УЧРЕЖДЕНИЕ РОССИЙСКОЙ АКАДЕМИИ НАУК

САНКТ-ПЕТЕРБУРГСКИЙ АКАДЕМИЧЕСКИЙ УНИВЕРСИТЕТ — НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР НАНОТЕХНОЛОГИЙ РАН

На правах рукописи

Диссертация допущена к защите
Зав. кафедрой

_____

“ ” _____ 2010 г.

ДИССЕРТАЦИЯ

НА СОИСКАНИЕ УЧЕНОЙ СТЕПЕНИ

МАГИСТРА

Тема: Нахождение эффективных булевых схем при помощи SAT-солверов

Направление: 010600.68 — Прикладные математика и физика

Магистерская программа: “Математические и информационные технологии”

Выполнил студент Г. Н. Ярославцев

(подпись)

Руководитель:
к. ф. - м. н., доцент Э. А. Гирш

(подпись)

Рецензент:
к. ф. - м. н. А. С. Куликов

(подпись)

Санкт-Петербург

2010 г.

Аннотация

В настоящей работе описаны результаты экспериментов по нахождению минимальных по числу гейтов булевых схем при помощи SAT-солверов. Представленные результаты содержат оптимальные схемы для некоторых функций с небольшим числом входов, а также блоки для построения эффективных схем с произвольным числом входов для некоторых функций из класса MOD_3^n. В частности, представлен блок, позволяющий построить схему размера $3n + O(1)c$ для функции MOD_3^n в полном бинарном базисе.

Благодарности

За многочисленные замечания и полезные советы я благодарен своему научному руководителю Эдуарду Алексеевичу Гиршу. Также я бы хотел поблагодарить за проделанную работу рецензента Александра Сергеевича Куликова и Ариста Александровича Кожевникова, в сотрудничестве с которым были получены результаты данной работы.

Работа над дипломным проектом велась при поддержке гранта Президента Российской Федерации МК-3912.2009.1, а также в рамках работ по второму этапу государственного контракта №265 от 23.07.2009.

Содержание

1 Введение

Задача нахождения оптимальных по числу элементов схем для реализации булевых функций давно привлекает интерес специалистов. Одним из первых классов схем, для которого были получены результаты в данной области, является класс контактных схем, описание которого можно найти в книге Р.Г. Нигматуллина "Сложность булевых функций"[1]. В работе Ю.Л. Васильева [2] были найдены минимальные контактные схемы для булевых функций четырех переменных. В работе В.Ю. Сусова [3] были предложены переборные алгоритмы для синтеза минимальных контактных схем.

В данной работе рассматривается задача нахождения оптимальных булевых схем из функциональных элементов над бинарными базисами. Для произвольной функции f соотношение между размером минимальных схем из контактных элементов и из функциональных элементов неизвестно [1]. Для краткости в дальнейшем изложении будем называть функциональные элементы просто гейтами, а схемы из функциональных элементов — просто булевыми схемами. Минимальное количество гейтов, необходимое для реализации функции будем называть схемной сложностью функции.

Задачу нахождения схемной сложности данной функции можно свести к решению задач распознавания следующего вида: "Существует ли схема размера s, вычисляющая функцию f?". Если функция $f_n : \{0,1\}^n \to \{0,1\}$ задана своей таблицей истинности из 2^n битов, то такая задача распознавания лежит в классе NP. В работе [4] показано, что из предположений о том, что эта задача лежит в P или $P/poly$, вытекают следствия, которые кажутся маловероятными. NP-полнота данной задачи не доказана.

Из мощностных соображений следует, что большинство булевых функций от n переменных $f_n\colon \{0,1\}^n \to \{0,1\}$ имеют сложность не менее, чем $2^n/n$ [5]. Однако, неизвестно ни одной явно заданной булевой функции, для которой была бы доказана суперлинейная оценка на схемную сложность. Лучшая из известных нижних оценок над полным бинарным базисом равняется $3n - o(n)$ и принадлежит Блюму [6].

В связи с тем, что нижние оценки на схемную сложность являются слабыми, важность изучения схемной сложности функций с небольшим количеством входов была подчеркнута Р. Вильямсом в обзорной статье [7]. Знание оптимальных схем для функций с небольшим количеством входов может помочь нам лучше понять структуру оптимальных схем для функций с произвольным количеством входов.

Сведение задачи оптимального дизайна логических схем к SAT было

предложено в работе [8]. В работе [9] были проведены эксперименты по построению оптимальных арифметических схем с использованием этого сведения и современных SAT-солверов. В данной работе используется сведение, которое похоже на предложенное в [8], но имеет ряд отличий, связанных с разницей в используемых моделях вычислений.

В данной работе представлены результаты экспериментов по нахождению эффективных булевых схем с помощью SAT-солверов. Особое внимание уделено схемной сложности MOD-функций, которые определяются следующим образом:

$$\mathrm{MOD}^n_{K,k}(x_1,\ldots,x_n) = 1 \iff \sum_{i=1}^{n} x_i \equiv k \pmod{K}$$

(параметры k и n могут быть опущены, если в них нет необходимости). Данные функции являются одними из простейших представителей класса симметрических булевых функций. Схемная сложность этих функций изучалась многими учеными. Однако, на сегодняшний день точная схемная сложность известна только для нескольких значений K. В таблице 1 приведены известные нижние и верхние оценки для MOD^n_K в различных моделях вычислений. Здесь под C и L имеются в виду схемная и формульная сложность соответственно; B_2 это полный бинарный базис, $U_2 = B_2 \setminus \{\oplus, \equiv\}$. Можно заметить, что для формул и схем в базисах U_2 и B_2 известно, что сложность MOD^n_K, $K = 3$ и $K = 5$, не меньше, чем сложность MOD^n_4. Однако, ни для одной из этих моделей не известно, что MOD^n_3 или MOD^n_5 строго сложнее, чем MOD^n_4.

MOD-функции могут быть вычислены индуктивно. Например, оптимальная схема размера $2.5n + O(1)$ для функции MOD^n_4, построенная Стокмайером [14], конструируется из блоков, состоящих из 10 гейтов, которые складывают 4 новые переменные с остатком по модулю 4, см. Рис. 1. При этом биты z_0, z_1 кодируют значение $\sum_{i=1}^{n} x_i \pmod 4$ следующим образом:

$$\sum_{i=1}^{n} x_i \pmod 4 = \begin{cases} 0, & (z_0, z_1) = (0,0), \\ 1, & (z_0, z_1) = (1,1), \\ 2, & (z_0, z_1) = (1,0), \\ 3, & (z_0, z_1) = (0,1). \end{cases}$$

Два выходных бита z'_0, z'_1 кодируют значение $\sum_{i=1}^{n+4} x_i \pmod 4$ таким же образом. Таким образом, можно доказывать верхние оценки на схемную сложность MOD-функций путем нахождения минимальных блоков константного размера.

Текст работы организован следующим образом:

K		L_{U_2}	L_{B_2}	C_{U_2}	C_{B_2}
2	lower	$\Theta(n^2)$ [10]	n	$3n - O(1)$ [11]	$n-1$
	upper				
3	lower	$\Omega(n^2)$ [10]	$\Omega(n \log n)$ [12]	$4n - O(1)$ [13]	$2.5n - O(1)$ [14]
	upper	$O(n^{2.58})$ [15]	$O(n^2)$ [16]	$7n + o(n)$ [17]	$5n + o(n)$ [17]
4	lower	$\Omega(n^2)$ [10]	$\Theta(n \log n)$ [12]	$4n - O(1)$ [13]	$2.5n - O(1)$ [14]
	upper	$O(n^2 \log^2 n)$[12]		$5n$ [13]	
≥ 5	lower	$\Omega(n^2)$ [10]	$\Omega(n \log n)$ [12]	$4n - O(1)$ [13]	$2.5n - O(1)$ [14]
	upper	$O(n^{4.57})$ [18]	$O(n^{3.13})$ [18]	$7n + o(n)$ [17]	$5n + o(n)$ [17]

Таблица 1: Известные нижние и верхние оценки на сложность MOD_K^n в различных моделях вычислений.

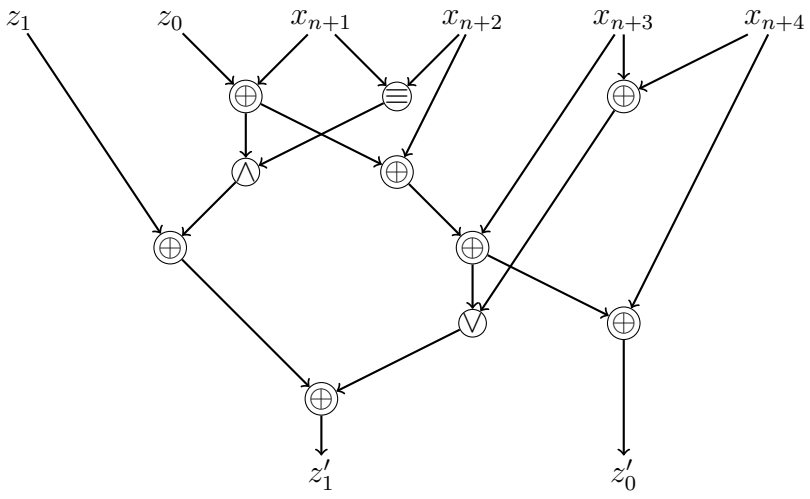


Рис. 1: Блок Стокмайера для функции MOD_4

- В разделе 2 даются необходимые определения,
- Раздел 3 посвящен описанию того, как факт существования схемы для конкретной функции кодируется в виде КНФ-формулы,
- В разделе 4 представлены новые верхние оценки, которые были доказаны автоматически.
- Раздел 5 посвящен описанию результатов о схемной сложности булевых функций с небольшим числом входов.

