратных) средств. При этом сохраняется потребность в совершенствовании математического аппарата (различных разделов дискретной математики). Поэтому в данной главе отрабатывается техника получения различных оценок показателей сложности на основе рекуррентных соотношений (функциональных уравнений).

2.1 Аналитическое определение оценок сложности.

Применим метод функциональных уравнений для реализации булевых функций $f^{(n)}$ в базисе G_3 суперпозиционной формулой $F_{\text{с.ф.}}^{(n)}$ и получения при этом верхней оценки сложности L_F .

Каждое применение функционального уравнения (1.2.1) порождает не более двух базисных функций из G_3 и не более двух остаточных функций

$\left(F_0^{(n-1)} \text{ и } F_1^{(n-1)}\right)$. Используя последовательно уравнение (1.2.1) и суммируя значения сложности L_F , выводим верхнюю оценку показателя сложности L_F для общего случая.

Таким образом, при реализации произвольной булевой функции $f^{(n)} \in P_2^{(n)}$ в базисе G_3 [23] верхняя оценка показателя сложности L_F уточнена как

$$L_F(F_{c.\phi.}^{(n)}, G_3) = (5/4) \cdot 2^{n-2} \quad (2.1.1)$$

Погрешность полученной оценки (2.1.1) может быть большой. Тогда как имеются классы симметрических булевых функций, задаваемых элементарными симметрическими полиномами Жегалкина $F_i^{(n)}$, где n — число переменных, i — степень полинома, для которых показатели сложности минимальны или получены другие аналитические оценки. Так, для полиномов $F_1^{(n)}$ (класс линейных функций — “ \oplus ”) и $F_n^{(n)}$ (класс “ $\&$ ”) показатели сложности L_F его минимального представления изучены ранее [45, 46, 47].

Например,

$$L_F^{\min}\left(F_1^{(n)}, G_3\right) = L_F^{\min}\left(F_n^{(n)}, G_3\right) = n - 1,$$

$$L_F(F_2^{(n)}, G_3) = (n^2 + n - 4)/2 \quad L_F(F_3^{(n)}, G_3) = (n^3 - n - 12)/6$$

и другие [22, 45, 46, 47, 48, 49, 50]. При этом, используя теорему об условии полноты системы булевых функций, можно получать оценки L_F сложности представления функций в других базисах.

Рекуррентное соотношение (1.2.1) фактически есть упрощенный алгоритм декомпозиции булевых функций, который будет уточнён в главе 4. Его основные операции: для формулы $F^{(n)}$ получить переменную x_j , максимально повторяющуюся в формуле $F^{(n)}$; преобразовать формулу $F^{(n)}$ к

виду (1.2.1), получая остаточные функции $F_0^{(n-1)}$ и $F_1^{(n-1)}$; записать формулы $F_0^{(n-1)}$ и $F_1^{(n-1)}$ в специальную таблицу, из которой затем выбираем формулы для последующих шагов алгоритма. Каждый шаг, начиная с первого, позволяет исключать одну переменную.

Приведем пример моделирования синтеза формул на основе рекуррентного соотношения (1.2.1).

П р и м е р 1. Функция $f^{(7)}$ задается в базисе G_3 полиномом $F^{(7)} = x_1x_2x_3x_4 \oplus x_1x_2x_5 \oplus x_1x_6 \oplus x_7$, строения $r = (4, 3, 2, 1)$. Представим её в классе скобочных формул, минимизируя показатель сложности L_F .

Выполним эквивалентные преобразования формулы $F^{(7)}$ в соответствии с ФУ (1.2.1).

$$\begin{aligned} F^{(7)} &= x_1x_2x_3x_4 \oplus x_1x_2x_5 \oplus x_1x_6 \oplus x_7 = \\ &= x_1(x_2x_3x_4 \oplus x_2x_5 \oplus x_6) \oplus x_7 = \\ &= x_1(x_2(x_3x_4 \oplus x_5) \oplus x_6) \oplus x_7 = \\ &= x_1(x_2((x_3x_4) \oplus x_5) \oplus x_6) \oplus x_7 = F_{\text{c.ф.}}^{(7)}. \end{aligned}$$

В результате декомпозиции функции $f^{(7)}$ в базисе G_3 получается бесприворная скобочная формула $F_{\text{c.ф.}}^{(7)}$, с минимальной сложностью $L_F^{\min}(F_{\text{c.ф.}}^{(7)}) = 6$.

2.2 Верхние оценки для класса БФ $F_{n-2}^{(n)}$

На основании (1.2.1) подробно рассмотрим нахождение ряда оценок показателей качества для отдельного класса симметрических БФ $F_{n-2}^{(n)}$. Данний класс характеризуется следующим строением: каждая конъюнкция K_i

содержит $n - 2$ переменных.

2.2.1 Показатель качества L_B

Исследуем сложность представления элементарного симметрического полинома Жегалкина $F_{n-2}^{(n)}$ в базисе G_3 . Для этого разложим функцию $F_{n-2}^{(n)}$:

$$F_{n-2}^{(n)} = F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)}.$$

Воспользуемся следующим леммами для нахождения показателя качества $L_B(F_{n-2}^{(n)})$:

Лемма 1. Число элементарных конъюнкций K_i есть

$$C_n^{n-2} = C_n^2 = \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n.$$

Лемма 2. Число букв в любой элементарной конъюнкции есть $(n - 2)$.

Исходя из лемм, получаем, что количество букв для формулы данного класса

$$L_B(F_{n-2}^{(n)}) = (n-2) \left(\frac{1}{2}n^2 - \frac{1}{2}n \right) = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n.$$

Отсюда можем перейти к следующей теореме.

Теорема 1. Для функции $F_{n-2}^{(n)}$ оценка $L_B(F_{n-2}^{(n)})$ не превосходит $\frac{1}{2}n^3 - \frac{3}{2}n^2 + n$.

Обозначим найденное ограничение как

$$L_{B1}(F_{n-2}^{(n)}, G_3) = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n.$$

2.2.2 Показатель качества L_F

Теперь рассмотрим другой показатель качества: количество подформул в формуле

$$F_{n-2}^{(n)} = F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)}.$$

Количество подформул в формуле соответствует количеству знаков действий в соответствующей формуле. Это значит, что для полиномов Жегалкина, имеющих в составе только положительно определённые элементарные конъюнкции оценку $L_F(F_{n-2}^{(n)})$ можно найти следующим образом:

$$\begin{aligned} L_F(F_{n-2}^{(n)}) &= L_B(F_{n-2}^{(n)}) - 1 = (n-2)C_n^{n-2} - 1 = \\ &= \frac{1}{2}(n-2)(n-1)n - 1 = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n - 1. \end{aligned}$$

Отсюда можем получить следующую теорему.

Теорема 2. Для функции $F_{n-2}^{(n)}$ оценка $L_F(F_{n-2}^{(n)})$ не превосходит $\frac{1}{2}n^3 - \frac{3}{2}n^2 + n - 1$.

Обозначим найденное ограничение как

$$L_{F1}(F_{n-2}^{(n)}, G_3) = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n - 1.$$

2.3 Минимизация полученных оценок

Основной целью данной части работы является минимизация оценок показателей качества функции $F_{n-2}^{(n)}$. Для этого воспользуемся методом

функциональных уравнений и методом конечных разностей

2.3.1 Показатель качества $L_B(F_{n-2}^{(n)}, G_3)$

Применим метод функциональных уравнений для улучшения полученной выше оценки $L_B(F_{n-2}^{(n)})$:

$$F_{n-2}^{(n)} = F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)}. \quad (2.3.1)$$

Функциональному уравнению (2.3.1) соответствует следующее функциональное уравнение для показателя качества L_B :

$$L_B(F_{n-2}^{(n)}) = L_B(F_{n-2}^{(n-1)}) + 1 + L_B(F_{n-3}^{(n-1)}).$$

Произведем следующую замену:

$$u_n = L_B(F_{n-2}^{(n)}), \quad u_{n-1} = L_B(F_{n-3}^{(n-1)}).$$

Таким образом, мы получили рекуррентное соотношение

$$u_n - u_{n-1} = L_B(F_{n-2}^{(n-1)}) + 1 \quad (2.3.2)$$

В (2.3.2) $L_B(F_{n-2}^{(n-1)})$ оценим как

$$L_B(F_{n-2}^{(n-1)}) = (n-2)C_{n-1}^{n-2} = (n-2)(n-1) = n^2 - 3n + 2,$$

получая следующее разностное уравнение

$$u_n - u_{n-1} = n^2 - 3n + 3.$$

Начальные условия подсчитываем непосредственно из полинома F_2^4 .

При $n = 4$:

$$F_2^4 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_3x_4. \quad (2.3.3)$$

Преобразуем формулу (2.3.3), получая:

$$F_2^4 = (x_1x_2 \oplus x_3(x_1 \oplus x_2)) \oplus x_4(x_1 \oplus x_2 \oplus x_3). \quad (2.3.4)$$

Считаем количество букв в формуле (2.3.4). Получаем таким образом начальное условие для $n = 4$: $u_4 = L_B(F_2^4, G_3) = 9$.

Полученное разностное уравнение (2.3.2) и начальное условие при $n = 4$ позволяют получить его решение в виде рекуррентного соотношения [6]. Для этого составляем конечные разности первого, второго и последующих порядков до тех пор, пока они не будут нулевыми, то есть не будет получено решение в виде многочлена с неопределёнными коэффициентами

n	4	5	6	7	8
u_n	9	22	43	74	117
Δ_1		13	21	31	43
Δ_2			8	10	12
Δ_3				2	2
Δ_4					0

Так как конечные разности четвёртого порядка равны нулю, то, из свойства конечных разностей, мы предполагаем решение в виде многочлена третьей степени:

$$u_n = a_0 n^3 + a_1 n^2 + a_2 n + a_3. \quad (2.3.5)$$

Полученную систему уравнений будем решать по методу Гаусса.

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 74 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 43 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 22 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 9 \end{array} \right.$$

$$\left\{ \begin{array}{l} 3a_0 = 1 \\ 15a_0 + a_1 = 4 \\ 61a_0 + 9a_1 + a_2 = 13 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 9 \end{array} \right.$$

$$\left\{ \begin{array}{l} 3a_0 = 1 \\ a_1 = -1 \\ 3a_2 = 5 \\ a_3 = -3 \end{array} \right.$$

Из последней системы найдем коэффициенты для многочлена (2.3.5)

$$a_0 = \frac{1}{3},$$

$$a_1 = -1,$$

$$a_2 = \frac{5}{3},$$

$$a_3 = -3.$$

Таким образом мы получили ещё одну оценку для показателя качества

$$L_B(F_{n-2}^{(n)}):$$

$$L_{B2}(F_{n-2}^{(n)}, G_3) = \frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 3.$$

Отсюда получаем следующую уточнение теоремы 1.

Теорема 3. Для функции $F_{n-2}^{(n)}$ оценка $L_B(F_{n-2}^{(n)})$ не превосходит $\frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 3$.

$$\text{Показатель качества } L_{B3}(F_{n-2}^{(n-1)})$$

Проведем ещё одну минимизацию показателя L_B .

$$\begin{aligned} F_{n-2}^{(n)} &= F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)} \\ L_B(F_{n-2}^{(n)}) &= L_B(F_{n-2}^{(n-1)}) + 1 + L_B(F_{n-3}^{(n-1)}) \\ u_n - u_{n-1} &= L_B(F_{n-2}^{(n-1)}) + 1. \end{aligned}$$

Рассмотрим $L_B(F_{n-2}^{(n-1)})$. Улучшим последнее полученное значение показателя $L_B(F_{n-2}^{(n)})$ за счёт получения более точной оценки рекуррентного

соотношения 2.3.2. Запишем функциональное уравнение для $F_{n-2}^{(n-1)}$:

$$F_{n-2}^{(n-1)} = F_{n-2}^{(n-2)} \oplus x_n \cdot F_{n-3}^{(n-2)}.$$

Ему будет соответствовать следующее разностное уравнение:

$$\begin{aligned} L_B(F_{n-2}^{(n-1)}) &= L_B(F_{n-2}^{(n-2)}) + 1 + L_B(F_{n-3}^{(n-2)}), \\ u_n &= L_B(F_{n-2}^{(n-1)}), \quad u_{n-1} = L_B(F_{n-3}^{(n-2)}), \\ L_B(F_{n-2}^{(n-2)}) &= n - 2, \\ u_n - u_{n-1} &= L_B(F_{n-2}^{(n-2)}) + 1 = (n - 2) + 1 = n - 1. \end{aligned}$$

Подсчитаем начальные условия непосредственно из полинома F_2^3 :

$$F_2^3 = F_2^2 \oplus x_3 F_1^2 = x_1 x_2 \oplus x_3(x_1 \oplus x_2).$$

$$u_4 = L_B(F_2^3) = 5$$

Составим сеточную функцию

n	4	5	6	7
u_n	5	9	14	20
Δ_1		4	5	6
Δ_2			1	1
Δ_3				0

Предполагаем решение в виде многочлена второй степени

$$u_n = a_0 n^2 + a_1 n + a_2.$$

$$\begin{cases} 36a_0 + 6a_1 + a_2 = 14 \\ 25a_0 + 5a_1 + a_2 = 9 \\ 16a_0 + 4a_1 + a_2 = 5 \end{cases}$$

$$\begin{cases} 2a_0 = 1 \\ 9a_0 + a_1 = 4 \\ 9a_0 + 3a_1 + a_2 = 5 \end{cases}$$

$$\begin{cases} 2a_0 = 1 \\ 2a_1 = -1 \\ a_2 = -1 \end{cases}$$

$$a_0 = \frac{1}{2},$$

$$a_1 = -\frac{1}{2},$$

$$a_2 = -1.$$

Решив систему уравнений, получаем следующую оценку:

$$L_B(F_{n-2}^{(n-1)}, G_3) = \frac{1}{2}n^2 - \frac{1}{2}n - 1.$$

Возвращаемся к исходному уравнению для показателя качества $L_B(F_{n-2}^{(n)})$. Подставим полученную оценку в уравнение (2.3.2), которое приобретает следующий вид:

$$u_n - u_{n-1} = L_B(F_{n-2}^{(n-1)}) = \frac{1}{2}n^2 - \frac{1}{2}n - 1 + 1 = \frac{1}{2}n(n-1).$$

Для составления сеточной функции воспользуемся уже найденными начальными условиями ($u_4 = 9$):

n	4	5	6	7	8
u_n	9	19	34	55	83
Δ_1		10	15	21	28
Δ_2			5	6	7
Δ_3				1	1
Δ_4					0

Отсюда можно определить общий вид решения:

$$u_n = a_0 n^3 + a_1 n^2 + a_2 n + a_3.$$

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 55 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 34 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 19 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 9 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 30a_0 + 2a_1 = 5 \\ 61a_0 + 9a_1 + a_2 = 10 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 9 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ a_1 = 0 \\ 6a_2 = -1 \\ a_3 = -1 \end{array} \right.$$

$$a_0 = \frac{1}{6},$$

$$a_1 = 0,$$

$$a_2 = -\frac{1}{6},$$

$$a_3 = -1.$$

Таким образом мы получили третью оценку показателя L_B :

$$L_{B3} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{6}n^3 - \frac{1}{6}n - 1,$$

а также получаем ещё одно уточнение теоремы 1.

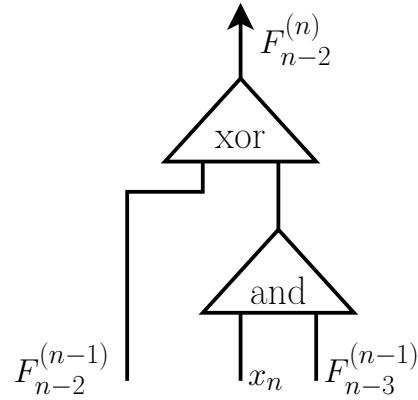
Теорема 4. Для функции $F_{n-2}^{(n)}$ оценка $L_B \left(F_{n-2}^{(n)} \right)$ не превосходит $\frac{1}{6}n^3 - \frac{1}{6}n - 1$.

2.3.2 Показатель качества $L_F \left(F_{n-2}^{(n)}, G_3 \right)$

Показатель качества $L_{F2} \left(F_{n-2}^{(n)} \right)$

$$F_{n-2}^{(n)} = F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)}.$$

Данному функциональному уравнению соответствует следующая схема для n -го элемента:



По данной схеме составляется функциональное уравнение для показателя качества L_F :

$$L_F(F_{n-2}^{(n)}) = L_F(F_{n-2}^{(n-1)}) + 2 + L_F(F_{n-3}^{(n-1)}).$$

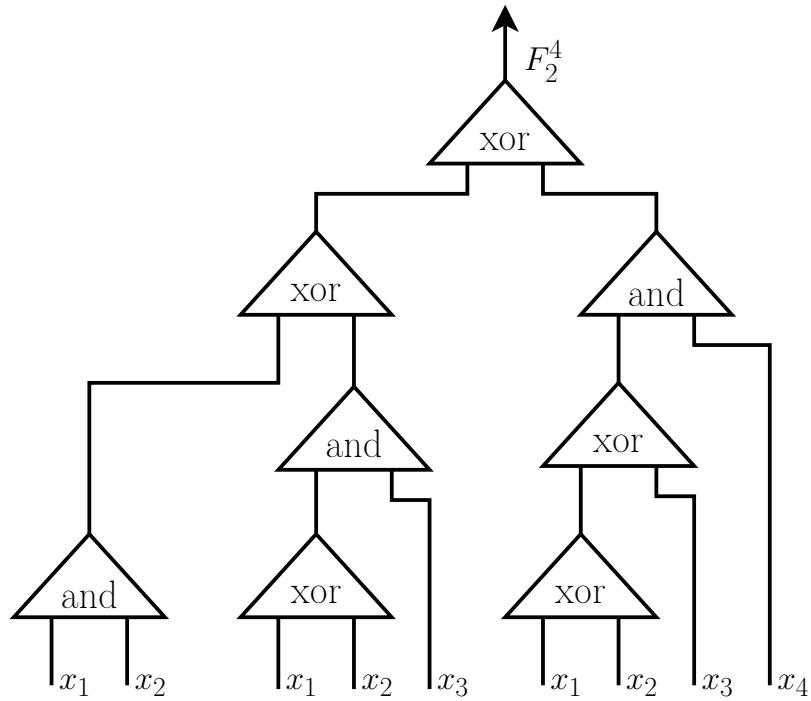
Разностные уравнения получаем так же, как в предыдущем разделе:

$$\begin{aligned} u_n &= L_F(F_{n-2}^{(n)}), & u_{n-1} &= L_F(F_{n-3}^{(n-1)}), \\ u_n &= u_{n-1} + L_F(F_{n-2}^{(n-1)}) + 2, \end{aligned} \quad (2.3.6)$$

$$L_F(F_{n-2}^{(n-1)}) = (n-2)C_{n-1}^{n-2} - 1 = (n-2)(n-1) - 1 = n^2 - 3n + 1.$$

$$u_n - u_{n-1} = n^2 - 3n + 3. \quad (2.3.7)$$

Подсчитаем начальные условия. По полученной суперпозиционной формуле F_2^4 восстанавливаем схему из функциональных элементов без ветвлений.



Количество подформул в формуле соответствует количеству базисных функциональных элементов в схеме. Т. о. для $n = 4$ получаем начальные условия $u_4 = 8$.

На основе разностного уравнения (2.3.7) составим сеточную функцию

n	4	5	6	7	8	9
u_n	8	21	42	73	116	173
Δ_1		13	21	31	43	57
Δ_2			8	10	12	14
Δ_3				2	2	2
Δ_4					0	0

Решение будем искать в следующем виде:

$$u_n = a_0 n^3 + a_1 n^2 + a_2 n + a_3.$$

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 73 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 42 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 21 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 8 \end{array} \right.$$

$$\left\{ \begin{array}{l} 3a_0 = 1 \\ 15a_0 + a_1 = 4 \\ 61a_0 + 9a_1 + a_2 = 13 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 8 \end{array} \right.$$

$$\left\{ \begin{array}{l} 3a_0 = 1 \\ a_1 = -1 \\ 3a_2 = 5 \\ a_3 = -4 \end{array} \right.$$

$$a_0 = \frac{1}{3},$$

$$a_1 = -1,$$

$$a_2 = \frac{5}{3},$$

$$a_3 = -4.$$

В итоге мы нашли оценку для показателя качества L_F :

$$L_{F2} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 4,$$

что позволяет уточнить теорему 1.

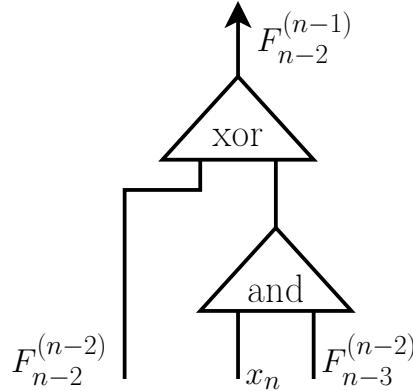
Теорема 5. Для функции $F_{n-2}^{(n)}$ оценка $L_F(F_{n-2}^{(n)})$ не превосходит $\frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 4$.

Показатель качества $L_{F3}(F_{n-2}^{(n)})$

Найдём показатель $L_F(F_{n-2}^{(n-1)})$ для получения более точной оценки разностного уравнения (2.3.6). Для этого разложим $F_{n-2}^{(n-1)}$

$$F_{n-2}^{(n-1)} = F_{n-2}^{(n-2)} \oplus x_n \cdot F_{n-3}^{(n-2)}.$$

На основе данной формулы составим схему:



И функциональное уравнение для $L_F(F_{n-2}^{(n-1)})$:

$$L_F(F_{n-2}^{(n-1)}) = L_F(F_{n-2}^{(n-2)}) + 2 + L_F(F_{n-3}^{(n-2)}).$$

Воспользуемся следующей заменой:

$$u_n = L_F(F_{n-2}^{(n-1)}), \quad u_{n-1} = L_F(F_{n-3}^{(n-2)}).$$

$L_F(F_{n-2}^{(n-2)})$ можно подсчитать непосредственно из формулы. Получаем

простой случай:

$$F_{n-2}^{n-2} = x_1 \cdot x_2 \cdot \dots \cdot x_{n-2}.$$

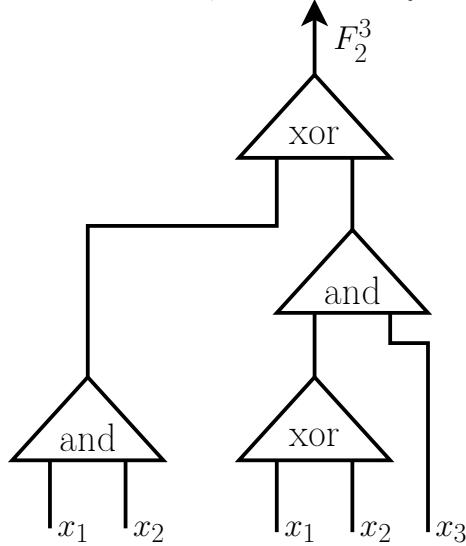
$$L_F(F_{n-1}^{(n-1)}) = n - 3.$$

Т. о. мы получаем разное уравнение

$$u_n - u_{n-1} = L_F(F_{n-2}^{(n-1)}) + 2 = (n - 3) + 2 = n - 1.$$

При $n = 4$ мы получаем полином $F_2^3 = F_2^2 \oplus x_3 F_1^2 = x_1 x_2 \oplus x_3(x_1 \oplus x_2)$.

Построим схему с ветвлением, соответствующую данной формуле:



Из данной схемы подсчитываем начальные условия: $u_4 = 4$. Теперь составим сеточную функцию.

n	4	5	6	7
u_n	4	8	13	19
Δ_1		4	5	6
Δ_2			1	1
Δ_3				0

Т. к. конечные разности имеют третий порядок, то решение будем ис-

кать в виде полинома второй степени.

$$u_n = a_0 n^2 + a_1 n + a_2.$$

$$\begin{cases} 36a_0 + 6a_1 + a_2 = 13 \\ 25a_0 + 5a_1 + a_2 = 8 \\ 16a_0 + 4a_1 + a_2 = 4 \end{cases}$$

$$\begin{cases} 2a_0 = 1 \\ 9a_0 + a_1 = 4 \\ 16a_0 + 4a_1 + a_2 = 4 \end{cases}$$

$$\begin{cases} 2a_0 = 1 \\ 2a_1 = -1 \\ a_2 = -2 \end{cases}$$

$$a_0 = \frac{1}{2},$$

$$a_1 = -\frac{1}{2},$$

$$a_2 = -2.$$

Мы получили оценку для полинома Жегалкина

$$L_F \left(F_{n-2}^{(n-1)}, G_3 \right) = \frac{1}{2}n^2 - \frac{1}{2}n - 2.$$

Возвращаемся к исходному уравнению для показателя качества $L_F(F_{n-2}^{(n)})$. Полученную оценку подставим в разностное уравнение 2.3.6:

$$u_n - u_{n-1} = \frac{1}{2}n^2 - \frac{1}{2}n - 2 + 2 = \frac{1}{2}n(n-1).$$

Используя начальные данные, полученные ранее, составим сеточную функцию.

n	4	5	6	7	8
u_n	8	18	33	54	82
Δ_1		10	15	21	28
Δ_2			5	6	7
Δ_3				1	1
Δ_4					0

$$u_n = a_0 n^3 + a_1 n^2 + a_2 n + a_3.$$

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 54 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 33 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 18 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 8 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 30a_0 + 2a_1 = 5 \\ 61a_0 + 9a_1 + a_2 = 10 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 8 \end{array} \right.$$

$$\left\{ \begin{array}{rcl} 6a_0 & = & 1 \\ a_1 & = & 0 \\ 6a_2 & = & -1 \\ a_3 & = & -2 \end{array} \right.$$

$$a_0 = \frac{1}{6},$$

$$a_1 = 0,$$

$$a_2 = -\frac{1}{6},$$

$$a_3 = -2.$$

Отсюда получаем

$$L_{F3} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{6}n^3 - \frac{1}{6}n - 2.$$

Теперь можем ещё раз уточнить теорему 2.

Теорема 6. Для функции $F_{n-2}^{(n)}$ оценка $L_F \left(F_{n-2}^{(n)} \right)$ не превосходит $\frac{1}{6}n^3 - \frac{1}{6}n - 2$.

2.4 Реализация в классе схем

2.4.1 Показатель качества $L_S \left(F_{n-2}^{(n)}, G_1 \right)$

Оценка L_S

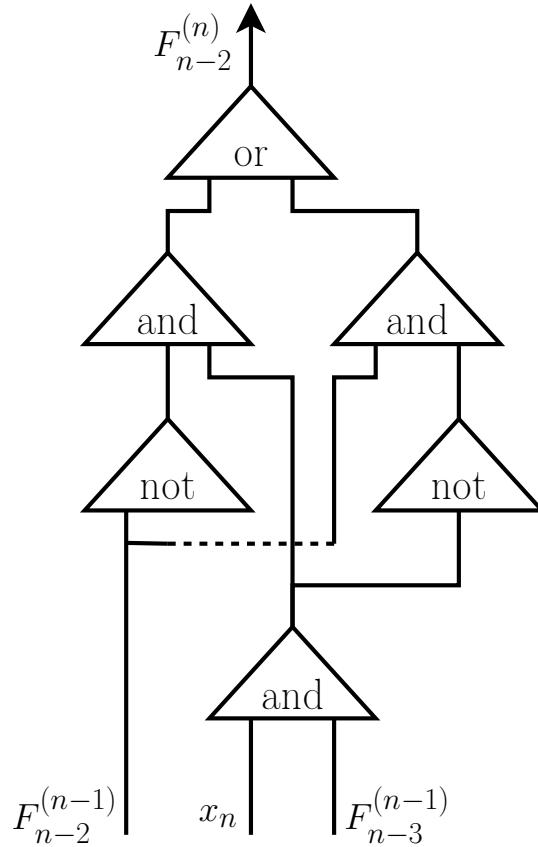
Исследуем сложность представления элементарного симметрического полинома Жегалкина $F_{n-2}^{(n)}$, построив формулу в базисе G_1 , на основе базиса

G_3 :

$$\begin{aligned} F_{n-2}^{(n)} &= F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)} = \\ &= \overline{F_{n-2}^{(n-1)}} x_n F_{n-3}^{(n-1)} \vee F_{n-2}^{(n-1)} \overline{x_n F_{n-3}^{(n-1)}}. \end{aligned} \quad (2.4.1)$$

Получаем реализацию полинома Жегалкина в базисе G_1 для текущей итерации. На данном этапе формулы $F_{n-2}^{(n-1)}$ и $F_{n-3}^{(n-1)}$ реализованы неявно, подразумевается, что они представлены в базисе G_3 , а их композиция — в базисе G_1 . Мы предполагаем, что на каждой следующей итерации получившиеся подформулы переводим в базис G_1 и строим для них соответствующую схему.

Формула (2.4.1) реализуется в виде следующей схемы:



Это общая схема, для произвольной итерации. Т. к. присутствует ветвь-

ление выходов, она частично оптимизирована. При подсчете числа элементов в схеме, мы видим, что в исходной формуле присутствуют одинаковые подформулы. В этом случае мы можем минимизировать число элементов в схеме следующим образом: реализуя некоторую, повторяющуюся далее подформулу, мы выполняем ветвление ее выхода, и, в дальнейшем, ее опускаем.