2 Определения

Пусть B_n это множество всех булевых функций $f\colon \{0,1\}^n \to \{0,1\}$. Функция $f \in B_n$ называется симметрической, если ее значение зависит только от суммы входов. А именно, должен существовать вектор $v \in \{0,1\}^{n+1}$ такой, что $f(x_1,\ldots,x_n) = v_s$, где $s = \sum_{i=1}^{n} x_i$ Типичной симметрической функцией является функция взятия остатка по модулю $\mathrm{MOD}^n_{K,k}$, которая определяется так:

$$\mathrm{MOD}^n_{K,k}(x_1,\ldots,x_n) = 1 \iff \sum_{i=1}^{n} x_i \equiv k \pmod{K}.$$

Схемой в базисе $A \subseteq B_2$ называется ориентированный ациклический граф с вершинами, имеющими входящую степень 0 или 2. Вершины со входящей степенью 0 помечены переменными из множества $\{x_1,\ldots,x_n\}$ и называются входами. Вершины со входящей степенью 2 помечены функциями из базиса A и называются гейтами. Также в схеме отдельно выделяют гейты, являющиеся выходами. Если выходов несколько, то на них задается порядок, так что схема с несколькими выходами вычисляет функцию $f\colon \{0,1\}^n \to \{0,1\}^m$. Размером схемы является количество гейтов в ней. В данной работе рассматриваются преимущественно схемы в полном бинарном базисе B_2.

Не умаляя общности, мы можем считать, что ни один из выходов схемы не вычисляет просто значение одной из переменных или ее отрицание. Поскольку такие выходы не влияют на схемную сложность, при ее вычислении они могут быть удалены.

Будем называть функцию $f \in B^n$ вырожденной, если она не зависит от некоторых своих аргументов, то есть существует такая переменная x_i, что подфункции $f|_{x_i=0}$ и $f|_{x_i=1}$ одинаковы. Легко видеть, что в схемах с одним выходом, состоящих более чем из одного гейта, гейт, вычисляющий вырожденную функцию от своих аргументов, может быть удален из схемы без увеличения ее размера (возможно, что для этого придется изменить функции, которые вычисляются в гейтах, от него зависящих). Например, гейт, вычисляющий отрицание, является вырожденным и может быть удален. В схемах с нескольким выходами может потребоваться вычисляющий вырожденную функцию гейт в том случае, если функция в одном из выходов является отрицанием функции в другом выходе. Если же это не так, то вычисляющие вырожденные функции гейты также могут быть удалены,. Множество B_2 содержит десять невырожденных функций:

- Восемь функций вида $((x \oplus a) \wedge (y \oplus b)) \oplus c$, где $a, b, c \in \{0,1\}$

- Две функции вида $x \oplus y \oplus a$, где $a \in \{0, 1\}$

3 Использование SAT-солверов для нахождения эффективных булевых схем

В этой части описаны детали сведения, которое используется при кодировке факта существования схемы для конкретной функции в виде КНФ. Сначала приводится описание общей конструкции сведения, которое похоже на описанное в [19] (где рассматриваются схемы в базисе U_2), а затем обсуждаем некоторые дополнительные особенности сведения, которые использовались для поиска схем для рассматриваемых нами функций.

3.1 Представление схем в виде КНФ

Пусть дана таблица истинности булевой функции $f\colon \{0,1\}^n \to \{0,1\}^m$ и нужно найти булеву схему в базисе $A \subseteq B_2$, которая вычисляет f и состоит из минимально возможного количества гейтов. Можно закодировать факт существования схемы из N гейтов, вычисляющей функцию f в виде КНФ, используя следующие пропозициональные переменные (входные аргументы пронумерованы индексами от 0 до $n-1$), а гейты — индексами от n до $n+N-1$ соответственно:

1. $t_{ib_1b_2}$ $(n \le i \le n+N-1,\ 0 \le b_1 \le 1,\ 0 \le b_2 \le 1)$ соответствует значению на выходе i-го гейта, если на первый вход ему подано значение b_1, а на второй — b_2. Таким образом, четыре переменные $t_{i00}, t_{i01}, t_{i10}, t_{i11}$ полностью определяют бинарную булеву функцию, вычисляемую i-ым гейтом. Всего переменных данного типа $O(N)$.

2. c_{ikj} $(n \le i \le n+N-1,\ 0 \le k \le 1,\ 0 \le j \le n+N-1)$ принимает значение истина, если значение в k-ый вход i-го гейта поступает из j-го гейта, в противном случае принимает значение ложь. Данные переменные полностью описывают структуры ориентированного графа, соответствующего схеме. Всего переменных данного типа $O(N^2)$.

3. o_{ij} $(n \le i \le n+N-1,\ 0 \le j \le m-1)$ принимает значение истина тогда и только тогда, когда j-ый выход схемы находится в i-ом гейте. Данные переменные полностью определяют положение выходов схемы. Всего переменных данного типа $O(Nm)$.

4. v_{it} $(0 \le i \le n+N-1,\ 0 \le t \le 2^n-1)$ соответствует значению на выходе i-го гейта, если входные аргументы имеют значения, задаваемые битовой маской t. Данные переменные используются для

того, чтобы описать тот факт, что значения, вычисляемые в выходах схемы совпадают со значениями из таблицы истинности при всех 2^n возможных значениях входных аргументов. Всего переменных данного типа $O(2^n N)$.

Следующие утверждения про схему записываются в виде дизъюнктов. Они полностью задают все требования к схеме, необходимые для того, чтобы она была построена корректно и вычисляла заданную функцию.

1. Бинарные функции, которые вычисляются в гейтах, принадлежат базису A.

2. Для всех (i, k) в точности одна из переменных c_{ikj} является истиной (k-ый вход i-го гейта подключен только к одному гейту). Это дает $O(N^3)$ 2-дизъюнктов и $O(N)$ $O(N)$-дизъюнктов.

3. Для всех j в точности одна из переменных o_{ij} является истиной (j-ый выход вычисляется в точности одним гейтом). Это дает $O(N^2 m)$ 2-дизъюнктов и $O(m)$ $O(N)$-дизъюнктов.

4. Для всех $0 \le i \le n-1$ и $0 \le t \le 2^n - 1$, v_{it} совпадает с соответствующим битом в t. Это дает $O(n \cdot 2^n)$ 1-дизъюнктов.

5. Для всех $n \le i \le n+N-1$ и $0 \le t \le 2^n - 1$, v_{it} равно значению, которое вычисляется i-ым гейтом в схеме, описываемой остальными переменными. Это дает $O(N^3 \cdot 2^n)$ 6-дизъюнктов и является частью сведения, которая приводит к появлению большинства дизъюнктов. Дизъюнкты данного типа записывают для всех $n \le i < n+N$, $n \le j_0 < i$, $j_0 < j_1 < i$, $0 \le i_0 < 2$, $0 \le i_1 < 2$, $0 \le r < 2^n$ и описывают следующее условие:

$$\neg c_{i_0 j_0} \vee \neg c_{i_1 j_1} \vee \neg(v_{j_0 r} = i_0) \vee \neg(v_{j_1 r} = i_1) \vee (v_{ir} = t_{i i_0 i_1}) .$$

Здесь первые два литерала позволяют найти те два гейта, которые являются входами i-го гейта, следующие два литерала нужны для нахождения значений этих гейтов на входном наборе r, а последнее условие необходимо для проверки того, что значение в i-ом гейте вычисляется правильно (на самом деле, так как описанное выше условие не является дизъюнктом, поскольку содержит равенство, то для его записи необходимо два похожих дизъюнкта для разбора случаев: обе переменные истинны или обе ложны):

$$\neg c_{i_0 j_0} \vee \neg c_{i_1 j_1} \vee \neg(v_{j_0 r} = i_0) \vee \neg(v_{j_1 r} = i_1) \vee \neg v_{ir} \vee t_{i i_0 i_1} .$$
$$\neg c_{i_0 j_0} \vee \neg c_{i_1 j_1} \vee \neg(v_{j_0 r} = i_0) \vee \neg(v_{j_1 r} = i_1) \vee v_{ir} \vee \neg t_{i i_0 i_1} .$$

6. Для всех входных значений аргументов значения выходов совпадают со значениями, заданными в таблице истинности. Это дает $O(N2^n m)$ 2-дизъюнктами. Дизъюнкты этого типа записываются для всех $0 \leq k \leq m-1$, $0 \leq r \leq 2^n - 1$, $n \leq i \leq n + N - 1$ и выглядят следующим образом:

$$\neg o_{ik} \vee (v_{ir} = value_{kr}),$$

где $value_{kr}$ это — значение k-го выхода на наборе входных значений r в соответствии с таблицей истинности.