Из формулы (2.4.1) получаем функциональное уравнение для показателя качества L_S - числа элементов в схеме:

$$L_S(F_{n-2}^{(n)}) = L_S(F_{n-2}^{(n-1)}) + 6 + L_S(F_{n-3}^{(n-1)}).$$

А также разностное уравнение

$$\begin{aligned} u_n &= L_S(F_{n-2}^{(n)}), & u_{n-1} &= L_S(F_{n-3}^{(n-1)}), \\ u_n &= u_{n-1} + L_S(F_{n-2}^{(n-1)}) + 6. \end{aligned} \tag{2.4.2}$$

Нам неизвестен показатель $L_S(F_{n-2}^{(n-1)})$, поэтому найдем его:

$$\begin{aligned} F_{n-2}^{(n-1)} &= F_{n-2}^{(n-2)} \oplus x_{n-1} \cdot F_{n-3}^{(n-2)} = \\ &= \overline{F_{n-2}^{(n-2)}} x_n F_{n-3}^{(n-2)} \vee F_{n-2}^{(n-2)} \overline{x_{n-1} F_{n-3}^{(n-2)}}; \end{aligned}$$

$$\begin{aligned} L_S(F_{n-2}^{(n-1)}) &= L_S(F_{n-2}^{(n-2)}) + 6 + L_S(F_{n-3}^{(n-2)}), \\ u_n &= L_S(F_{n-2}^{(n-1)}), & u_{n-1} &= L_S(F_{n-3}^{(n-2)}). \end{aligned}$$

Число элементов в схеме соответствует числу символов операций в фор-

мудле:

$$F_{n-2}^{(n-2)} = x_1 \cdot x_2 \cdot \dots \cdot x_{n-2}.$$

Очевидно, что $L_S(F_{n-2}^{(n-2)}) = n - 3$, получаем следующее разностное уравнение:

$$u_n - u_{n-1} = (n - 3) + 6 = n + 3. \quad (2.4.3)$$

Подсчитаем начальные условия для $n = 2$:

$$F_1^{(2)} = x_1 \oplus x_2 = \overline{x_1}x_2 \vee x_1\overline{x_2},$$

$$u_2 = L_S(F_1^{(2)}) = 5.$$

На основе разностного уравнения (2.4.3) получаем для показателя качества $L_S(F_{n-1}^{(n)})$ сеточную функцию, представленную в таблице:

n	3	4	5	6
u_n	5	12	20	29
Δ_1		7	8	9
Δ_2			1	1
Δ_3				0

Будем искать решение в виде: $u_n = a_0n^2 + a_1n + a_2$.

$$\begin{cases} 25a_0 + 5a_1 + a_2 = 20 \\ 16a_0 + 4a_1 + a_2 = 12 \\ 9a_0 + 3a_1 + a_2 = 5 \end{cases}$$

$$\begin{cases} 2a_0 &= 1 \\ 7a_0 + a_1 &= 7 \\ 9a_0 + 3a_1 + a_2 &= 5 \end{cases}$$

$$\begin{cases} 2a_0 &= 1 \\ 2a_1 &= 7 \\ a_2 &= -10 \end{cases}$$

$$a_0 = \frac{1}{2},$$

$$a_1 = \frac{7}{2},$$

$$a_2 = -10.$$

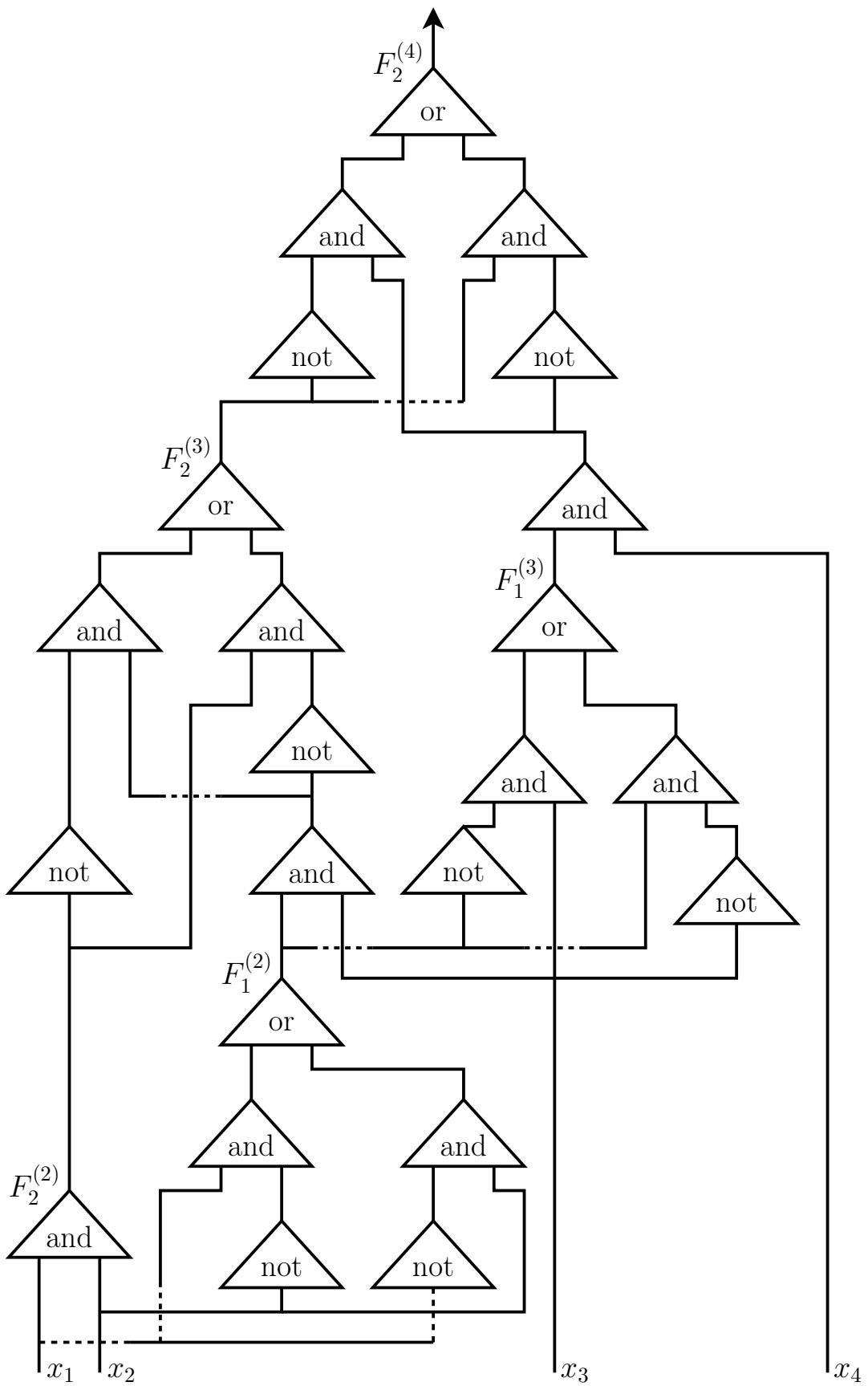
Решив систему уравнений, получим значение для показателя качества

$$L_S \left(F_{n-2}^{(n-1)}, G_1 \right) = \frac{1}{2}n^2 + \frac{7}{2}n - 10.$$

Подставив получившееся значение в формулу (2.4.2), выведем следующее разностное уравнение:

$$u_n - u_{n-1} = \frac{1}{2}n^2 + \frac{7}{2}n - 10 + 6 = \frac{1}{2}n^2 + \frac{7}{2}n - 4. \quad (2.4.4)$$

Подсчитаем начальные условия для $n = 4$, на основе соответствующей схемы:



На основе начальных данных $u_4 = 23$, и разностной схемы (2.4.4), со-

ставим таблицу:

n	4	5	6	7	8
u_n	23	49	84	129	185
Δ_1		26	35	46	56
Δ_2			9	10	11
Δ_3				1	1
Δ_4					0

Решение будем искать в виде: $u_n = a_0n^3 + a_1n^2 + a_2n + a_3$.

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 120 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 79 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 47 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 23 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 15a_0 + a_1 = 4 \\ 61a_0 + 9a_1 + a_2 = 24 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 23 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 2a_1 = 3 \\ 3a_2 = 1 \\ a_3 = -13 \end{array} \right.$$

$$a_0 = \frac{1}{6},$$

$$a_1 = 2,$$

$$a_2 = -\frac{13}{6},$$

$$a_3 = -11.$$

Решив систему уравнений, получаем:

$$L_{S1} \left(F_{n-2}^{(n)}, G_1 \right) = \frac{1}{6}n^3 + 2n^2 - \frac{13}{6}n - 11.$$

Соответственно, приходим к теореме.

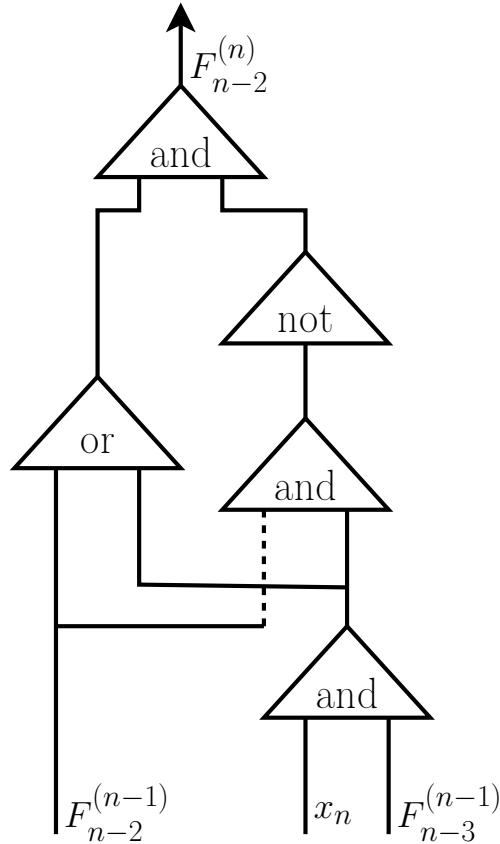
Теорема 7. Для функции $F_{n-2}^{(n)}$ оценка $L_S \left(F_{n-2}^{(n)} \right)$ не превосходит $\frac{1}{6}n^3 + 2n^2 - \frac{13}{6}n - 11$.

Улучшение оценки L_S

Для улучшения оценки воспользуемся ещё одним способом представления операции \oplus в базисе G_1 : $x_1 \oplus x_2 = x_1x_2(x_1 \vee x_2)$. Данная формула содержит меньшее количество знаков действий, а значит соответствующая схема будет иметь меньшее количество функциональных элементов, что позволит улучшить оценку.

$$\begin{aligned} F_{n-2}^{(n)} &= F_{n-2}^{(n-1)} \oplus x_n \cdot F_{n-3}^{(n-1)} = \\ &= \overline{F_{n-2}^{(n-1)} x_n F_{n-3}^{(n-1)}} \left(F_{n-2}^{(n-1)} \vee x_n F_{n-3}^{(n-1)} \right). \end{aligned}$$

Данная формула представима в виде следующей схемы:



На основе схемы строим функциональное уравнение для показателя качества L_S :

$$L_S(F_{n-2}^{(n)}) = L_S(F_{n-2}^{(n-1)}) + 5 + L_S(F_{n-3}^{(n-1)}),$$

$$u_n = L_S(F_{n-2}^{(n)}), \quad L_S(F_{n-3}^{(n-1)}).$$

Произведя замену, мы получили разностное уравнение

$$u_n = u_{n-1} + L_S(F_{n-2}^{(n-1)}) + 5. \quad (2.4.5)$$

Для вычисления данного уравнения необходимо получить оценку поли-

нома $F_{n-2}^{(n-1)}$:

$$\begin{aligned} F_{n-2}^{(n-1)} &= F_{n-2}^{(n-2)} \oplus x_{n-1} \cdot F_{n-3}^{(n-2)} = \\ &= \overline{F_{n-2}^{(n-2)} x_{n-1} F_{n-3}^{(n-2)}} \left(F_{n-2}^{(n-2)} \vee x_{n-1} F_{n-3}^{(n-2)} \right). \end{aligned}$$

Составим функциональное уравнение для показателя качества L_S :

$$\begin{aligned} L_S(F_{n-2}^{(n-1)}) &= L_S(F_{n-2}^{(n-2)}) + 5 + L_S(F_{n-3}^{(n-2)}), \\ u_n &= L_S(F_{n-2}^{(n-1)}), \quad L_S(F_{n-3}^{(n-2)}). \end{aligned}$$

$$F_{n-2}^{(n-2)} = x_1 \cdot x_2 \cdot \dots \cdot x_{n-2}. \quad (2.4.6)$$

Т. к. формуле (2.4.6) соответствует схема без ветвления, то количество элементов в схеме равно количеству символов операций в формуле, т. е.

$$L_S(F_{n-1}^{(n-1)}) = n - 3.$$

Строим разностное уравнение:

$$u_n - u_{n-1} = (n - 3) + 5 = n + 2.$$

Подсчитаем начальные условия, представив полином $F_1^{(2)}$ следующим образом

$$F_1^{(2)} = x_1 \oplus x_2 = \overline{x_1 x_2}(x_1 \vee x_2).$$

Мы получили $u_3 = L_S(F_1^{(2)}) = 4$.

По полученным данным строим сеточную функцию

n	3	4	5	6
u_n	4	10	17	25
Δ_1		6	7	8
Δ_2			1	1
Δ_3				0

Решение ищем в виде многочлена $u_n = a_0n^2 + a_1n + a_2$, по которому составляем систему уравнений.

$$\left\{ \begin{array}{l} 25a_0 + 5a_1 + a_2 = 17 \\ 16a_0 + 4a_1 + a_2 = 10 \\ 9a_0 + 3a_1 + a_2 = 4 \end{array} \right.$$

$$\left\{ \begin{array}{l} 2a_0 = 1 \\ 7a_0 + a_1 = 6 \\ 9a_0 + 3a_1 + a_2 = 4 \end{array} \right.$$

$$\left\{ \begin{array}{l} 2a_0 = 1 \\ 2a_1 = 5 \\ a_2 = -8 \end{array} \right.$$

Из полученной системы уравнений посчитаем коэффициенты

$$a_0 = \frac{1}{2},$$

$$a_1 = \frac{5}{2},$$

$$a_2 = -8.$$

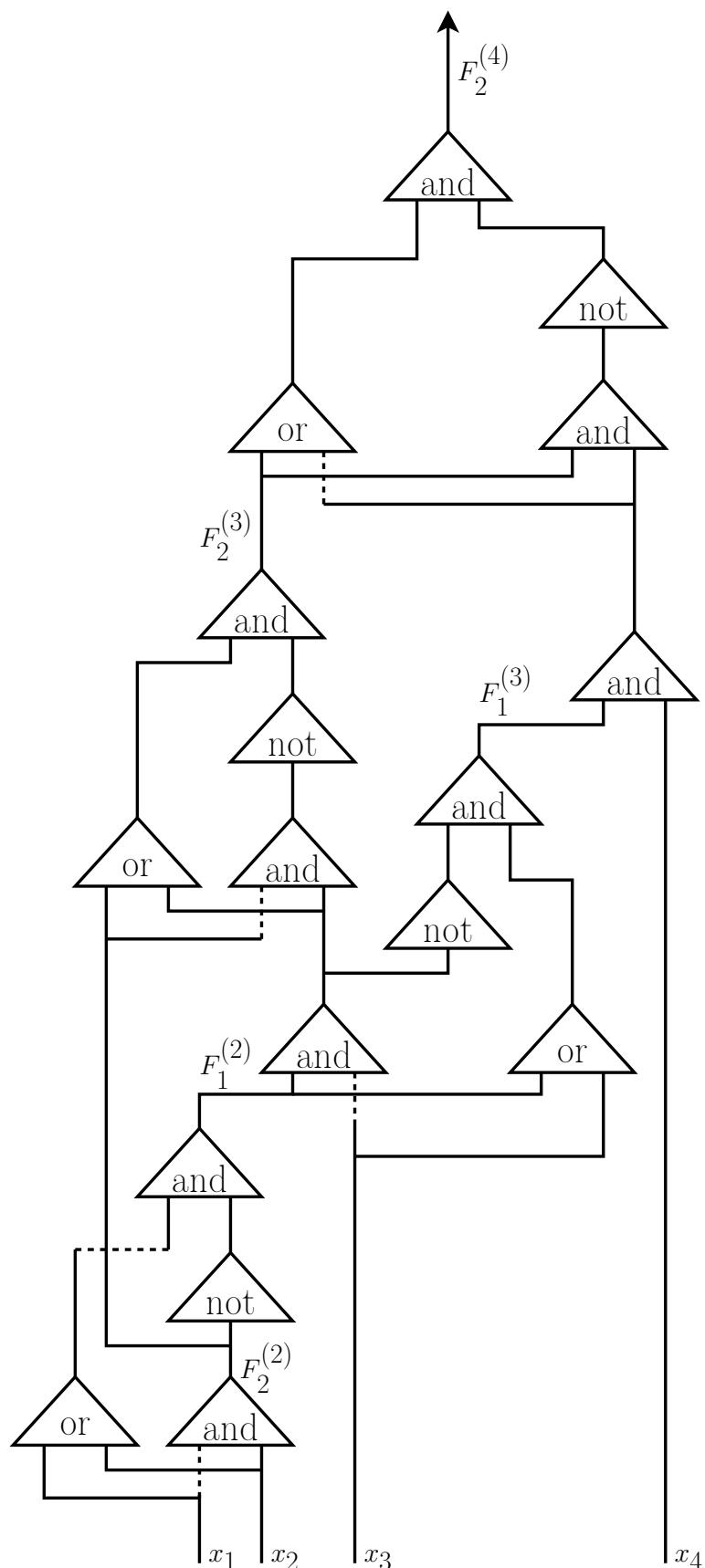
Мы получили оценку для полинома:

$$L_S \left(F_{n-2}^{(n-1)}, G_1 \right) = \frac{1}{2}n^2 + \frac{5}{2}n - 8.$$

Подставим её в разностное уравнение (2.4.5)

$$u_n - u_{n-1} = \frac{1}{2}n^2 + \frac{5}{2}n - 8 + 5 = \frac{1}{2}n^2 + \frac{5}{2}n - 3.$$

Подчитаем начальные условия, для чего составим схему полинома $F_2^{(4)}$,
с учётом ветвления выходов.