Для сокращения перебора также можно считать без потери общности, что верны следующие утверждения, которые кодируются дополнительными дизъюнктами при сведении.

1. Оба входа каждого гейта вычисляются гейтами с меньшими номерами (то есть гейты топологически отсортированы в порядке, задаваемом нашей нумерацией).

2. Для каждого гейта номер его первого входа меньше номера второго.

3. Гейты не вычисляют вырожденные функции.

4. Хотя бы один из выходов находится в последнем гейте.

Во многих интересных для нас случаях формулы, получавшиеся описанным выше способом, оказывались слишком трудными для современных SAT-солверов. Например, формулы, кодирующие факт существования схемы, вычисляющей случайный предикат от 5 переменных за 9 гейтов уже достаточно трудны.

Это связано с тем, что количество схем экспоненциально растет с ростом числа гейтов. Грубая оценка сверху на число схем, вычисляющих предикаты от n аргументов и состоящих из N гейтов, дает их количество $(10(n+N+2)^2)^N$. Столько итераций потребуется наивному перебору для доказательство того, что не существует схемы данного размера для конкретной функции. В случае же, если схема существует, перебор может работать быстрее, особенно если схем данного размера много. При использовании солверов такой эффект тоже наблюдается — оптимальные схемы часто находятся гораздо быстрее, чем доказывается отсутствие схем меньшего размера.

Также в некоторых случаях, когда не удавалось за разумное время найти схему определенного размера, были использованы дополнительные ограничения пространства перебора. В отличие от приведенных выше условий эти ограничения ограничивают пространство перебора так,

что оно начинает включать уже не все схемы данного размера. Поэтому невыполнимость формул с этими ограничениями уже не говорит о том, что требуемые схемы не существуют. Однако, на практике за счет введения ограничений зачастую удается существенно сократить время работы солверов при поиске эффективных схем. Наиболее удачные ограничения приведены ниже.

1. Ограничение на исходящую степень гейтов.

2. Ограничение, описывающее существование ориентированного пути, проходящего через все гейты (для этого одним из входов i-го гейта должен быть $(i-1)$-ый гейт).

3.2 Кодировка остатков

В предыдущем пункте была рассмотрена ситуация, когда функция задается таблицей истинности. Заметим, что можно работать также с частично определенными функциями, а также с функциями, которые обладают некоторыми свойствами. Например, при поиске индуктивного блока для функций класса MOD вовсе не очевидно, какой выбор кодировки для остатка приведет к нахождению блока минимального размера. Таким образом, вместо задания таблицы истинности, удобнее записать тот факт, что блок осуществляет добавление новых переменных к остатку, закодированному некоторым образом.

Предположим, что нужно найти блок для функции MOD_K. Такой блок складывает несколько новых переменных с остатком по модулю K, который как-то закодирован. Остаток по модулю K может быть закодирован в $\lceil \log_2 K \rceil$ битах. Поскольку кодировку неизвестна, то введем новые переменные e_{ij}, где e_{ij} истинно тогда и только тогда, когда битовое представление $0 \leq j < 2^{\lceil \log_2 K \rceil}$ кодирует остаток $0 \leq i \leq K-1$. Таким образом, каждый остаток может кодироваться несколькими значениями j (такая возможность оказалось ключевой при поиске оптимального блока для функции MOD_3).

Кроме очевидных дизъюнктов, описывающих то, что каждый остаток i кодируется хотя бы одной кодировкой j и что каждая кодировка j используется ровно для одного остатка i, добавляется также следующее условие. Для всех возможных входных наборов переменных, всех возможных сумм битов s, которые данный блок добавляет к закодированному остатку, и всех возможных значений закодированных остатков i (то есть соответствующая переменная e_{ij} истинна), остаток на выходе должен быть равен $i + s$ (фактически же в виде КНФ на самом деле за-

писывается тот факт, что значения выходов схемы не равны j' для всех j' таких, что $e_{i'j'}$ истинно для некоторого $i' \not\equiv i + s \pmod{K}$)

В качестве примера предположим, что нужно найти блок, который получает на вход остаток t по модулю 3, который некоторым образом закодирован в виде двух битов (z_0, z_1), и новую переменную x_n и выдает два бита (z'_0, z'_1), которые кодируют в той же самой кодировке $t + x_n \pmod 3$. Тогда переменная e_{23} истинна тогда и только тогда, когда из того, что $(z_0, z_1) = (1, 1)$ следует, что $t = 2$, а переменная e_{11} истинна тогда и только тогда, когда из того, что $(z_0, z_1) = (0, 1)$ следует, что $t = 1$. Описанное условие тогда записывается в виде следующего выражения:

$$(e_{23} \wedge z_0 \wedge z_1 \wedge x_n \wedge e_{11}) \Rightarrow (z'_0 \vee \neg z'_1)\,.$$

Такой автоматический поиск кодировки остатка оказался весьма полезным, поскольку только с его использованием Удалось найти эффективный блок, из которого следует верхняя оценка $5.5n+O(1)$ для $C_{U_2}(\mathrm{MOD}_3^n)$. Однако, понятно, что поиск блока с произвольной кодировкой представляет из себя более трудную задачу в общем случае.

4 Верхние оценки для функции MOD_3

В данном разделе описываются верхние оценки для функции MOD_3^n в базисах B_2 и U_2, полученные при помощи описанного выше сведения к задаче SAT с использованием SAT-солверов. Верхние оценки получаются путем построения схемы из блоков, изображенных на рис. 2 и рис. 3. Каждый из блоков получает на вход значение $\sum_{i=1}^{n} x_i \pmod 3$, закодированное парой битов (z_1, z_2) и несколько новых переменных (три и две соответственно). Выходом блока является пара битов (z'_1, z'_2), кодирующая значения $\sum_{i=1}^{n+3} x_i \pmod 3$ и $\sum_{i=1}^{n+2} x_i \pmod 3$ соответственно. В блоках используются следующие кодировки остатков:

$$\sum_{i=1}^{n} x_i \pmod 3 = \begin{cases} 0, & (z_0, z_1) = (0, 0), \\ 1, & (z_0, z_1) = (0, 1, \\ 2, & z_0 = 1, \end{cases}$$

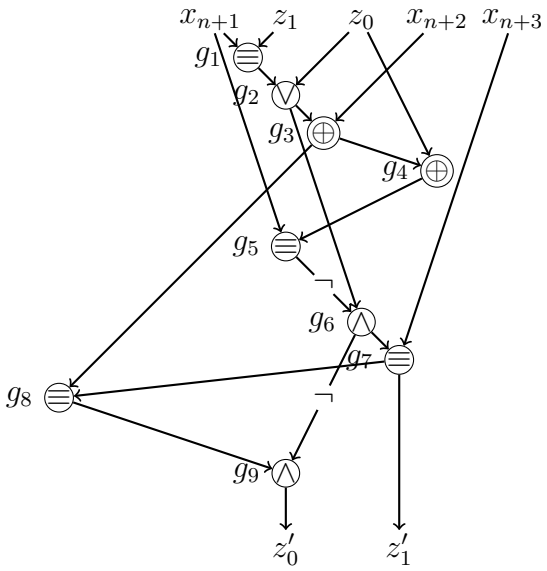
и

$$\sum_{i=1}^{k} x_i \pmod 3 = \begin{cases} 0, & z_0 = 0, \\ 1, & (z_0, z_1) = (1, 0), \\ 2, & (z_0, z_1) = (1, 1). \end{cases}$$

Верхние оценки $C_{B_2}(\mathrm{MOD}_3^n) \leq 3n + O(1)$ и $C_{U_2}(\mathrm{MOD}_3^n) \leq 5.5n + O(1)$ следуют из существования блоков непосредственно.

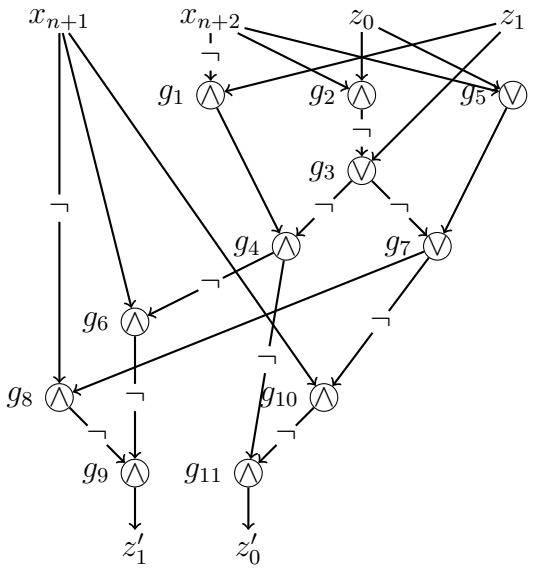
Данные блоки были найдены в результате длинной последовательности экспериментов с различными ограничениями из числа описанных выше, поскольку без ограничений формулы оказывались слишком трудными для солверов (например, формула для поиска описанного выше индуктивного блока без ограничений после упрощений имеет 470 переменных и 87201 дизъюнкт). Поскольку блоки были найдены с использованием ограничений, то неизвестно, являются ли они оптимальными, поэтому улучшение верхних оценок с использованием таких же конструкций возможно, хотя и представляется маловероятным.