Из схемы получаем $u_4 = 17$.

По начальным данным и разностному уравнению составим сеточную

функцию:

n	4	5	6	7	8
u_n	17	39	69	108	157
Δ_1		22	30	39	49
Δ_2			8	9	10
Δ_3				1	1
Δ_4					0

Будем искать решение в виде полинома третьей степени:

$$u_n = a_0 n^3 + a_1 n^2 + a_2 n + a_3.$$

$$\left\{ \begin{array}{l} 343a_0 + 49a_1 + 7a_2 + a_3 = 108 \\ 216a_0 + 36a_1 + 6a_2 + a_3 = 69 \\ 125a_0 + 25a_1 + 5a_2 + a_3 = 39 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 17 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 15a_0 + a_1 = 4 \\ 61a_0 + 9a_1 + a_2 = 22 \\ 64a_0 + 16a_1 + 4a_2 + a_3 = 17 \end{array} \right.$$

$$\left\{ \begin{array}{l} 6a_0 = 1 \\ 2a_1 = 3 \\ 3a_2 = -5 \\ a_3 = -11 \end{array} \right.$$

$$a_0 = \frac{1}{6},$$

$$a_1 = \frac{3}{2},$$

$$a_2 = -\frac{5}{3},$$

$$a_3 = -11.$$

Решив систему и получив коэффициенты, мы нашли следующую оценку для показателя качества L_S :

$$L_{S2} \left(F_{n-2}^{(n)}, G_1 \right) = \frac{1}{6}n^3 + \frac{3}{2}n^2 - \frac{5}{3}n - 11.$$

Таким образом получаем уточнение теоремы 7.

Теорема 8. Для функции $F_{n-2}^{(n)}$ оценка $L_S \left(F_{n-2}^{(n)} \right)$ не превосходит $\frac{1}{6}n^3 + \frac{3}{2}n^2 - \frac{5}{3}n - 11$.

2.5 Сравнение результатов

2.5.1 Сравнение оценок L_B

Сравним найденные оценки показателя L_B :

$$L_{B1} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n,$$

$$L_{B2} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 3,$$

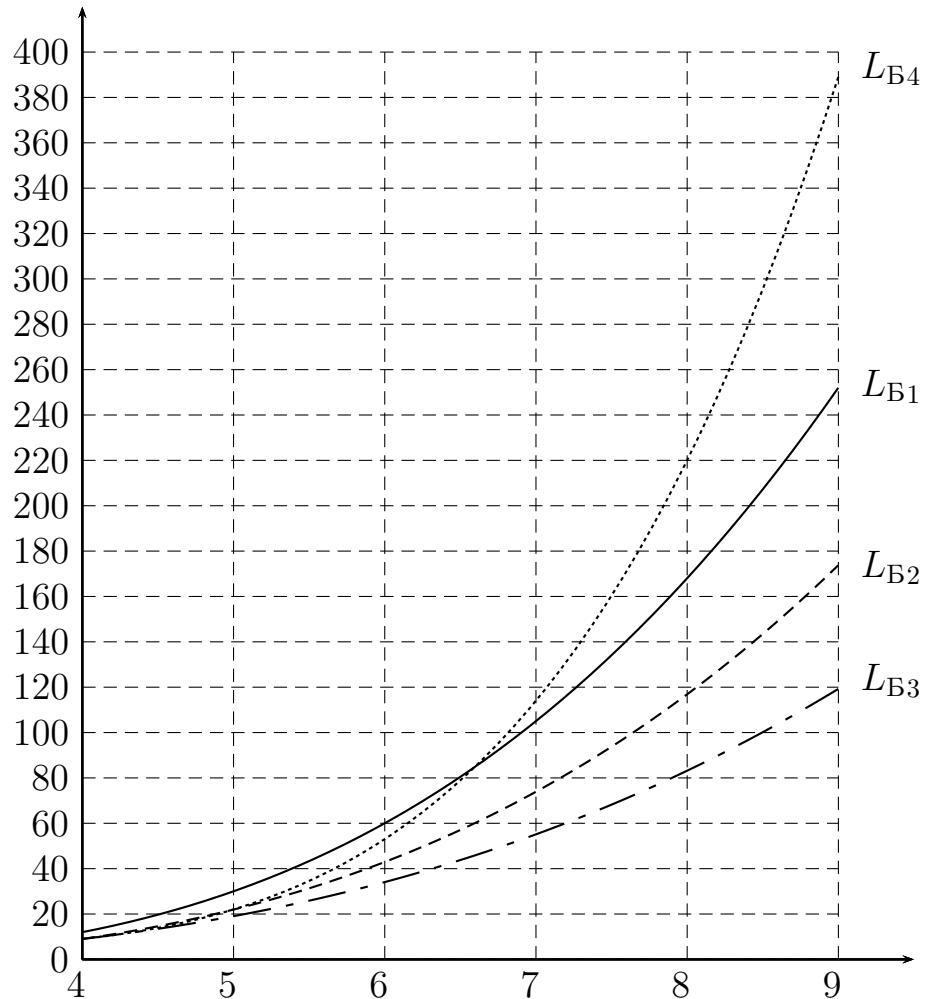
$$L_{B3} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{6}n^3 - \frac{1}{6}n - 1$$

и оценку, полученную в работе [16] —

$$L_{B4} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{8}n^4 - \frac{3}{4}n^3 + \frac{11}{8}n^2 + \frac{1}{4}n + 2.$$

Как видно, оценки L_{B1} , L_{B2} и L_{B3} , полученные в данной работе, на порядок меньше предыдущей оценки (L_{B4}). Отразим полученные оценки на графике.

Легко заметить, что удалось значительно уменьшить показатель качества $L_B \left(F_{n-2}^{(n)} \right)$ (и это проверяется аналитически). Минимальная, из полученных оценок: $L_{B3} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{6}n^3 - \frac{1}{6}n - 1$.



Сравнение оценок L_F

Сравним все полученные оценки:

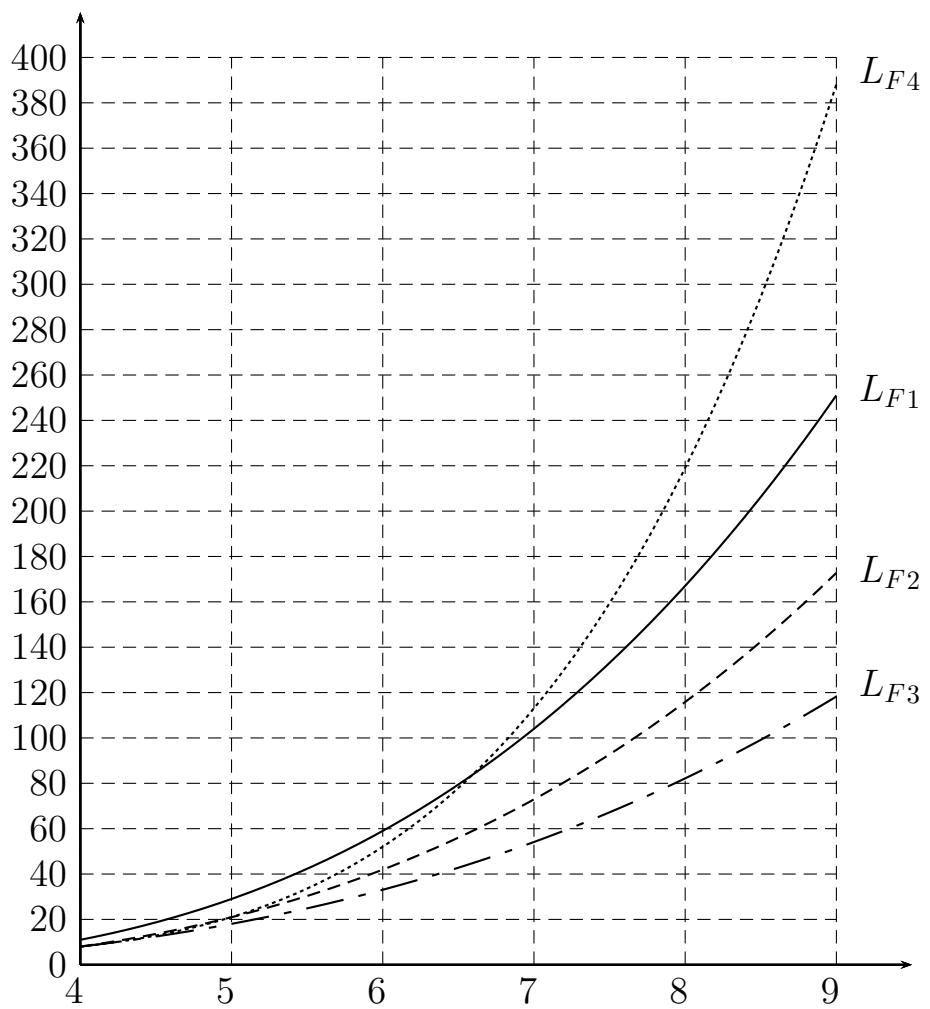
$$L_{F1} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{2}n^3 - \frac{3}{2}n^2 + n - 1,$$

$$L_{F2} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{3}n^3 - n^2 + \frac{5}{3}n - 4,$$

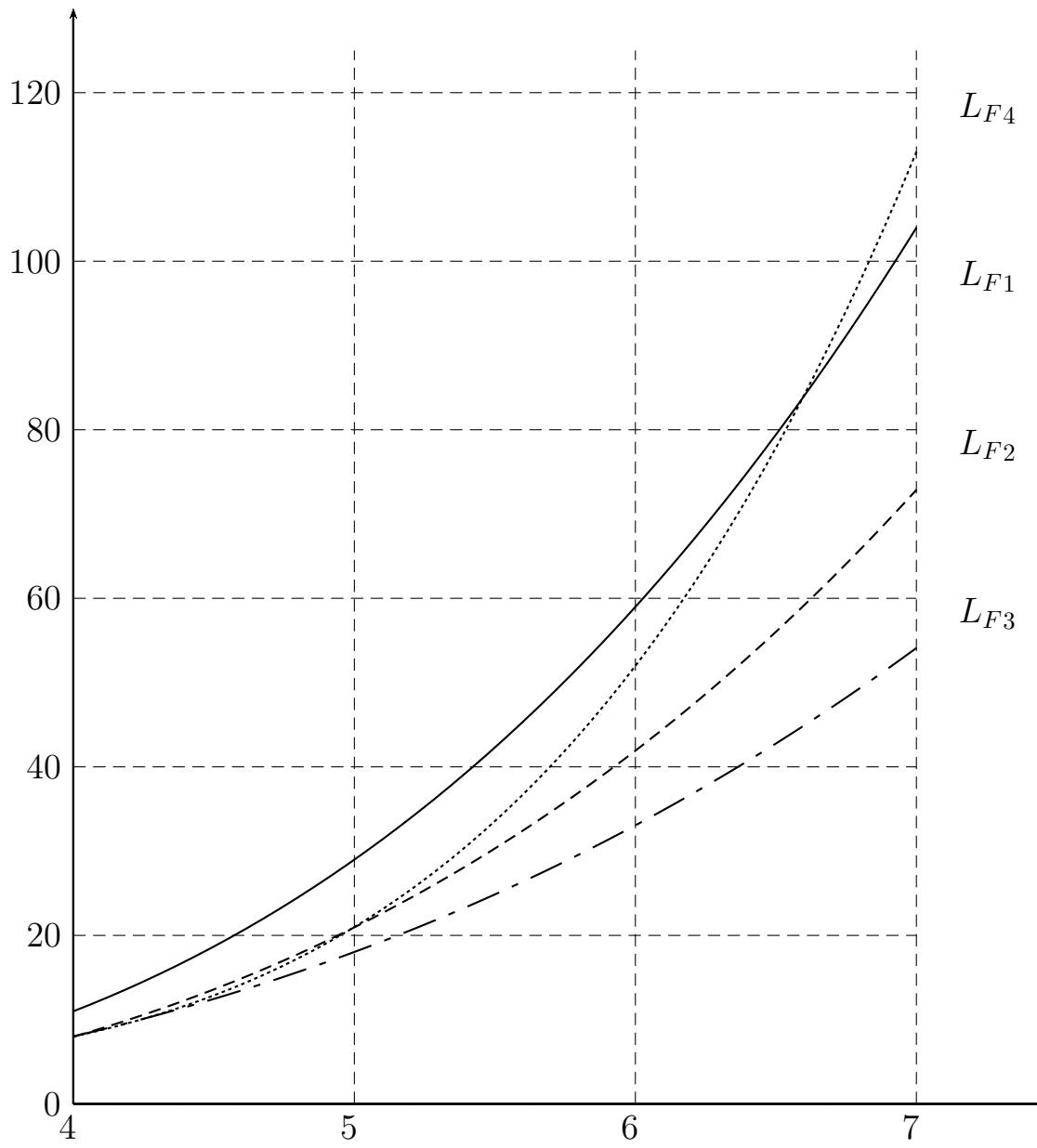
$$L_{F3} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{6}n^3 - \frac{1}{6}n - 2$$

и оценку, полученную в работе [16]

$$L_{F4} \left(F_{n-2}^{(n)}, G_3 \right) = \frac{1}{8}n^4 - \frac{3}{4}n^3 + \frac{11}{8}n^2 + \frac{1}{4}n + 1.$$



Для $n = 4$ оценки L_{F2} , L_{F3} и L_{F4} одинаковы и меньше оценки L_{F1} , т.к. была проведена оптимизация начальных условий. Рассмотрим кусок графика с увеличением:



Для $n = 5$ оценки L_{F2} и L_{F4} совпадают. Минимальной, в данной точке, является оценка L_{F3} , максимальной — L_{F1} . Для $n = 6$ — $L_{F3} < L_{F2} < L_{F4} < L_{F1}$. Для $n \geq 7$ оценка L_{F4} становится наибольшей.