В таблице 2 также приводятся размеры оптимальных схем в базисе B_2 для $\mathrm{MOD}_{3,k}^n$ для различных значений n и k. $C_{B_2}(\mathrm{MOD}_{3,2}^5) \leq 10$ означает, что была найдена схема размера 10, но доказать невыполнимость формулы, описывающей существование схемы (без ограничений) размера 9, не удалось. Сами оптимальные схемы приведены на рис. 4



x_{n+1}	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
x_{n+2}	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
x_{n+3}	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
z_0	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
z_1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
g_1	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
g_2	1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1
g_3	1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0
g_4	1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1
g_5	0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1
g_6	1 0 0 0 1 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0
g_7	0 1 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 0
g_8	0 0 0 1 1 1 1 0 1 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1
g_9	0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1
z'_0	0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1
z'_1	0 1 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 0

Рис. 2: Индуктивный блок для функции MOD_3 в базисе B_2 и его таблица истинности.



x_{n+1}	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
x_{n+2}	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
z_0	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
z_1	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
g_1	0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0
g_2	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
g_3	1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1
g_4	1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 1
g_5	0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1
g_6	1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
g_7	1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1
g_8	0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
g_9	0 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1
g_{10}	1 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0
g_{11}	0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0
z'_0	0 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1
z'_1	0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0

Рис. 3: Индуктивный блок для функции MOD_3 в базисе U_2 и его таблица истинности.

	$n=3$	$n=4$	$n=5$
$k=0$	3	7	≤ 10
$k=1$	4	7	≤ 9
$k=2$	4	6	≤ 10

Таблица 2: Размеры оптимальных схем в базисе B_2 для $\mathrm{MOD}^n_{3,k}$

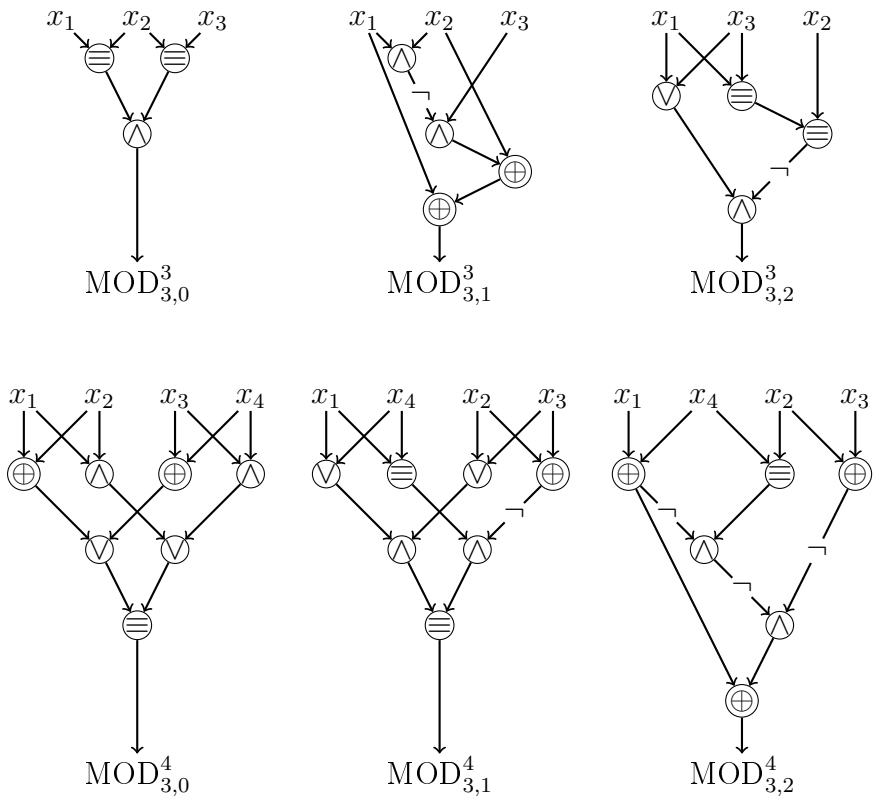


Рис. 4: Оптимальные схемы для функции $\mathrm{MOD}^n_{3,k}$ для $n = 3, 4$

5 Схемная сложность булевых функций с небольшим числом входов

5.1 Схемная сложность булевых функций от четырех переменных

Как показано в работе [20], существует 402 различных типа булевых функций от четырех переменных. При этом функции принадлежат одному типу, если можно добиться их совпадения путем перестановки переменных и заменой некоторых переменных на их логические отрицания. Достаточно найти схемную сложность одной функции каждого из типов, поскольку схемы для функций одного типа имеют одинаковую структуру и отличаются только перестановкой переменных и наличием или отсутствием отрицаний перед входами. В статье [2] предъявлены оптимальные контактные схемы для всех различных типов булевых функций от четырех переменных и доказана оптимальность этих схем.

Путем использования описанных выше методов были найдены оптимальные булевы схемы для всех различных функций от четырех переменных, доказав их оптимальность. Таким образом было получено значение функции Шеннона $C(n) = \max\limits_{f \in B_2^n} C_{B_2}(f)$ для $n = 4$, $C(4) = 7$. В книге [21] доказано, что $C(3) = 4$. Из предыдущего раздела видно, что максимум $C(n)$ при $n = 3, 4$ достигается в частности на функции $\mathrm{MOD}_{3,1}^n$. При $n = 4$ максимум $C(n)$ достигается на функциях пяти различных типов. Представители данных типов и оптимальные схемы для них приведены ниже.

5.2 Схемная сложность симметрических булевых функций от пяти переменных

Существует всего $1,228,158$ различных типов булевых функций от пяти переменных, как показано в [22], поэтому изучение схемной сложности всех булевых функций от пяти переменных требует методов отличных от предложенных, поскольку нахождение по отдельности оптимальных схем для представителей каждого типа при помощи SAT-солверов занимает слишком много времени.

Однако, некоторые важные классы булевых функций содержат не так много функций от пяти переменных. Например, в классе симметрических функций от n переменных S_n всего 2^{n+1} различная функция, (напомним, что каждая такая функция задается вектором $v \in \{0,1\}^{n+1}$ таким, что $f(x_1, \ldots, x_n) = v_s$, где $s = \sum_{i=1}^n x_i$). При рассмотрении схем-

ной сложности можно считать, что различных типов симметрических функций из класса S_n на самом деле 2^n, поскольку они разбиваются на пары одинаковых с точностью до отрицания, так что можно не умаляя общности полагать $v_{n+1} = 0$. Также можно считать, что симметрические функции, задаваемые векторами v и v', где v' получается из v перестановкой элементов вектора в обратном порядке, имеют один тип, поскольку схемы для этих функций можно переделать одну в другую путем добавления отрицаний ко всем переменным.

Ниже приведена таблица верхних и нижних оценок на схемную сложность различных типов симметрических функций от пяти переменных (всего имеется 19 типов). Для функций приведены элементы вектора $v = [v_4, \ldots, v_0]$ ($v_5 = 0$), а также формулы. Несовпадение нижних и верхних оценок связано с тем, что полученные в результате сведения к SAT формулы не удалось решить за несколько дней.

Функция	$v = [v_4, \ldots, v_0]$	Формула	Верхняя оценка	Нижняя оценка
s_1	[0, 0, 0, 0, 1]	$\bigwedge_i \neg x_i$	4 10	4
s_2	[0, 0, 0, 1, 0]	$\sum_i x_i = 1$	10 11	9
s_3	[0, 0, 0, 1, 1]	$\sum_i x_i \leq 1$	10 12	9
s_4	[0, 0, 1, 0, 0]	$\sum_i x_i = 2$	9 13	9
s_5	[0, 0, 1, 0, 1]	$\sum_i x_i = 0 \vee \sum_i x_i = 2$	10 14	9
s_6	[0, 0, 1, 1, 0]	$\sum_i x_i = 1 \vee \sum_i x_i = 2$	10 15	9
s_7	[0, 0, 1, 1, 1]	$\sum_i x_i \leq 2$	9 16	9
s_8	[0, 1, 0, 0, 1]	$\mathrm{MOD}_{3,0}$	10 17	9
s_9	[0, 1, 0, 1, 0]	$\sum_i x_i = 1 \vee \sum_i x_i = 3$	8 18	8
s_{10}	[0, 1, 0, 1, 1]	$\sum_i x_i = 0 \vee \sum_i x_i = 1 \vee \sum_i x_i = 3$	9 19	9
s_{11}	[0, 1, 1, 0, 0]	$\mathrm{MOD}_{4,2} \vee \mathrm{MOD}_{4,3}$	8 20	8
s_{12}	[0, 1, 1, 0, 1]	$\sum_i x_i = 0 \vee \sum_i x_i = 2 \vee \sum_i x_i = 3$	10 21	9
s_{13}	[0, 1, 1, 1, 0]	$\sum_i x_i = 1 \vee \sum_i x_i = 2 \vee \sum_i x_i = 3$	10 22	9
s_{14}	[1, 0, 0, 0, 1]	$\mathrm{MOD}_{4,0}$	9 23	9
s_{15}	[1, 0, 0, 1, 0]	$\mathrm{MOD}_{3,1}$	9 24	9
s_{16}	[1, 0, 1, 0, 1]	$\mathrm{MOD}_{2,0}$	4 25	4
s_{17}	[1, 0, 1, 1, 0]	$\sum_i x_i = 1 \vee \sum_i x_i = 2 \vee \sum_i x_i = 4$	11 26	9
s_{18}	[1, 1, 0, 0, 1]	$\sum_i x_i = 0 \vee \sum_i x_i = 3 \vee \sum_i x_i = 4$	9 27	9
s_{19}	[1, 1, 1, 1, 0]	$\neg\mathrm{MOD}_{5,0}$	7 28	7

При нахождении нижних оценок помимо дизъюнктов, описывающих основную конструкцию сведения, использовались также дополнительные условия. Поскольку непонятно, как можно коротко сформулировать в виде формулы в КНФ от переменных сведения необходимое и достаточное условие того, что схема вычисляет симметрическую функцию, использовалось условие, которое является только необходимым. Это условие не позволяет ограничить перебор только схемами для симметрических функций, однако позволяет существенно сократить его.

Поскольку известно, что функция является симметрической, то все ее входные переменные эквивалентны, то есть их можно переставлять в произвольном порядке. Тогда можно сформулировать необходимое условие на схему, вычисляющую симметрическую функцию так: если расположить гейты в топологическом порядке, то при рассмотрении их в этом порядке, если гейт принимает на вход переменную, которая до этого не

использовалась другими гейтами, то эта переменная имеет минимальный номер среди всех неиспользованных пока предыдущими гейтами переменных. Например, первый гейт принимает на вход переменные x_1 и x_2 и т. д.

5.3 Схемная сложность перестановок с небольшим числом входов

Перестановкой называется обратимая функция $f\colon \{0,1\}^n \to \{0,1\}^n$. Важную роль в криптографии играют перестановки, для которых $C(f^{-1}) > C(f)$. В работе Хильтгена [23] было построено первое семейство так называемых односторонних в слабом смысле перестановок — перестановок, для которых $\frac{C(f^{-1})}{C(f)} = k$, при $k = 2$. До сих пор неизвестно семейство односторонних в слабом смысле перестановок для $k > 2$.

Всего существует $2^n!$ перестановок от n переменных. Как и в случае с функциями с одним выходом, функции с несколькими выходами могут быть разбиты на типы так, что функции одного типа не отличаются с точностью до перестановки переменных и расстановки отрицаний перед ними. В случае с функциями с несколькими выходами можно также считать, что одному типу принадлежат функции, которые могут быть получены друг из друга перестановкой выходов и расстановкой отрицаний перед ними.

Тем самым получается, что существует не менее $\frac{2^n!}{n!^2 2^{2n}}$ различных типов перестановок от n переменных. При $n = 4$ получается, что различных типов хотя бы $1.4 * 10^8$, что делает поиск оптимальных схем для всех типов перестановок слишком трудоемким для применения описанных методов.

Поэтому в данной работе схемная сложность перестановок была изучена только для случая $n = 3$. Автоматически были найдены все перестановки от 3 переменных, количество которых равно 118 (при этом не учитывались перестановки, у которых один из выходов дает значение переменной или ее отрицания, поскольку нахождение схемной сложности таких перестановок сводится к случаю $n = 2$, который интереса не представляет). В слайдах [24] говорится, что в данном случае $C(f^{-1}) = C(f)$. Данный факт был в работе проверен. Также была найдено значение функции Шеннона для перестановок при $n = 3$, $C(3) = 7$. Максимум достигается на перестановках 14 различных типов.

Список литературы

[1] Р. Г. Нигматуллин. *Сложность булевых функций.* Издательство Казанского Университета, 1983.

[2] Ю. Л. Васильев. Минимальные контактные схемы для булевых функций четырех переменных. *Доклады Академии наук СССР*, 127(2):242–245, 1959.

[3] В. Ю. Сусов. Два алгоритма переборного типа для синтеза минимальных контактных схем и их реализация. Дипломная работа, МГУ им. М. В. Ломоносова, 1981.

[4] Valentine Kabanets and Jin yi Cai. Circuit minimization problem. *Electronic Colloquium on Computational Complexity (ECCC)*, (45), 1999.

[5] C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell systems technical journal*, 28(1):59–98, 1949.

[6] Norbert Blum. A boolean function requiring 3n network size. *Theor. Comput. Sci.*, 28:337–345, 1984.

[7] Ryan Williams. Applying practice to theory. *CoRR*, abs/0811.1305, 2008.

[8] A. P. Kamath, N. K. Karmarkar, K. G. Ramakrishnan, and M. G. C. Resende. An interior point approach to boolean vector function synthesis. In *Proceedings of the 36th International Midwest Symposium on Circuits and Systems (MSCAS'93)*, pages 185–189, 1993.

[9] Giovani Gomez Estrada. A note on designing logical circuits using SAT. In *Proceedings of the 5th International Conference on on Evolvable Systems (ICES'03)*, volume 2606 of *Lecture Notes in Computer Science*, pages 410–421, 2003.

[10] V. M. Khrapchenko. Complexity of the realization of a linear function in the case of Π-circuits. *Math. Notes Acad. Sciences*, 9:21–23, 1971.

[11] C. Schnorr. Zwei lineare untere Schranken für die Komplexität Boolescher Funktionen. *Computing*, 13:155–171, 1974.

[12] Michael J. Fischer, Albert R. Meyer, and Michael S. Paterson. $\Omega(n \log n)$ lower bounds on length of Boolean formulas. *SIAM Journal on Computing*, 11:416–427, 1982.

[13] Uri Zwick. A $4n$ lower bound on the combinational complexity of certain symmetric boolean functions over the basis of unate dyadic Boolean functions. *SIAM Journal on Computing*, 20:499–505, 1991.

[14] Larry J. Stockmeyer. On the combinational complexity of certain symmetric Boolean functions. *Mathematical Systems Theory*, 10:323–336, 1977.

[15] Andrew Chin. On the depth complexity of the counting functions. *Information Processing Letters*, 35:325–328, 1990.

[16] D. C. van Leijenhorst. A note on the formula size of the "mod k" functions. *Information Processing Letters*, 24:223–224, 1987.

[17] Roshal G. Nigmatullin. *Slognost' bulevikh funktsii.* Moskva, Nauka, 1991. In Russian.

[18] Michael S. Paterson and Uri Zwick. Shallow circuits and concise formulae for multiple addition and multiplication. *Computational Complexity*, 3:262–291, 1993.

[19] Niklas Eén. Practical SAT — a tutorial on applied satisfiability solving. Slides of invited talk at FMCAD, 2007.

[20] George Polya. Sur les types des propositions composées. *The Journal of Symbolic Logic*, 5(3):98–102, 1940.

[21] А. В. Чашкин. *Лекции по дискретной математике.* МГУ им. Ломоносова, 2007.

[22] David Slepian. On the number of symmetry types of boolean functions of n variables. *Canadian Journal of Mathematics*, 5:185–193, 1953.

[23] Alain P. Hiltgen. Constructions of freebly-one-way families of permutations. In *AUSCRYPT*, pages 422–434, 1992.

[24] J Massey. The difficulty with difficulty a guide to the transparencies from the eurocrypt'96 iacr distinguished lecture, 1996.