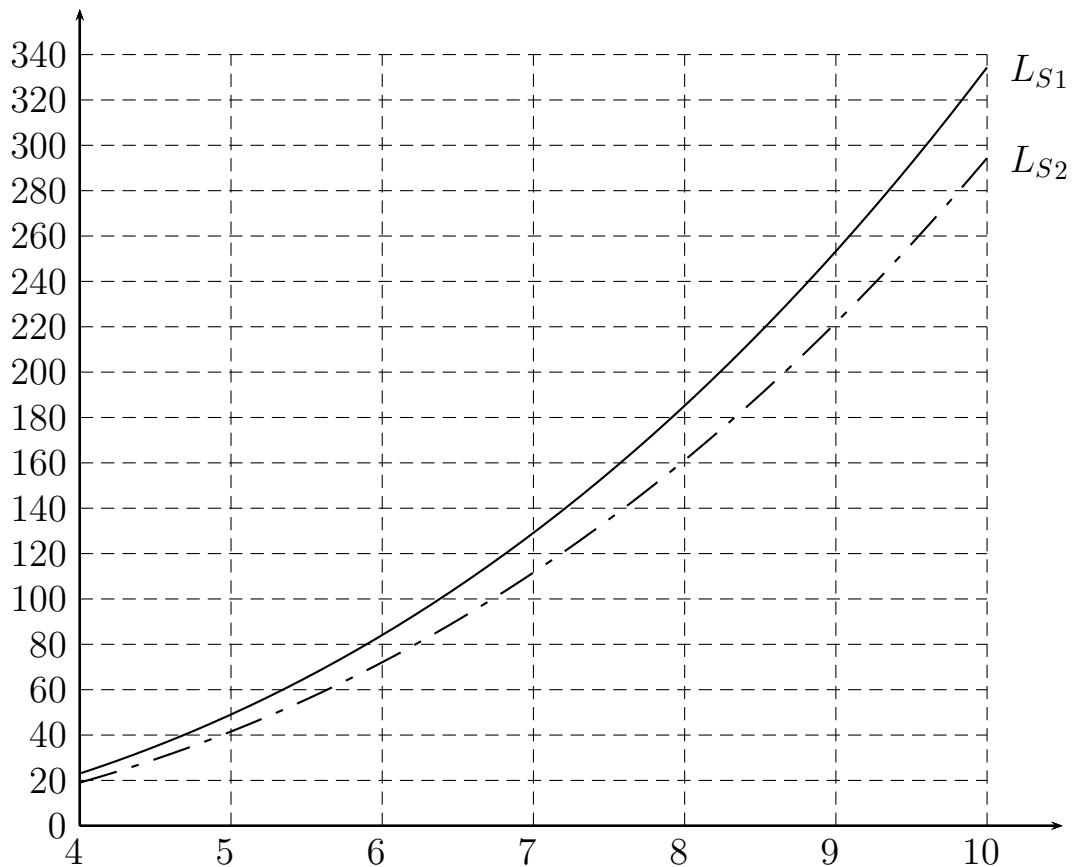
2.5.2 Сравнение оценок L_S

Сравним полученные оценки показателя L_S

$$L_{S1} \left(F_{n-2}^{(n)}, G_1 \right) = \frac{1}{6}n^3 + 2n^2 - \frac{13}{6}n - 11,$$

$$L_{S2} \left(F_{n-2}^{(n)}, G_1 \right) = \frac{1}{6}n^3 + \frac{3}{2}n^2 - \frac{5}{3}n - 11.$$

Данные оценки различаются незначительно. Для наглядного сравнения отразим полученные оценки на графике.



Как видно из графика, произошло некоторое улучшение показателя L_S .

Глава 3

Вычислительный метод определения оценки сложности.

В данной главе метод алгебраической декомпозиции применяется для автоматизации получения оценок любой функции в базисе Жегалкина. В результате формализации метода, использованного в предыдущей главе был разработан алгоритм, позволяющий это сделать. Также в главе рассматриваются особенности реализации данного алгоритма на практике и возникающие при этом особые случаи. В результате работы алгоритма происходит не только вычисление оценок произвольной булевой функции, но и ее минимизация. Основные моменты разработки алгоритма были доложены на конференции «Гагаринские чтения» [15].

3.1 Разработка алгоритма

Аналитическое нахождение показателя качества для булевых функций возможно путем декомпозиции этих функций в определенном базисе. При выполнении алгоритма применяется ФУ (1.2.1). В общем виде данная фор-

мула выглядит следующим образом:

$$F^{(n)} = x_j F_1^{(n-1)} \oplus F_2^{(n-1)} \quad (3.1.1)$$

Данное разложение возможно для любой булевой функции в базисе Жегалкина. Составим алгоритм для упрощения произвольной функции и нахождения её показателей качества.

1. Оптимальным для данного разложения будет выбор с максимальным значением повторений x_j в формуле.
2. После нахождения такого x_j , мы выносим его из всех конъюнкций, в которых оно встречается. Эти конъюнкции, с исключенным запищутся в первую остаточную подформулу. Те конъюнкции, в которых x_j не встречалось, запишутся во вторую остаточную подформулу.
3. После выделения двух остаточных подформул к каждой из них можно применить то же разложение (3.1.1).
4. Алгоритм будет рекурсивным: разложение на подформулы будет продолжаться, пока все формулы не перейдут в разряд реализованных (т. е. в остаточных не останутся только базисные функции).

3.1.1 Используемые обозначения

Пусть $X = \{x_1, x_2, \dots, x_j, \dots, x_n\}$ — множество булевых переменных. Произвольная булева функция $f^{(n)}(X)$ задается полиномом Жегалкина $F^{(n)} = K_1 \oplus K_2 \oplus \dots \oplus K_i \oplus \dots \oplus K_m$ в базисе $G_3 = \{\&, \oplus, 0, 1\}$, где

- n — число переменных,
- m — длина полинома Жегалкина,
- K_i — монотонная элементарная конъюнкция (ЭК) ранга r_i , $i = \overline{1, m}$,
- $\mathbf{r} = (r_1, r_2, \dots, r_m)$ — вектор рангов полинома Жегалкина.
- K'_i — монотонная ЭК (элементарная конъюнкция) ранга r'_i , $i = \overline{1, m'}$,
- r' — строение полинома Жегалкина F' уже упорядоченного вектора.

	x_1	\dots	x_j	\dots	x_n	\mathbf{r}
K_1	0/1		0/1		0/1	r_1
\dots						
K_m	0/1		0/1		0/1	r_m
\mathbf{p}	p_1	\dots	p_j	\dots	p_j	L_B

Таблица 3.1: Таблица, задающая полином Жегалкина

Полином Жегалкина $F^{(n)}$ задаётся при помощи матрицы $K_{i,j}$ (таблица 3.1) размером $[m \times n]$, представляющей в виде таблицы с числом строк — $(m+1)$ и столбцов — $(n+1)$. Определяются матрица и таблица следующим образом: в ячейку $K_{i,j}$ пишется 1, если $x_j \in \{K_i\}$, иначе $K_{i,j} = 0$ ($i = \overline{1, m}$, $j = \overline{1, n}$), где под $\{K_i\}$ понимаем множество переменных, образующих ЭК K_i .

В столбец ($i = \overline{1, m}$, $n+1$) записывается ранг элементарной конъюнкции K_i , вычисляемый следующим образом:

$$r_i = r_{i,n+1} = \sum_{j=1}^n K_{i,j}$$

Вектор $\mathbf{r} = (r_1, \dots, r_i, \dots, r_m)$ рангов ЭК полинома Жегалкина, упорядочивается для алгоритма один раз отношением “ \geq ”. Получаем $r_1 \geq \dots \geq r_i \geq \dots \geq r_m$. [10-13, 41]

В строку $(n+1, j = \overline{1, n})$, записывается p_j — число повторений переменной x_j , $j = \overline{1, n}$, в формуле $F^{(n)}$:

$$p_j = p_{m+1,j} = \sum_{i=1}^m K_{i,j}.$$

Так мы получаем вектор $\mathbf{p} = (p_1, p_2, \dots, p_i, \dots, p_n)$ повторяемости переменных из множества $X = \{x_1, \dots, x_j, \dots, x_n\}$ в формуле $F^{(n)}$, т. е. переменная x_i , $j = \overline{1, n}$, повторяется в формуле $F^{(n)}$ число p_i раз.

В ячейку $(m+1, n+1)$ пишется $L_B = \sum_{i=1}^m r_i$ — число букв в формуле $F^{(n)}$ (это вспомогательный параметр).

Для решения задачи нами применяется функциональное уравнение (ФУ)

$$F^{(n)} = \left(\left(x_i F_1^{(n-1)} \right) \oplus F_2^{(n-1)} \right), \quad (3.1.2)$$

где нижние индексы 1 и 2 — номера соответствующих остаточных функций, рассматриваемых на одном множестве $X' = X \setminus \{x_i\}$. Будем их в алгоритме записывать соответственно как $F' = F^{(n-1),1}$ и $F'' = F^{(n-1),2}$.

На основе частично упорядоченного отношения “ \geq ” множества рангов r_i ЭК K_i , $i = \overline{(1, m)}$, извлекаем p_j соответствующих ЭК, и из них выносим за скобки переменную x_j , получая с тем же порядком полином F' , длиной p_j . Оставшееся частично упорядоченное множество ЭК (на основе их рангов) определяет с тем же порядком полином F'' , длиной $m - p_j$.

- $t_1 = 1, 2, \dots$ — начало и продолжение записи функций F' и F'' , а так

n' или n''	m' или m''	F' или F''	L_F
$n' = n - 1$	$m' = p_{jmax}$	F'	$L_F = L_F + 1$
$n'' = n'$	$m'' = m = m'$	F''	$L_F = L_F + 1$
...

Таблица 3.2: Декомпозиция и запись результатов (1-я и 2-я строки и т. д.).

же номер последней записанной функции в табл. 3.2;

- $t_2 = 1, 2, \dots$ — начало и продолжение чтения таблицы 3.2, а так же номер последней прочитанной функции.
- $jmax$ — индекс переменной x_j с максимальным числом p_j повторений в табл. 3.1. Таких переменных может быть несколько, выбираем из них с меньшим номером переменной и меньшим номером ЭК, для сохранения порядка рангов ЭК K_i . Табл. 3.1 упрощается после каждого применения ФУ (3.1.2).

Как следует из (3.1.2), реализуемая функция $f^{(n)}$ — формула $F^{(n)}$ (и получаемые остаточные), в общем случае, разбивается на две более простые по сравнению с ней и две двухместные, базисные, соединяющие первые функции и выделяемую переменную x_j , в одну формулу, записываемые в таблицу. Процесс продолжается, пока все остаточные функции на основе декомпозиции не перейдут в группу реализованных или пока выполняется условие $t_2 < t_1$. Так будет получена формула $F_{\text{с.ф.}}^{(n)}$ и оценка $L_F(F_{\text{с.ф.}}^{(n)}, G_3)$. Приведём таблицу 3.3 для функции $F_{\text{с.ф.}}^{(7)}$ из примера 1 раздела 3.1

Вначале, по мере выполнения алгоритма, число остаточных функций, как правило, удваивается, а сами функции упрощаются. При этом число

№	Аргумент 1	Аргумент 2	Функция	Логический элемент
1	x_3	x_4	$g_{\&,1}$	$g_{\&,1}$
2	$g_{\&,1}$	x_5	$g_{\oplus,1}$	$G_{\oplus,1}$
3	x_2	$g_{\oplus,1}$	$g_{\&,2}$	$g_{\&,2}$
4	$g_{\&,2}$	x_6	$g_{\oplus,2}$	$G_{\oplus,2}$
5	x_1	$g_{\oplus,2}$	$g_{\&,3}$	$g_{\&,3}$
6	$g_{\&,3}$	x_7	$g_{\oplus,3}$	$G_{\oplus,3}$

Таблица 3.3: Результат работы алгоритма для $F_{\text{с.ф.}}^{(7)}$ из раздела 3.1.

базисных функций, соединяющих их в одну формулу, увеличивается.

Обсуждаемый алгоритм на основе ФУ (3.1.2) относится к алгоритмам градиентного типа. Он позволяет строить формулу $F_{\text{с.ф.}}^{(n)}$, реализующую функцию $f^{(n)}$, возможно, с минимальным значением показателя L_F за число шагов j , $j = \overline{1, n}$.

3.1.2 Алгоритм

Построим алгоритм определения значения верхней оценки показателя сложности L_F . Для каких-то классов функций оценки будут минимальными.

Дано:

$n, m, F^{(n)}$ — {задана правильная формула, при вводе проверяется}

Шаг 1. Подготовка начальных данных.

Для исходной формулы $F^{(n)}$ нужно заполнить таблицу 3.1, получая векторы r и p , т. е. правый столбец ($i = \overline{1, m}, n+1$), и нижнюю строку ($m+1, j = \overline{1, n}$). Теперь выполним инициализацию: $L_F = 0, t_1 = 0, t_2 = 0$.

Шаг 2.

Если $m = 1$, то

Начало $L_F = L_F + r_1 - 1$,

Переход к шагу 6 {чтение}

Конец

Шаг 3.

Если $r_1 = 1$, то

Начало $L_F = L_F + m - 1$,

Переход к шагу 6 {чтение}

Конец

Шаг 4.

Найдём $p_{jmax} = \max(p_1, p_2, \dots, p_j, \dots, p_n)$, где $j = \overline{1, n}$, и соответствующее x_{jmax} равное $jmax$. Отметим, что в общем случае $p_{jmax} \neq jmax$.

Шаг 5. {Запись.}

Теперь применим ФУ (3.1.1) и определим первую остаточную функцию $F' = F_1$.

По x_i табл. 3.1 нужно выбрать номера ЭК, содержащие переменную x_{jmax} , число которых равно p_{jmax} , и сложить (виртуально) эти ЭК, исключая переменную x_{jmax} , по модулю 2.

Фактически мы переписываем ЭК из табл. 3.1 в новую табл. 3.1, сохранив порядок и исключая переменную x_{jmax} . Этим мы получаем полином Жегалкина F' , а также подсчитываем векторы r и p .

После этого вычисляем: $n' = n - 1$ — число переменных полинома Жегалкина F' ,

$m = p_j$ — длину полинома Жегалкина F' , и увеличиваем t_1 : $t_1 = t_1 + 1$.

В строку с номером t_1 табл.2 запишем n' , m' , F' .

Отсюда нам необходимо определить вторую остаточную функцию $F'' = F_2$ полинома Жегалкина из табл. 3.1.