6 Приложение

6.1 Наиболее сложные функции от четырех переменных

x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	f_5
0	0	0	0	0	1	0	1	1
0	0	0	1	1	1	1	0	1
0	0	1	0	1	1	0	1	1
0	0	1	1	0	0	1	1	1
0	1	0	0	1	1	1	1	1
0	1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1
1	0	0	0	1	1	1	1	1
1	0	0	1	0	0	0	1	1
1	0	1	0	0	0	1	0	0
1	0	1	1	0	0	1	1	1
1	1	0	0	0	0	1	0	0
1	1	0	1	0	0	1	1	1
1	1	1	0	0	0	0	1	1
1	1	1	1	0	0	0	0	0

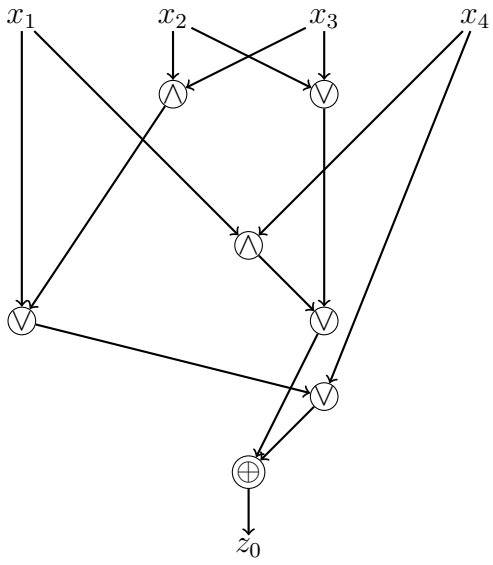


Рис. 5: Схема для функции f_1

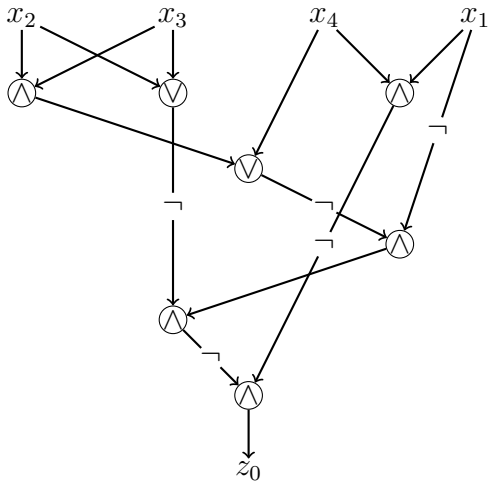


Рис. 6: Схема для функции f_2

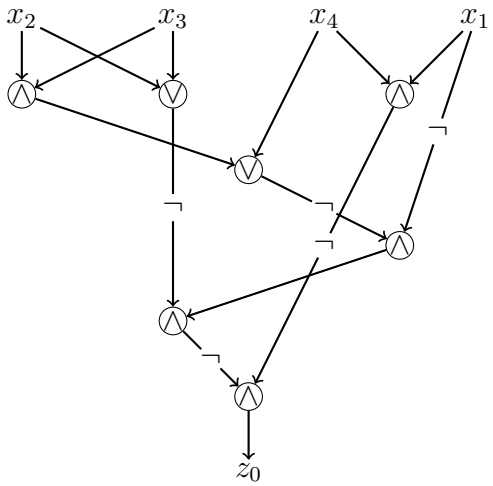


Рис. 7: Схема для функции f_3

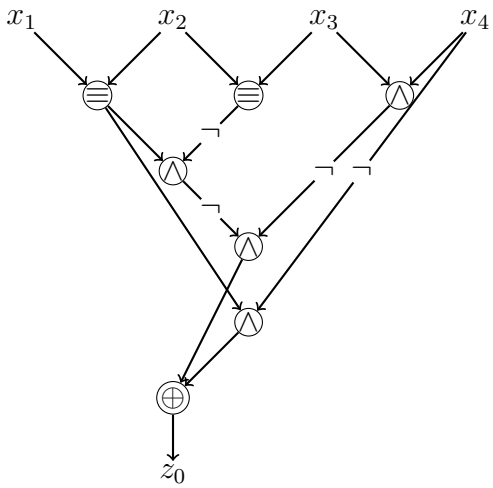


Рис. 8: Схема для функции f_4

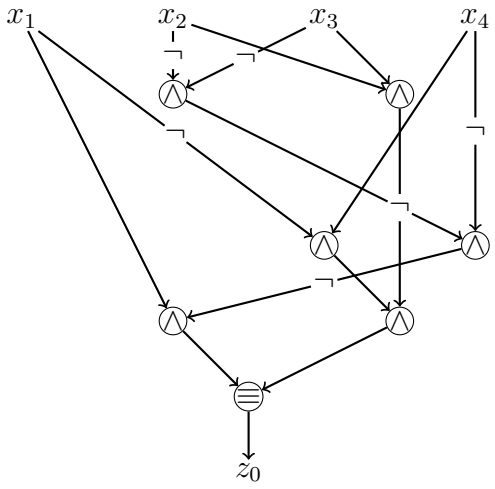


Рис. 9: Схема для функции f_5

6.2 Схемы для симметрических функций от пяти переменных

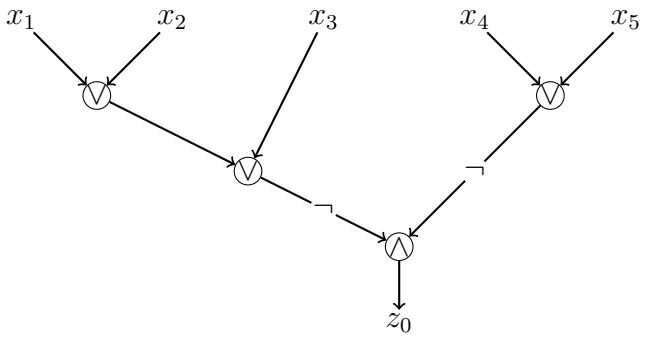


Рис. 10: Схема для функции s_1

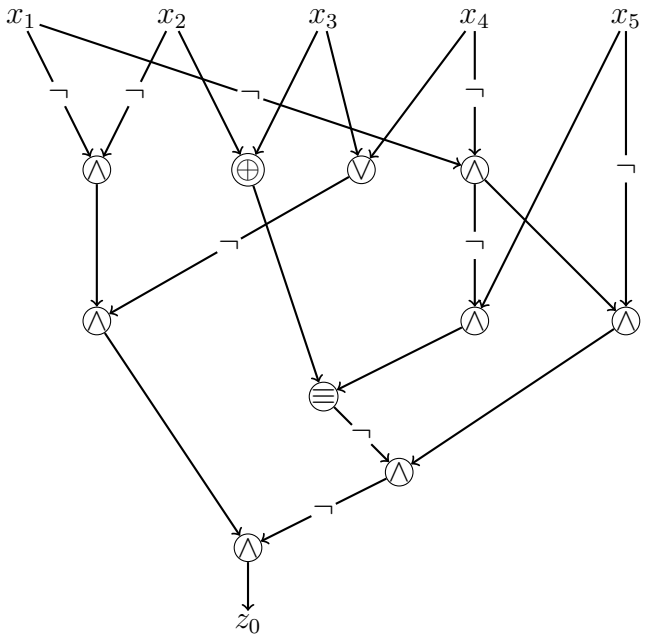


Рис. 11: Схема для функции s_2

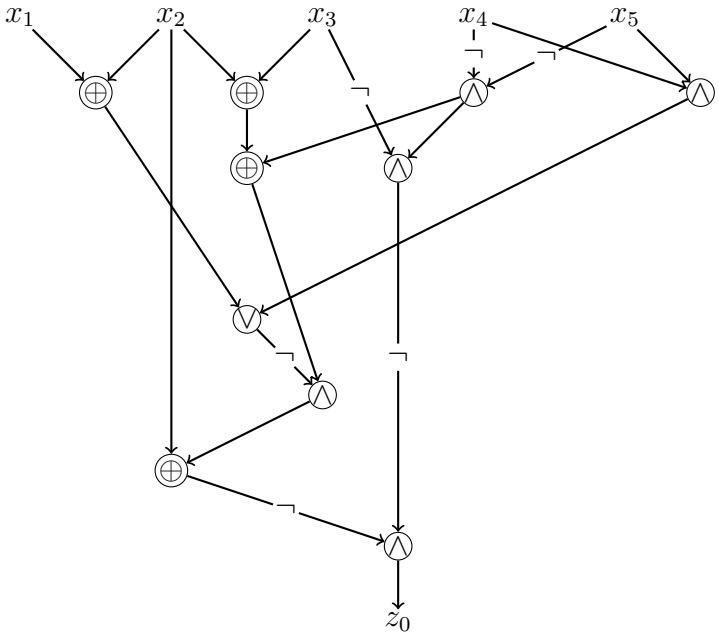


Рис. 12: Схема для функции s_3

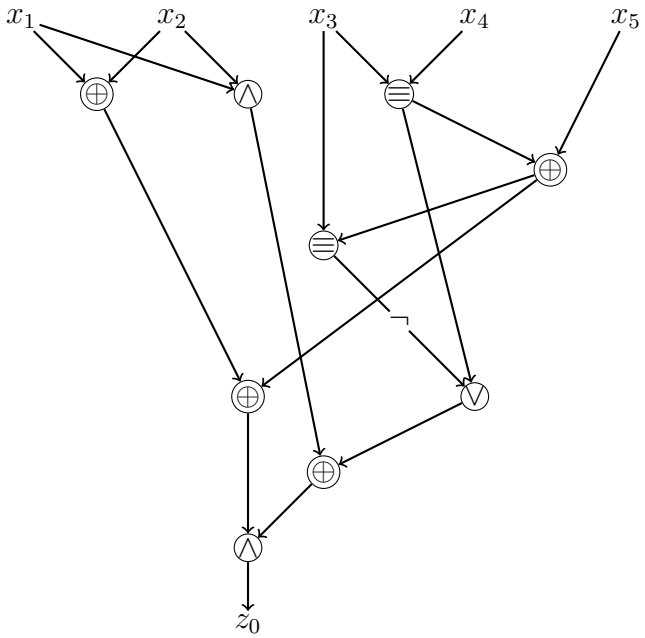


Рис. 13: Схема для функции s_4

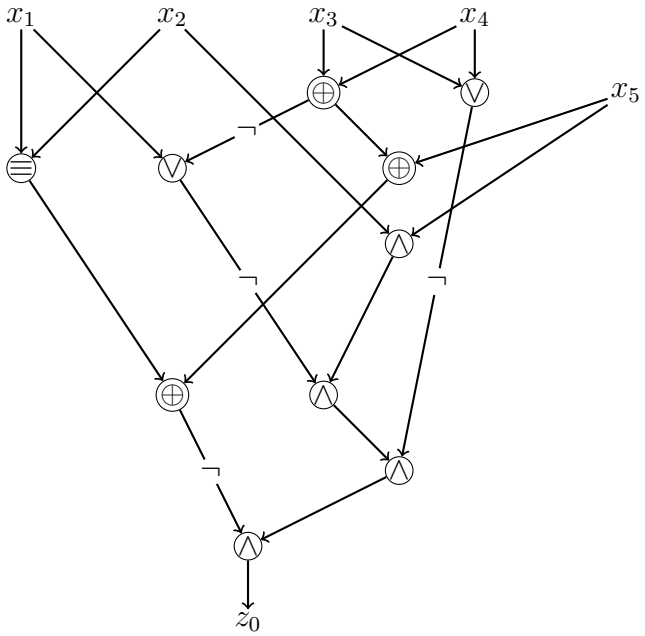


Рис. 14: Схема для функции s_5

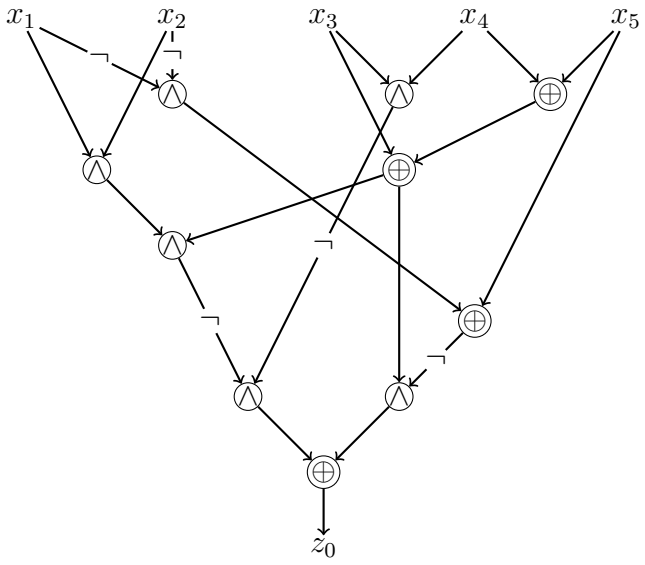


Рис. 15: Схема для функции s_6

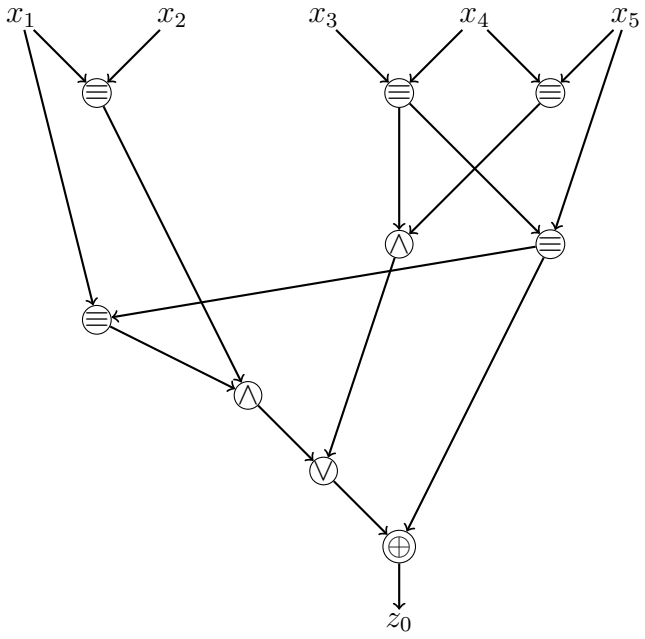


Рис. 16: Схема для функции s_7

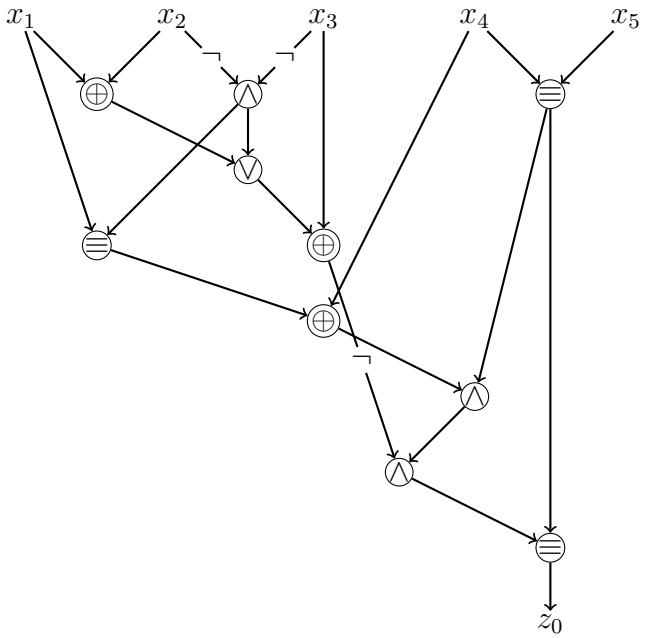


Рис. 17: Схема для функции s_8

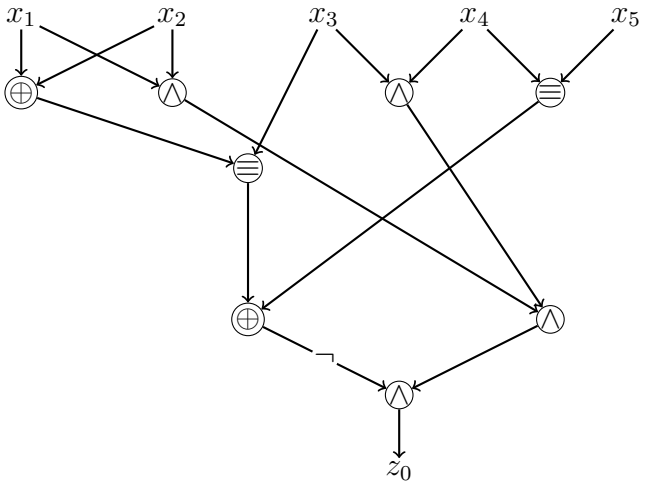


Рис. 18: Схема для функции s_9