Для этого по табл. 3.1 выберем номера ЭК, не содержащие переменную x_{jmax} , и сложим (виртуально) эти ЭК, по модулю 2.

Проще говоря переписываем ЭК из табл. 3.1 в новую табл. 3.1, сохраняя порядок. Потом подсчитываем векторы r и p и получаем полином Жегалкина F'' (F_2).

Далее вычисляем: $n'' = n'$ — число переменных полинома Жегалкина F'' ,

$m'' = m - m' = m - p_j$ — длину полинома Жегалкина F'' , и увеличиваем t_1 : $t_1 = t_1 + 1$.

В строку с номером t_1 табл. 3.2 пишем n'' , m'' , F''

И так далее ...

Шаг 6. {Чтение.}

Сначала вычислим $t_2 = t_2 + 1$.

Для результатов здесь используется табл. 3.2, в которую с начала работы записываем не более двух полиномов Жегалкина F' и F'' , и затем, пока они имеются, из нее (табл.2) считываем по одному полиному.

$$L_F(F^{(n)}, G_3) = L_F + 1.$$

Шаг 7.

Если $t_2 \leq t_1$, то переход к шагу 3.2.

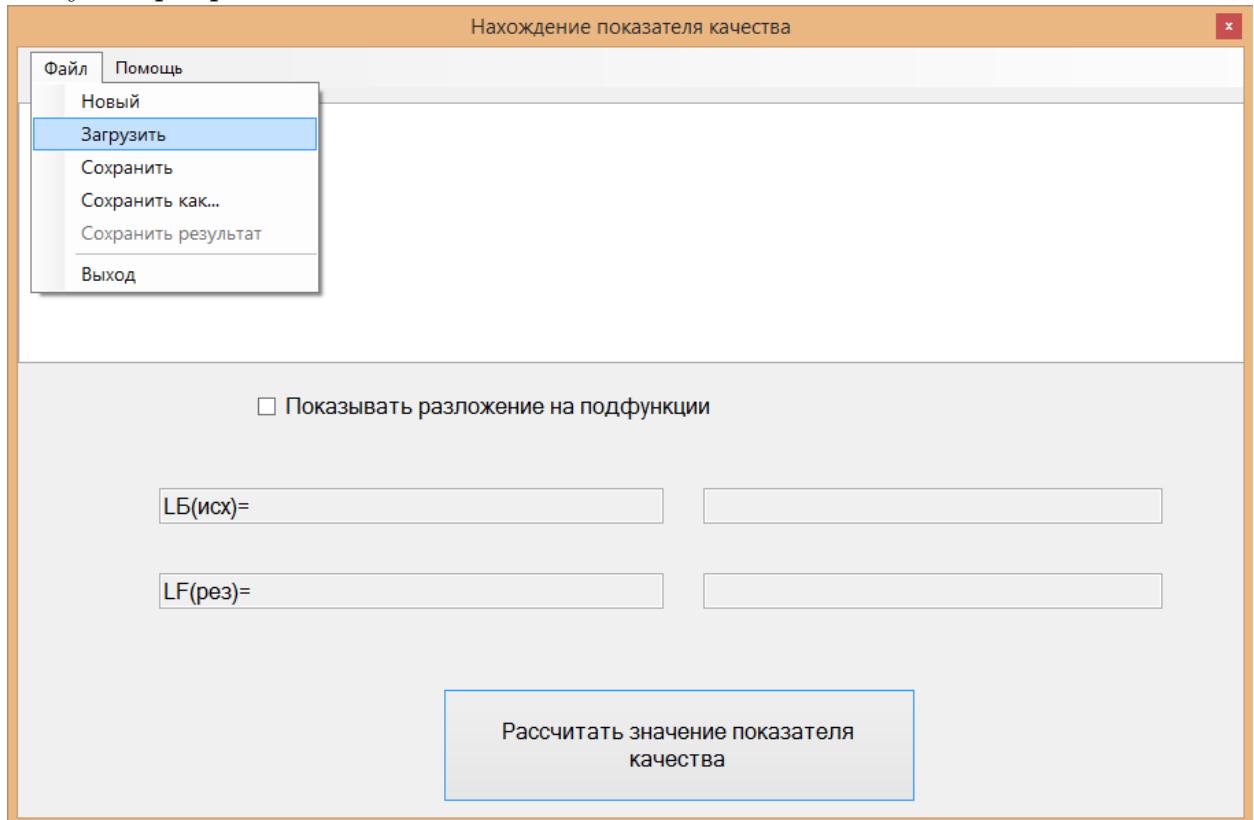
КОНЕЦ.

В результате работы алгоритма мы получаем верхнюю оценку сложности заданной в виде полинома Жегалкина логической функции.

3.2 Программная реализация

3.2.1 Пример работы программы

Запуск программы:



Минимизируемую функцию можно ввести в текстовое поле, или загрузить из текстового файла. При этом для удобства ввода переменные записываются в виде xn , знак умножения опускается, а под знаком "+" подразумевается сложение по модулю 2 " \oplus ".

Результат работы программы:

Нахождение показателя качества

Файл Помощь

$x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_4 + x_5 + x_6 + x_7$

Показывать разложение на подфункции

ЛБ(исх) = 12	Количество операций = 244
LF(рез) = 8	Время работы: 12 мс

Рассчитать значение показателя
качества

Выводятся: количество букв в исходной формуле, количество подформул в минимизированной формуле, время работы алгоритма и количество совершенных операций

При желании можно показать разложение введённой формулы на подформулы:

Нахождение показателя качества

Файл Помощь

$x_1 x_2 x_3 x_4 x_5 + x_1 x_2 x_3 x_4 x_6 + x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_3 x_4 x_5 x_6 + x_2 x_3 x_4 x_5 x_6$

Показывать разложение на подфункции

ЛБ(исх) = 30	Количество операций = 290
LF(рез) = 19	Время работы: 16 мс

Рассчитать значение показателя
качества

$$F_0 = x_1 F_1 + F_2$$

$$F_1 = x_2 x_3 x_4 x_5 + x_2 x_3 x_4 x_6 + x_2 x_3 x_5 x_6 + x_2 x_4 x_5 x_6 + x_3 x_4 x_5 x_6 = x_2 F_3 + F_4$$

$$F_2 = x_2 x_3 x_4 x_5 x_6$$

$$F_3 = x_3 x_4 x_5 + x_3 x_4 x_6 + x_3 x_5 x_6 + x_4 x_5 x_6 = x_3 F_5 + F_6$$

$$F_4 = x_3 x_4 x_5 x_6$$

$$F_5 = x_4 x_5 + x_4 x_6 + x_5 x_6 = x_4 F_7 + F_8$$

$$F_6 = x_4 x_5 x_6$$

$$F_7 = x_5 + x_6$$

$$F_8 = x_5 x_6$$

Сохранить Результат

Полученное разложение возможно сохранить в файл.

3.2.2 Примеры работы алгоритма

Разберем алгоритм на основе следующего примера:

$$F^{(3)} = x_1x_2x_3 \oplus x_1x_4 \oplus x_5.$$

Для начала представим данный полином в виде таблицы (матрицы):

	x_1	x_2	x_3	x_4	x_5	\mathbf{p}
K_1	1	1	1	0	0	3
K_2	1	0	0	1	0	2
K_3	0	0	0	0	1	1
\mathbf{r}	2	1	1	1	1	6

Запишем получившуюся матрицу в табл. 3.2, и установим начальные $t_1 = 1$ и $t_2 = 1$.

Для начального прохода установим счётчик $L_F = 0$.

Теперь найдём переменную с максимальным рангом. Очевидно, что в данной таблице максимальный ранг имеет переменная x_1 .

Отсюда легко видеть, что разложение функции $F^{(3)}$ принимает вид:

$$F^{(5)} = x_1(x_2x_3 \oplus x_4) \oplus x_5.$$

Таким образом получим две остаточные функции:

$$F_1 = x_2x_3 \oplus x_4, \quad F_2 = x_5.$$

Следовательно нам нужно увеличить счётчик подфункций $L_F = L_F + 2$, а также счётчик записи t_1 : $t_1 = t_1 + 2$.

Теперь из матрицы мы выбираем те строки, в которых элемент $x_1 = 1$, и копируем их в новую таблицу, исключая столбец с элементом x_1 . Аналогично мы выбираем строки с элементом $x_1 = 0$, и копируем их во вторую таблицу. Эти таблицы соответствуют функциям F_1 и F_2 :

	x_2	x_3	x_4	x_5	p
K_1	1	1	0	0	2
K_2	0	0	1	0	1
r	1	1	1	0	3

(3.2.1)

	x_2	x_3	x_4	x_5	p
K_1	0	0	0	1	1
r	0	0	0	1	1

(3.2.2)

В полученных таблицах (3.2.1) и (3.2.2) исключим элементы с рангом $r_i = 0$ и получим новые таблицы:

	x_2	x_3	x_4	x_5	p
K_1	1	1	0		2
K_2	0	0	1		1
r	1	1	1		3

(3.2.3)

	x_5	p
K_1	1	1
r	1	1

Следовательно, нам нужно увеличить счётчик чтения $t_2 = t_2 + 1$, таким образом он стал равным 2.

Аналогично предыдущему действию мы считываем функцию с индексом $t_2 = 2$ из табл. 2. Теперь текущая функция имеет вид (3.2.3):

$$F_1 = x_2 x_3 \oplus x_4.$$

Далее эта функция раскладывается на подфункции описанным ранее способом.

Чтение из таблицы продолжается, пока значение счётчика t_2 не превысит значение t_1 . При этом текущее значение L_F является количеством подфункций в минимизированной формуле. Вид этой формулы легко восстанавливается из табл. 2.

3.2.3 Частные случаи

Пример 1. Рассмотрим случай, когда количество конъюнкций $m = 1$. В этом случае формула представляет перемножение переменных.

$$F^{(n)} = x_1 x_2 \dots x_n.$$

Т.к. количество подфункций на единицу меньше количества переменных, то в таком случае можно считать, что

$$L_F = n - 1.$$

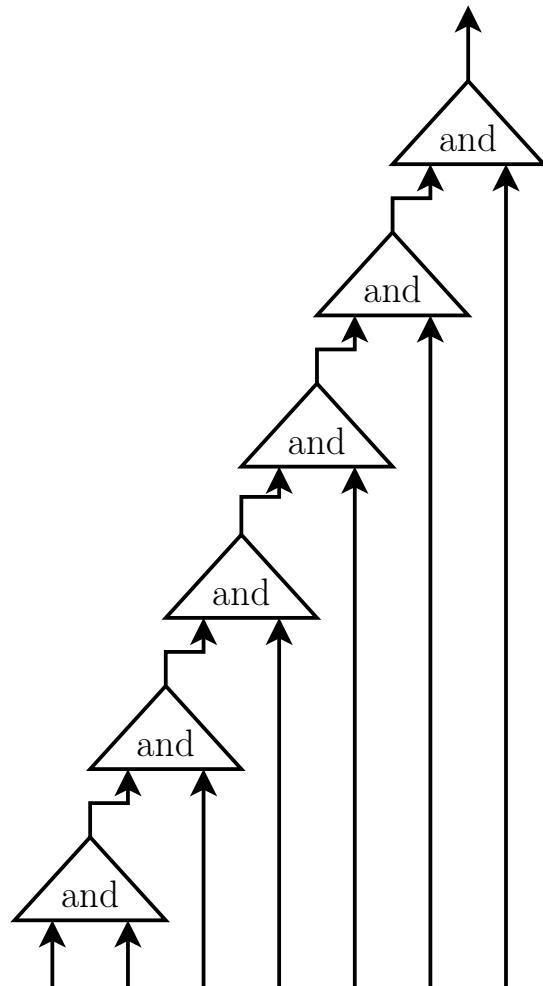
Если в виде формулы данный случай не представляет особых сложностей, то в виде схемы может быть несколько вариантов размещения элементов.

Пусть

$$F^{(7)} = x_1 x_2 x_3 x_4 x_5 x_6 x_7.$$

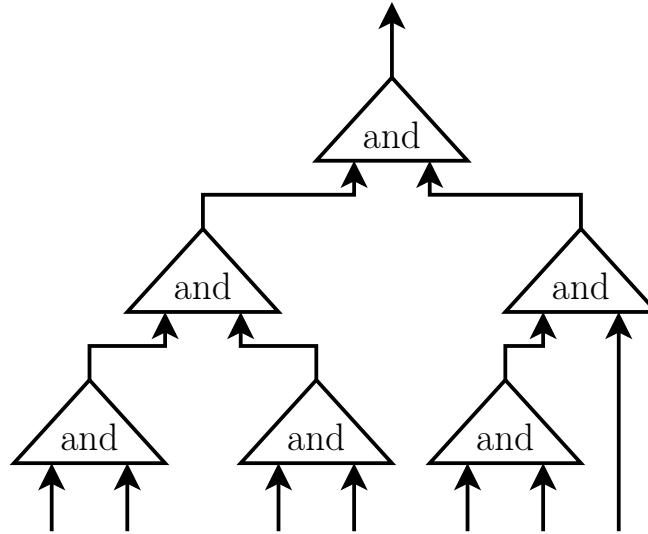
Тогда возможны два случая:

1й случай:



$$F^{(7)} = (((((x_1 x_2) x_3) x_4) x_5) x_6) x_7). \quad (3.2.4)$$

2й случай:



$$F^{(7)} = (((x_1 x_2) (x_3 x_4)) (x_5 x_6) x_7). \quad (3.2.5)$$

Как видно, в первом случае формула имеет глубину $Dep_F = 6$, тогда как во втором $Dep_F = 3$. Очевидно, что параллельная декомпозиция будет иметь преимущества по скорости быстродействия. Для облегчения вычислений и большей скорости работы алгоритма используется формула (3.2.4), но при выводе выходных данных используется форма (3.2.5). Она облегчает восприятие.

Пример 2. Вторым особыенным случаем является вариант, когда $x_{jmax} = 1$. Т.к. максимальный ранг переменной, а, следовательно, и все ранги равны , значит формула состоит из переменных, сложенных по модулю 2

$$F^{(n)} = x_1 \oplus x_2 \oplus \dots \oplus x_n.$$

Разбирается данный случай аналогично примеру 1

$$F^{(5)} = (((x_1 \oplus x_2) \oplus x_3) \oplus x_4) \oplus x_5.$$

$$F^{(5)} = (((x_1 \oplus x_2) \oplus (x_3 \oplus x_4)) \oplus x_5).$$

Пример 3. Пусть

$$F^{(2)} = x_1 \oplus x_1 x_2. \quad (3.2.6)$$

Если вынести в данном примере переменную с максимальным рангом за скобки, то мы получим следующее:

$$F^{(2)} = x_1 (1 \oplus x_2).$$

Это никак не влияет на количество знаков действий. Следовательно, формулу (3.2.6) можно не раскладывать, а только учесть количество подформул.