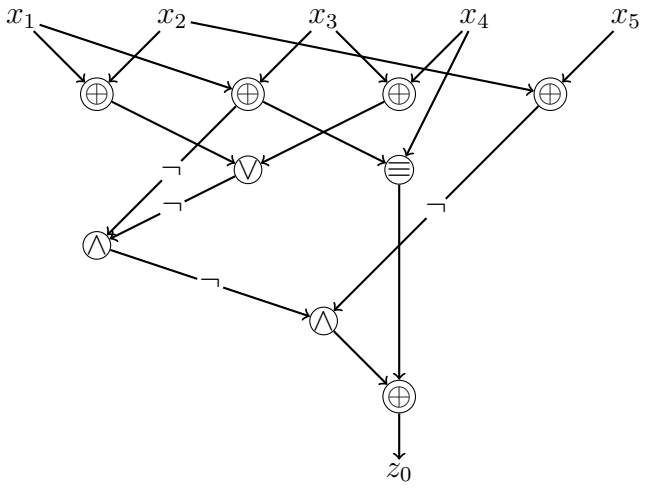


Рис. 19: Схема для функции s_{10}

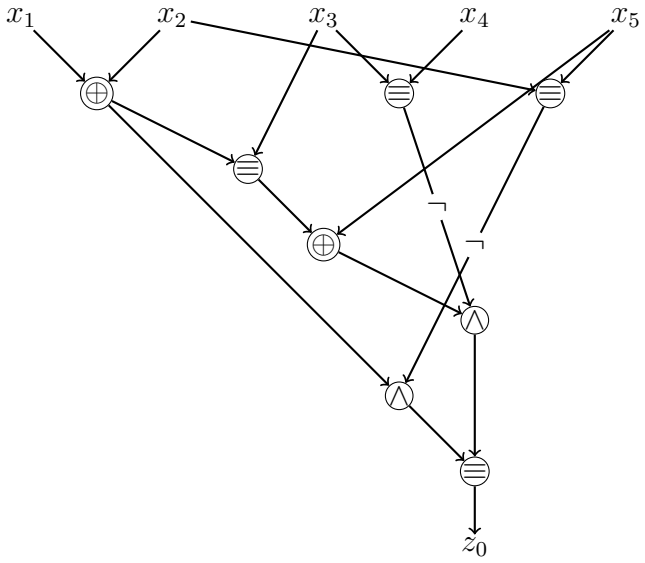


Рис. 20: Схема для функции s_{11}

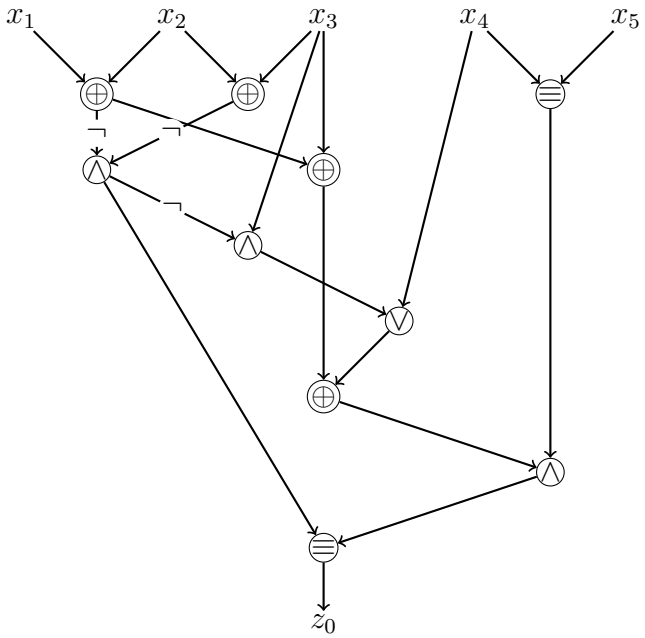


Рис. 21: Схема для функции s_{12}

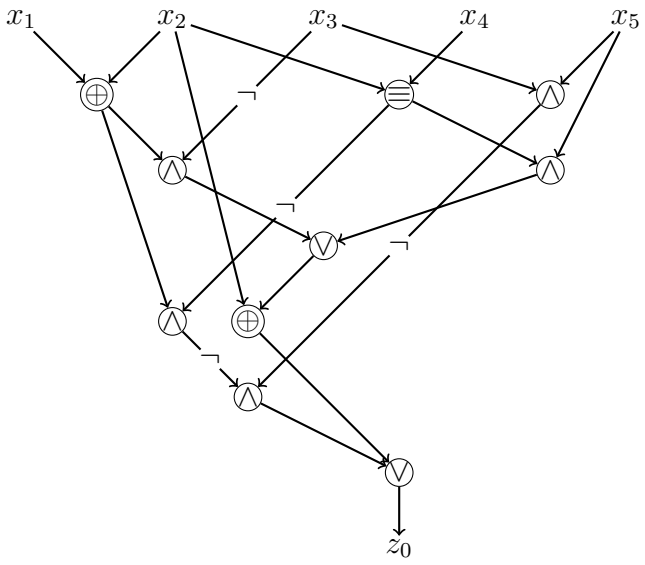


Рис. 22: Схема для функции s_{13}

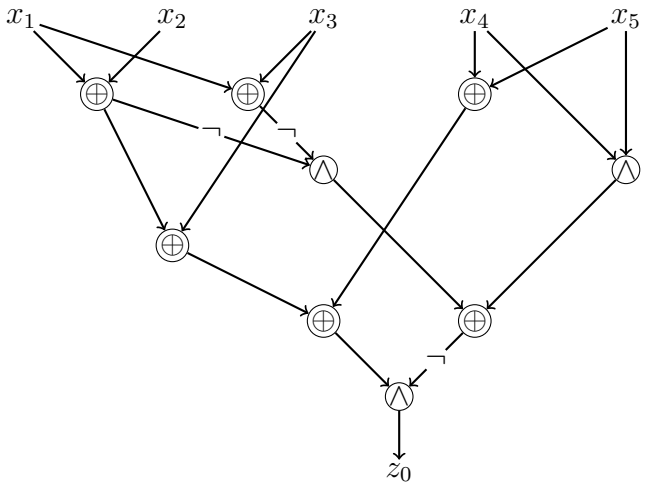


Рис. 23: Схема для функции s_{14}

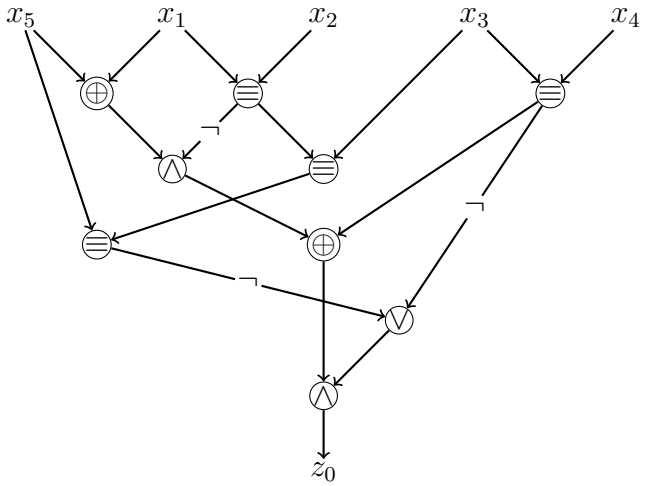


Рис. 24: Схема для функции s_{15}

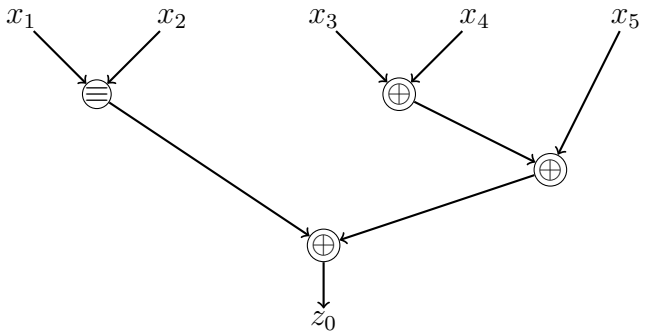


Рис. 25: Схема для функции s_{16}

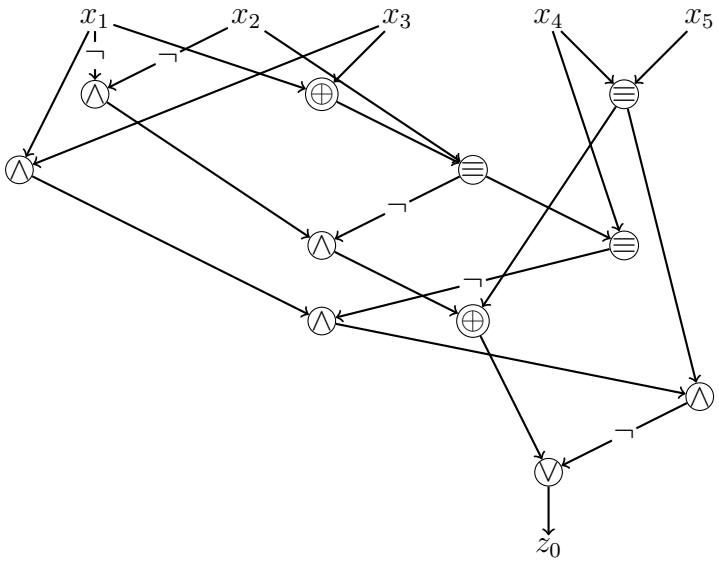


Рис. 26: Схема для функции s_{17}

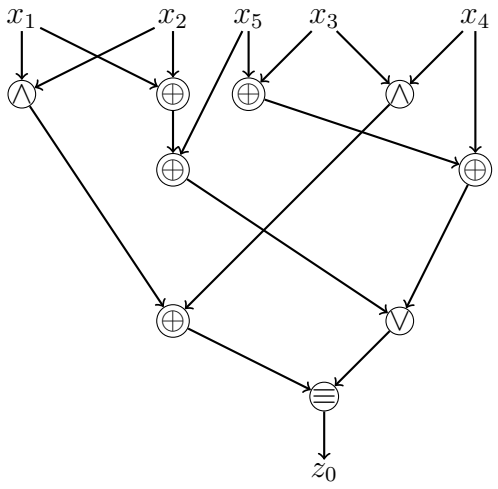


Рис. 27: Схема для функции s_{18}

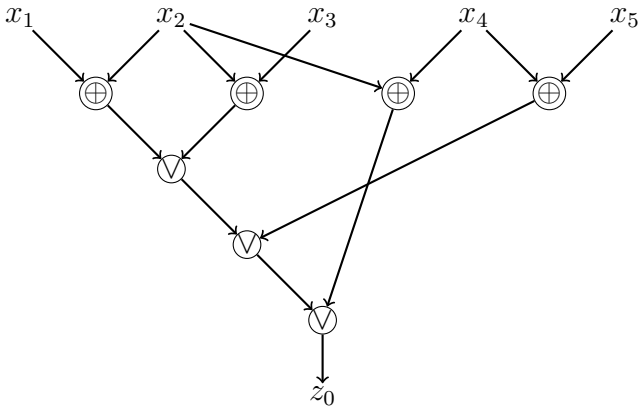


Рис. 28: Схема для функции s_{19}