Теперь рассмотрим те случаи, когда при разложении отсутствует одна из подформул.

$$F^{(n)} = x_{j_{max}} F^{(n-1),1} \oplus F^{(n-1),2}.$$

Пример 4. Пусть

$$F^{(n)} = x_{j_{max}} \oplus F^{(n-1),2}.$$

В данном случае очевидно, что переменная $x_{j_{max}}$ имеет количество повторений $p_{j_{max}} = 1$, иначе существовала бы подформула $F^{(n-1),1}$. Но значит и остальные переменные в таком случае повторяются всего один раз. Можно привести подобный пример.

$$F^{(4)} = x_1 \oplus x_2 x_3 x_4. \quad (3.2.7)$$

Для того, чтобы избежать подобной ситуации в алгоритме выполняется

первоначальное упорядочивание по количеству переменных в конъюнкциях. Т.о. пример (3.2.7) приобретёт следующий вид

$$F^{(4)} = x_2x_3x_4 \oplus x_1,$$

где максимальный ранг будет иметь переменная. Следовательно, единственный возможный вид формулы в таком случае

$$F^{(4)} = x_1 \oplus x_2 \oplus x_3 \oplus x_4.$$

Данный вид формулы разбирается в примере 2.

Пример 5. Рассмотрим последний случай, когда

$$F^{(n)} = x_{j_{max}} F^{(n-1),1}.$$

Пусть

$$F^{(4)} = x_1x_2 \oplus x_1x_3 \oplus x_1x_4 = x_1(x_2 \oplus x_3 \oplus x_4).$$

В данном случае мы просто учитываем, что появляется всего одна остаточная подформула, а, следовательно, мы увеличиваем счётчик записи всего на единицу $t_1 = t_1 + 1$.

Заключение

Итак в работе достигнуты следующие результаты:

- Рассмотрены основные методы синтеза логических формул, и соответствующих схем из функциональных элементов.
- Предложен метод распараллеливающей структурно-функциональной декомпозиции булевых функций, позволяющие аналитически получать верхнюю оценку сложности показателя L_F для представления функции — полинома $F^{(n)}$ в классе формул, а также — в классе схем S .
- Данный метод применён для аналитического нахождения ряда оценок для полинома Жегалкина строения $F_{n-2}^{(n)}$ в классах формул и схем из функциональных элементов.
- Выделены частные случаи получения счетных множеств булевых функций минимальной сложности L_F функции — полинома $F^{(n)}$ (а, также для сложности показателя L_S для схем S).
- Найдена эффективная реализация вычислительного алгоритма синтеза булевых формул на основе приведения их к скобочному виду.
- Проведена рационализация вычислительного алгоритма синтеза схем из функциональных элементов на основе операции ветвления некоторых их выходов.
- На основании построенного алгоритма написана программа, которая была зарегистрирована в РосПатенте [24].

Следующим шагом может быть распространение метода на другие классы формул, представленных в базисе G_3 . Кроме того не менее важным представляется получение представления полученных результатов в классе схем.

Литература

- [1] Александров П. С. Комбинаторная топология. М.: Гос. изд. технико-теоретической литературы, 1947.
- [2] Андреев А. Е. О минимизации булевых функций //Докл. АН СССР. 1984, т. 274, № 2.
- [3] Андреев А. Е. О сложности монотонных функций //Докл. АН СССР. 1986, т. 283, 2, С. 265-269.
- [4] Андреев А. Е. Математический сборник. 1985, т. 127(169), № 6, с. 147-172.
- [5] Berge C., Ray-Chaudhuri D. // "Hypergraph Seminar, Ohio State University 1972 Lecture Notes in Mathematics 411 Springer-Verlag.
- [6] Березин И. С., Жидков Н. П. Методы вычислений. В 2 т. - М.: Государственное издательство физико-математической литературы, 1962.
- [7] Бибило П. Н., Енин С. В. Синтез комбинационных схем методами функциональной декомпозиции. — Минск: Наука и техника, 1987.
- [8] Бибило П. Н. Декомпозиция булевых функций на основе решения логических уравнений. Часть I //Изв. РАН. ТиСУ. 2002, № 4.

- [9] Ванг Л., Егорова Е. К., Мокряков А. В. Развитие теории гиперграфов // Изв. РАН. ТИСУ. 2018, № 1, С. 111-116.
- [10] Выхованец В. С. Синтез эффективных математических моделей дискретной обработки данных на основе алгебраической и понятийной декомпозиции предметной области // Дисс. ...докт. техн. наук. М., 2007.
- [11] Гаврилов М. А. Декомпозиция комбинационных автоматов — В кн.: Избр. тр. Теория релейных устройств и конечных автоматов. — М.: Наука, 1983.
- [12] Городецкий Л. М., Гурский Л. И., Мищенко В. А. и др. // Интеллектуальные системы автоматизированного проектирования больших и сверхбольших интегральных микросхем — М.: Радио и связь, 1988. 328 с.
- [13] Гурченков А. А., Егорова Е. К. Автоматизация задачи определения сложности булевой функции // Инженерный журнал: наука и инновации. 2014, № 5(29), С. 10-21.
- [14] Гурченков А. А., Егорова Е. К. Особенности автоматизации синтеза булевых функций // Инженерный журнал: наука и инновации. 2013, № 12(24), С. 53-61.
- [15] Егорова Е. К. Алгоритм минимизации сложности представления произвольной булевой функции // М.: МАТИ, статья на XXXIX международной молодежной научной конференции «Гагаринские чтения», 2013, С. 56-57.

- [16] Егорова Е. К. Показатели качества реализации симметрических полиномов Жегалкина степени $n - 2$ // Труды межвузовской молодёжной научно-практической конференции «Информационные и телекоммуникационные технологии». — М.: МАТИ, 2009, С. 30-46.
- [17] Егорова Е. К. Программный комплекс минимизации сложности представления произвольной булевой функции в классе формул // М.: МАТИ, статья на XXXVII международной молодежной научной конференции «Гагаринские чтения», т. 5, 2012, С. 56-58.
- [18] Егорова Е. К. Сложность представления симметрических булевых функций в классе полиномов Жегалкина // М.: МАТИ, статья на XXXVII международной молодежной научной конференции «Гагаринские чтения». т. 5, 2011, С. 88-90.
- [19] Егорова Е. К. О методах синтеза булевых формул и схем // М.: МАТИ, статья на XLII международной молодежной научной конференции «Гагаринские чтения». 2016, С. 211-212.
- [20] Егорова Е. К., Мокряков А. В. Минимизация сложности представления произвольной булевой функции в классе формул программным методом // Минск: I Форум союзного государства вузов инженерно-технического профиля – Материалы встречи молодых учёных «Молодёжные идеи и проекты», 2012, С 7-8.
- [21] Егорова Е. К., Мокряков А. В. Об исчислении экстремальных 2-комплексов // М.: МАТИ, статья на XXXVII международной мо-

лодежной научной конференции «Гагаринские чтения». т. 5, 2011,
С. 111-112.

- [22] Егорова Е. К., Чебурахин И. Ф. Автоматизация и оптимизация синтеза дискретных устройств обработки информации и управления // Севастополь: СевНГУ, статья на Международной научно-технической конференции молодых ученых, аспирантов и студентов «Управление, автоматизация и окружающая среда-2012».
- [23] Егорова Е. К., Чебурахин И. Ф. О минимизации сложности и автоматизации эффективного представления булевых функций в классах формул и схем // Изв. РАН. ТиСУ. 2013, № 3, С. 121-129.
- [24] Егорова Е. К., Чебурахин И. Ф. ПРОГРАММА ДЛЯ ЭВМ Минимизация сложности представления произвольной булевой функции $f(n)$ (n переменных) в классе формул. Версия 1.0 // Официальный бюллетень «Программы для ЭВМ. Базы данных. Топологии интегральных микросхем» регистрационный № 2012616794 от 30 июля 2012.
- [25] Ерусалимский Я. М. Дискретная математика: теория, задачи, приложения. — М.: Вузовская книга, 2000, 280 с.
- [26] Журавлев Ю. И. Алгоритм построения минимальных дизъюнктивных нормальных форм для функций алгебры логики // Дискретная математика и математические вопросы кибернетики. — М.: Наука, 1974.
- [27] Журавлев Ю. И. Теоретико-множественные методы в алгебре логики // Проблемы кибернетики. 1962, № 8.

- [28] Закревский А. Д. Автоматизация логического синтеза дискретных устройств // Кибернетика. 1975, № 4, С. 100-108.
- [29] Зыков А. А. Гиперграфы // УМН. 1974. Т. XXIX. №6 (180). С. 89–154.
- [30] Ложкин С. А., Коноводов В. А. О сложности реализации булевых функций из некоторых классов, связанных с конечными грамматиками, формулами глубины альтернирования 3 // Вестник московского университета. Серия 1: Математика. Механика. 2014, № 3, С. 14-19.
- [31] Ложкин С. А., Шуплецов М. С. О динамической активности схем из функциональных элементов и построении асимптотически оптимальных по сложности схем с линейной динамической активностью // Ученые записки Казанского университета. Серия: физико-математические науки. 2014, № 3, С. 84-97.
- [32] Лупанов О. Б. О сложности реализации функций алгебры логики формулами // Проблемы кибернетики. Вып. 3. — М.: Наука, 1960.
- [33] Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: МГУ, 1984.
- [34] Месарович М., Такахара Я. Общая теория систем: математические основы. — М.: Мир, 1978, 311 с.
- [35] Миронов А. А., Мокряков А. В., Колбанов В. М. О k -мерных комплексах полностью описываемые степенями вершин // Труды ИСА РАН. Динамика неоднородных систем (под редакцией член.-корр. РАН Попкова Ю. С.), 2006 №10(2). 2006 г.

- [36] Моисеев Н. Н. Математические задачи системного анализа. — М.: Наука, 1981, 487 с.
- [37] А.В. Мокряков, П.С. Селин, В.И. Цурков Минимакс и восстановление по вектору в графах — М.: Физматлит, 2017, 309 с.
- [38] Новиков Ф. А. Дискретная математика для программистов. — С.-П.: ПИТЕР, 2000, 304 с.
- [39] Редькин Н. П. Доказательство минимальности некоторых схем из функциональных элементов // Проблемы кибернетики. Вып. 23 — М.: Наука, 1970.
- [40] Супрун В.П., Городецкий Д.А. Реализация бисимметрических булевых функций логическими схемами // Вычислительная техника. 2010.
- [41] Угольников А. Б. О реализации монотонных функций схемами из функциональных элементов // Проблемы кибернетики. Вып. 31. — М.: Наука, 1976.
- [42] Угольников А. Б. О реализации функций из замкнутых классов схемами из функциональных элементов // Математические вопросы кибернетики. Вып. 1. — М.: Наука, 1988.
- [43] Угольников А. Б. Глубина формул в неполных базисах // Математические вопросы кибернетики. / Под ред. С. В. Яблонского. Вып. 1. — М.: Наука, 1988.

- [44] Чебурахин И. Ф. Логическое управление: задачи, алгоритмы, показатели качества. - М.: МАТИ, 2003, 200 с.
- [45] Чебурахин И. Ф. Синтез дискретных управляющих систем и математическое моделирование - М.: Физматлит, 2004, 247 с.
- [46] Чебурахин И. Ф. Преобразования функциональных уравнений и показатели сложности булевых функций // Матер. IX Междунар. семинара «Дискретная математика и ее приложения», посвященного 75-летию со дня рождения акад. О. Б. Лупанова. М.: МГУ, 2007, С. 126–129.
- [47] Чебурахин И. Ф. Математические модели для интеллектуализации синтеза дискретных логических управляющих устройств на основе цифровых интегральных схем // Изв. РАН. ТиСУ. 2008, № 1, С. 68–77.
- [48] Чебурахин И. Ф. Сложность симметрических полиномов Жегалкина // XVII Междунар. школа-семинар «Синтез и сложность управляющих систем» им. акад. О. Б. Лупанова. Новосибирск, 2008, С. 180–185.
- [49] Чебурахин И. Ф., Цурков В. И. Синтез дискретных логических устройств обработки информации на основе теории агентов // Мехатроника, автоматизация, управление. 2011, № 3, С. 27–34.
- [50] Чебурахин И. Ф. Математические модели для минимизации и автоматизации синтеза дискретных управляющих систем // Мехатроника, автоматизация, управление. 2012, № 4, С. 5–13.
- [51] Чебурахин И. Ф. Методы декомпозиции булевых функций: алгорит-

мы, показатели качества, приложения // Изв. РАН. ТиСУ. 2003, № 5, С. 56-61.

- [52] Чебурахин И. Ф., Егорова Е. К. Автоматизация конструирования определённых счётных классов булевых функций и минимизация их сложности // Мехатроника, автоматизация, управление. 2014, № 8, С. 3-9.
- [53] Шеннон К. Работы по теории информации и кибернетики. — М.: ИЛ, 1963.
- [54] Юдин Д. Б., Горяшко А. П., Немировский А. С. Математические методы оптимизации устройств и алгоритмов АСУ / Под ред. Ю. В. Асафьева, В. А. Шабалина. — М.: Радио и связь, 1982, 288 с.
- [55] Яблонский С. В. Введение в дискретную математику. - М.: Наука, 1986.
- [56] Яблонский С. В. Об алгоритмических трудностях синтеза минимальных контактных схем // Проблемы кибернетики, М.: Физматгиз. 1959, № 2.
- [57] Яблонский С. В. Обзор некоторых результатов в области дискретной математики // Тр. Всес. конф. по проблемам теоретической кибернетики. — В сб.: Информационные методы / Под ред. С. И. Самойленко. — М.: АН СССР, Научный совет по комплексной проблеме «Кибернетика», 1970, № 5.
- [58] Boyar J., Peralta R., Pochuev D. On the multiplicative complexity

of Boolean functions over the basis $\{\wedge, \oplus, 1\}$ // Theor. Comput. Sci. Elsevier, 2000. T. 235, № 1. C. 43–57.

- [59] Mirwald R., Schnorr C.P. The multiplicative complexity of quadratic boolean forms // Theor. Comput. Sci. Elsevier, 1992. T. 102, № 2. C. 307–328.
- [60] Selezneva S.N. On the multiplicative complexity of quasi-quadratic Boolean functions // Moscow Univ. Comput. Math. Cybern. Allerton Press, 2015. T. 39, № 1. C. 18–25.
- [61] Sergeev I.S. A relation between additive and multiplicative complexity of Boolean functions. 2013.
- [62] Suprun V.P. The complexity of Boolean functions in the Reed–Muller polynomials class // Autom. Control Comput. Sci. Pleiades Publishing, 2017. T. 51, № 5. C. 285–293.

Приложение

Ниже представлен код программы, реализующей описанный выше метод алгебраической декомпозиции.

```
1 using System;
  using System.Text;
3 using System.Windows.Forms;
  using System.IO;

5

7 namespace WindowsFormsApplication1
{
9     public partial class Polinomial : Form
{
11         public Polinomial()
{
13             InitializeComponent();
Width = 910;
15             Height = 600;

17             openFD.InitialDirectory = saveFD.InitialDirectory = Directory.
GetDirectoryName();
}
19

21         private void SaveToFile(string path)
{
23             StreamWriter sw = new StreamWriter(path, false, Encoding.Default);
sw.WriteLine(textBoxFormula.Text);
sw.Close();
25         }

27         private void SaveResult(string file)
{
29             StreamWriter sw = new StreamWriter(file, false, Encoding.Default);
sw.WriteLine(textBoxResult.Text);
```

```

31     sw.Close();
32 }
33
34 private void loadToolStripMenuItem_Click(object sender, EventArgs e)
35 {
36     if (openFD.ShowDialog() == DialogResult.OK)
37     {
38         string path = openFD.FileName;
39         StreamReader sr = new StreamReader(path, Encoding.Default);
40         textFormula.Text = sr.ReadLine();
41         sr.Close();
42         saveFD.FileName = path;
43         openFD.FileName = "";
44     }
45 }

46
47 private void saveToolStripMenuItem_Click(object sender, EventArgs e)
48 {
49     string path = saveFD.FileName;
50     if (path == "")
51         saveAsToolStripMenuItem_Click(saveAsToolStripMenuItem, null);
52     else
53         SaveToFile(path);
54 }
55

56 private void newToolStripMenuItem_Click(object sender, EventArgs e)
57 {
58     textFormula.Clear();
59     saveFD.FileName = "";
60 }
61

62 private void saveAsToolStripMenuItem_Click(object sender, EventArgs e)
63 {
64     if (saveFD.ShowDialog() == DialogResult.OK)
65     {

```

```

        string path = saveFD.FileName;
67      SaveToFile(path);
    }

69  }

71  private void buttonGo_Click(object sender, EventArgs e)
{
73  int i;
    ZhPolinom pl;
75  pl = new ZhPolinom();
    int start = Environment.TickCount;
77  pl.LoadFromString(textBoxFormula.Text);
    textResult.Text = "LFрэз() = " + pl.IndexOfQualityLF();
    textLBinit.Text = "БЛисх() = " + Convert.ToString(pl.LBinit);
79  int ttime = Environment.TickCount - start;
81  textTime.Text = "Время_работы: " + Convert.ToString(ttime) + " _мс";
    textNop.Text = "Количество_операций = " + Convert.ToString(pl.Nop);
83  if (checkBoxShowResult.Checked)
{
85  textBoxResult.Text = "";
    Width = 1220;
87  textBoxResult.Visible = true;
    buttonSaveResult.Visible = true;
    SaveResultToolStripMenuItem.Enabled = true;
89  for (i = 0; i < pl.size; i++)
{
91  textBoxResult.Text += "F" + Convert.ToString(i) + " = "
    pl.Printf(i) + Environment.NewLine;
93  }
}
95 }

97  private void ExitToolStripMenuItem_Click(object sender, EventArgs e)
{
99  this.Close();
}

```

```

    }

101

    private void checkBoxShowResult_CheckedChanged(object sender ,
        EventArgs e)
103
    {
        if (!checkBoxShowResult.Checked)
105
        {
            textBoxResult.Visible = false;
107
            buttonSaveResult.Visible = false;
            SaveResultToolStripMenuItem.Enabled = false;
109
            Width = 910;
        }
111
    }

113
    private void SaveResultToolStripMenuItem_Click(object sender ,
        EventArgs e)
115
    {
        if (saveFD.ShowDialog() == DialogResult.OK)
        {
            SaveResult(saveFD.FileName);
        }
119
    }

121
    private void buttonSaveResult_Click(object sender , EventArgs e)
123
    {
        SaveResultToolStripMenuItem_Click(SaveResultToolStripMenuItem ,
            null);
    }
125
}

127

129
class ZhPolinom
{
131
    private string polinom ;

```

```

private System.Collections.ArrayList sz;
133
private System.Collections.ArrayList arrvar;
public long Nop;
135
public int LBinit;

137
private System.Collections.ArrayList tb = new System.Collections.
    ArrayList();
int countK, countvar;
139
int [ , ] table;
public int size;
141
public ZhPolinom()
143 {
    polinom = "x_1*x_2*x_4\oplus x_1*x_2*x_3\oplus x_1*x_3*x_4\oplus
        x_2*x_3\0";
145 }

147
private int AnalyzeString()
{
149     System.Collections.ArrayList arrK;
    int i, j, tmp, pos;
    char ch;
    string xj, ki = "", str;
153
    countK = 0;
155
    arrvar = new System.Collections.ArrayList();
    arrK = new System.Collections.ArrayList();
157
    for (i = 0; i < polinom.Length; i++)
159 {
    ch = polinom[ i ];
    switch (ch)
161 {
        case '\u': break;
        case 'x':
163

```

```

165      {
166          xj = "x";
167          i++; // LBinit++;
168          ch = polinom[ i ];
169          while ((ch >= '0') && (ch <= '9'))
170          {
171              xj = xj + ch;
172              i++;
173              ch = polinom[ i ];
174          }
175
176
177          if (arrvar.Count == 0) arrvar.Add(xj);
178          else
179          {
180              tmp = 0;
181              for (j = 0; j < arrvar.Count; j++)
182              {
183                  str = (string)arrvar[j];
184                  if (xj.Contains(str)) tmp++;
185              }
186              if (tmp == 0) arrvar.Add(xj);
187          }
188          ki = ki + xj;
189          i++;
190          break;
191      }
192      case '+':
193      {
194          countK++;
195          arrK.Add(ki);
196          ki = "";
197          break;
198      }
199      case '\0':

```

```

201                                     {
202                                         countK++;
203                                         arrK.Add( ki );
204                                         ki = "";
205                                         break;
206                                     }
207                                     default:
208                                     {
209                                         return 1;
210                                     }
211                                     }

213                                     string [] K = (string [])arrK.ToArray(typeof(string));
214                                     string [] var = (string [])arrvar.ToArray(typeof(string));
215
216                                     countvar = var.Length;
217                                     table = new int [countK + 1, countvar + 1];
218                                     Nop++;
219                                     for ( i = 0; i < countK; i++)
220                                     {
221                                         tmp = 0;
222                                         for ( j = 0; j < countvar; j++)
223                                         {
224                                             Nop++;
225                                             pos = K[ i ].IndexOf( var[ j ] );
226                                             if (( pos >= 0)
227                                                 table[ i , j ] = 1;
228                                             else
229                                                 table[ i , j ] = 0;
230                                             tmp += table[ i , j ];
231                                         }
232                                         table[ i , countvar ] = tmp;
233                                         Nop++;
234                                     }

```

```

235
for ( j = 0; j < countvar; j++)
237 {
    tmp = 0;
239     for ( i = 0; i < countK; i++)
241 {
    tmp += table [ i , j ];
243     Nop++;
}
245     table [ countK , j ] = tmp;
Nop++;
}
247     table [ countK , countvar ] = var.Length;
tb.Add( table );
249
return 0;
}
251

253     public long IndexOfQualityLF ()
{
255
    long LF;
257
    int i , j , k1 , k2 , tmp;
    int t1 , t2 ;
259
    int maxrg;
    int xmax , jmax;
261
    int [,] tbwr1 , tbwr2 , tbr ;
    int [] szwr1;
263
    int [] szwr2;
    int [] szr = new int [ 6 ];
265
    AnalyzeString();
267
    sz = new System.Collections.ArrayList();
    t1 = t2 = 0;
269
    LF = 0;

```

```

maxrg = 0;
271 szwr1 = new int [ 6 ];
szwr1[0] = countK;
273 szwr1[1] = countvar;
szwr1[2] = szwr1[3] = szwr1[4] = szwr1[5] = 1;
275 sz .Add( szwr1 );

277 while ( t2 <= t1 )
{
279     tmp = 0;
280     szr = ( int [] ) sz [ t2 ];
281     tbr = ( int [ , ] ) tb [ t2 ];
282     while (tmp < szr [ 1 ])
283     {
284         if ( tbr [ szr [ 0 ] , tmp ] == 0 )
285         {
286             tbr [ szr [ 0 ] , szr [ 1 ] ] ;
287         }
288         tmp++;
289     }
290     for ( i = 0; i < szr [ 0 ]; i ++ )
291     {
292         if ( tbr [ i , szr [ 1 ] ] == 0 ) tbr [ szr [ 0 ] , szr [ 1 ] ] ++ ;
293     }

294     xmax = jmax = 0;
295     for ( j = 0; j < szr [ 1 ]; j ++ )
296     {
297         if ( xmax < tbr [ szr [ 0 ] , j ] )
298         {
299             Nop++;
300             xmax = tbr [ szr [ 0 ] , j ];
301             jmax = j ;
302         }
303     }

```

```

305      szr [ 2 ] = jmax;

307      maxrg = 0;
308      for ( i = 0; i < szr [ 0 ]; i++)
309      {
310          if ( maxrg < tbr [ i , szr [ 1 ] ] )
311          {
312              Nop++;
313              maxrg = tbr [ i , szr [ 1 ] ];
314          }
315      }

317      if ( szr [ 0 ] == 1 )
318      {
319          Nop++;
320          LF += maxrg - 1;
321          t2++;
322          continue;
323      }

325      if ( maxrg == 1 )
326      {
327          Nop++;
328          LF += tbr [ szr [ 0 ] , szr [ 1 ] ] - 1;
329          t2++;
330          continue;
331      }

333          t1++;
334          szr [ 4 ] = t1;
335          szwr1 = new int [ 6 ];
336          szwr1 [ 1 ] = ( szr [ 1 ] - 1 );
337          szwr1 [ 0 ] = xmax;
338          szwr1 [ 2 ] = szwr1 [ 4 ] = szwr1 [ 5 ] = 1;
339          szwr1 [ 3 ] = t2;

```

```

sz .Add( szwr1 ) ;

341 tbwr1 = new int [ szwr1 [ 0 ] + 1, szwr1 [ 1 ] + 1 ];
tb .Add( tbwr1 ) ;

343 LF++;

345 szwr2 = new int [ 6 ];
szwr2 [ 1 ] = ( szr [ 1 ] - 1 );
347 szwr2 [ 0 ] = szr [ 0 ] - xmax;
szwr2 [ 2 ] = szwr2 [ 4 ] = szwr2 [ 5 ] = - 1 ;
349 szwr2 [ 3 ] = t2 ;
tbwr2 = new int [ szwr2 [ 0 ] + 1, szwr2 [ 1 ] + 1 ];
351 if ( szwr2 [ 0 ] > 0 )
{
353 t1++;
szr [ 5 ] = t1 ;
355 sz .Add( szwr2 );
tb .Add( tbwr2 );
357 LF++;
}
359

k1 = k2 = 0;
361 if ( szwr2 [ 0 ] > 0 )
{
363 for ( i = 0; i < szr [ 0 ]; i++)
{
365 if ( tbr [ i , jmax ] == 1 )
{
367 tmp = 0;
for ( j = 0; j < jmax; j++)
{
369
tbwr1 [ k1 , j ] = tbr [ i , j ];
371 tmp += tbwr1 [ k1 , j ];
Nop++;
373 }
375 for ( j = jmax + 1; j < szr [ 1 ]; j++)

```

```

375          {
376          tbwr1[k1, j - 1] = tbr[i, j];
377          tmp += tbwr1[k1, j - 1];
378          Nop++;
379      }
380      tbwr1[k1, szwr1[1]] = tmp;
381      Nop++;
382      k1++;
383  }
384  else
385  {
386      tmp = 0;
387      for (j = 0; j < jmax; j++)
388      {
389          tbwr2[k2, j] = tbr[i, j];
390          tmp += tbwr2[k2, j];
391          Nop++;
392      }
393      for (j = jmax + 1; j < szr[1]; j++)
394      {
395          tbwr2[k2, j - 1] = tbr[i, j];
396          tmp += tbwr2[k2, j - 1];
397          Nop++;
398      }
399      tbwr2[k2, szwr2[1]] = tmp;
400      Nop++;
401      k2++;
402  }
403
404  for (j = 0; j < szwr1[1]; j++)
405  {
406      tmp = 0;
407      for (i = 0; i < szwr1[0]; i++)
408      {

```

```

tmp += tbwr1[i , j];
411 Nop++;
}
413 tbwr1[szwr1[0] , j] = tmp;
Nop++;
415 }
417 for (j = 0; j < szwr2[1]; j++)
{
tmp = 0;
419 for (i = 0; i < szwr2[0]; i++)
{
tmp += tbwr2[i , j];
421 Nop++;
}
423 tbwr2[szwr2[0] , j] = tmp;
425 Nop++;
}
427 tbwr1[szwr1[0] , szwr1[1]] = tbwr2[szwr2[0] , szwr2[1]] =
szr[1] - 1;
}
429 else
{
431 for (i = 0; i < szr[0]; i++)
{
tmp = 0;
433 for (j = 0; j < jmax; j++)
{
tbwr1[i , j] = tbr[i , j];
435 tmp += tbwr1[i , j];
Nop++;
}
437 for (j = jmax + 1; j < szr[1]; j++)
{
tbwr1[i , j - 1] = tbr[i , j];
439 tmp += tbwr1[i , j - 1];
}
441 for (j = jmax + 1; j < szr[1]; j++)
{
tbwr1[i , j - 1] = tbr[i , j];
443 tmp += tbwr1[i , j - 1];
}
}
}

```

```

        Nop++;
445      }
        tbwr1[ i , szwr1[ 1 ] ] = tmp;
447      Nop++;
        }

449      for ( j = 0; j < szwr1[ 1 ]; j++)
{
        tmp = 0;
        for ( i = 0; i < szwr1[ 0 ]; i++)
{
        tmp += tbwr1[ i , j ];
453      Nop++;
        }

455      tbwr1[ szwr1[ 0 ] , j ] = tmp;
        Nop++;
459      }

461      tbwr1[ szwr1[ 0 ] , szwr1[ 1 ] ] = szr[ 1 ] - 1;
        }

463      t2++;

465      }

467      size = tb.Count;
        return LF;
        }

469      public int LoadFromString( string plnm )
471      {
        polinom = plnm + '\0';
473      return 0;
        }

475      public string PrintF(int indx)
{

```

```

479     int [] szf = new int [2];
500     szf = (int []) sz [indx];
501     int [,] tbf = new int [szf [0] + 1, szf [1] + 1];
502     tbf = (int [,]) tb [indx];
503     string func = "";
504
505     int i, j;
506
507     string [] var = (string []) arrvar.ToArray (typeof (string));
508     System.Collections.ArrayList exvar = new System.Collections.
509         ArrayList ();
510
511     System.Collections.ArrayList varf = new System.Collections.
512         ArrayList (var);
513
514     int tmp=0;
515
516
517     tmp = ((int []) sz [indx]) [3];
518
519     while (tmp >= 0)
520     {
521
522         exvar.Add (((int []) sz [tmp]) [2]);
523
524         tmp = ((int []) sz [tmp]) [3];
525
526     }
527
528
529     for (j = 0; j < exvar.Count; j++)
530     {
531
532         varf.RemoveAt ((int) exvar [j]);
533
534     }
535
536
537     if (indx > 0)
538     {
539
540         for (i = 0; i < szf [0]; i++)
541         {
542
543             for (j = 0; j < szf [1]; j++)
544             {
545
546                 if (tbf [i, j] == 1) func = func + "\u" + (string) varf [j]
547                     + "\u";
548
549             }
550
551             if (tbf [i, szf [1]] == 0) func += "\u1\u";
552
553         }
554
555     }
556
557
558
559

```

```

511      if ( i < szf[0] - 1) func += "++";
512      }
513  }

515      if (szf[4] > 0)
516      {
517          if (indx > 0) func += "=";
518          func += varf[szf[2]] + "F" + szf[4];
519          if (szf[5] >= 0)
520          {
521              func += "+F" + szf[5];
522          }
523  }

525      return func;
526  }

527  }
528 }

529 }
```