



ФАКУЛЬТЕТ  
ВЫЧИСЛИТЕЛЬНОЙ  
МАТЕМАТИКИ И  
КИБЕРНЕТИКИ  
МГУ ИМЕНИ  
М.В. ЛОМОНОСОВА

*teach-in*  
ЛЕКЦИИ УЧЕНЫХ МГУ

# ОСНОВЫ КИБЕРНЕТИКИ

ЛОЖКИН  
СЕРГЕЙ АНДРЕЕВИЧ

ВМК МГУ

КОНСПЕКТ ПОДГОТОВЛЕН  
СТУДЕНТАМИ, НЕ ПРОХОДИЛ  
ПРОФ. РЕДАКТУРУ И МОЖЕТ  
СОДЕРЖАТЬ ОШИБКИ.  
СЛЕДИТЕ ЗА ОБНОВЛЕНИЯМИ  
НА [VK.COM/TEACHINMSU](https://vk.com/teachinmsu).

ЕСЛИ ВЫ ОБНАРУЖИЛИ  
ОШИБКИ ИЛИ ОПЕЧАТКИ,  
ТО СООБЩИТЕ ОБ ЭТОМ,  
НАПИСАВ СООБЩЕСТВУ  
[VK.COM/TEACHINMSU](https://vk.com/teachinmsu).

## Содержание

<b>1</b>	<b>Лекция 1. ДНФ функции алгебры логики</b>	<b>10</b>
1.1	Введение: о кибернетике, системах управления, курсе, его разделах и сферах применения . . . . .	10
1.2	Минимизация ДНФ и связанные с ней задачи . . . . .	11
1.3	Отношения полного/линейного и частичного монотонного порядков. Отношение соседства. Вес набора . . . . .	12
1.4	Геометрическая интерпретация: рёбра, слои, грани (их представления, мощность и количество в кубе) . . . . .	13
1.5	Определение функции алгебры логики (ФАЛ) . . . . .	15
1.6	Существенная и несущественная зависимости ФАЛ от переменных. Равенство ФАЛ. Множества ФАЛ . . . . .	16
1.7	Примеры со стандартными ФАЛ . . . . .	17
1.8	Представление ФАЛ дизъюнктивными нормальными формами (ДНФ)	18
<b>2</b>	<b>Лекция 2. Сокращённая ДНФ</b>	<b>22</b>
2.1	Конъюнктивная нормальная форма (КНФ). Совершенная КНФ (СКНФ) . . . . .	23
2.2	Пример построения СКНФ. Сравнение с СДНФ . . . . .	25
2.3	Некоторые тождества алгебры логики, необходимые в дальнейшем	25
2.4	Импликанта. Простая импликанта . . . . .	27
2.5	Утверждение о построении сокращённой ДНФ и его следствие . .	28
2.6	Тождество обобщённого склеивания. Расширение и строгое расширение ДНФ . . . . .	29
2.7	Метод Блейка построения сокращённой ДНФ и его следствие . . .	29
<b>3</b>	<b>Лекция 3. Особенности сокращённых ДНФ</b>	<b>32</b>
3.1	Окрестность грани, степень локальности . . . . .	32
3.2	Локальность критериев вхождения простой импликанты . . . . .	32
3.3	Задача данной лекции. Линейные ФАЛ и их свойства . . . . .	33
3.4	Функции, монотонно зависящие от переменной и их свойство . . .	35
3.5	Монотонные ФАЛ, особенность их простых импликант. Имонотонность ФАЛ . . . . .	36

3.6	Особенности сокращённых ДНФ монотонных ФАЛ . . . . .	36
3.7	Покрытие, задача покрытия, функция покрытия и её свойства . . . . .	39
3.8	Построение всех тупиковых ДНФ . . . . .	41
<b>4</b>	<b>Лекция 4. Задача минимизации ДНФ</b>	<b>43</b>
4.1	Функционал сложности ДНФ. Примеры . . . . .	43
4.2	Пси-сложность, Пси-оптимальность . . . . .	43
4.3	Задача минимизации ДНФ в общем виде. Общая схема её решения. Пример . . . . .	44
4.4	Функция Шеннона. Примеры для длины и ранга, их точные значения. Разложение Шеннона . . . . .	44
4.5	Отображение набора $k$ -мерного единичного куба на набор независимых случайных величин . . . . .	47
<b>5</b>	<b>Лекция 5. Сложность задачи минимизации ДНФ</b>	<b>49</b>
5.1	Неравенства для длины и ранга и их доказательства. Эффект Шеннона . . . . .	51
5.2	Параметры трудоёмкости задачи минимизации ДНФ . . . . .	52
5.3	Утверждение о числе тупиковых (минимальных) ДНФ и его следствие . . . . .	53
5.4	Нижняя оценка длины сокращённой ДНФ. Симметрическая ФАЛ, примеры . . . . .	55
5.5	Теорема Журавлёва о ДНФ сумма минимальных. Цепная ФАЛ, примеры . . . . .	58
<b>6</b>	<b>Лекция 6. Формула алгебры логики</b>	<b>61</b>
6.1	Базис, функциональный элемент, входной и выходной алфавиты . . . . .	61
6.2	Индуктивное (по глубине) определение формулы . . . . .	61
6.3	Формула как граф. Пример . . . . .	63
6.4	Функционалы сложности формул. . . . .	64
6.5	Утверждение о связи функционалов сложности . . . . .	65
6.6	Эквивалентность и изоморфизм формул. Подформула. Принцип эквивалентной замены . . . . .	66
6.7	Известные тождества формул. Преобразования подобия . . . . .	67
6.8	Формула с поднятыми отрицаниями . . . . .	68

6.9	Оптимизация подобных формул по глубине. Альтернирование . . .	68
<b>7</b>	<b>Лекция 7. Схема из функциональных элементов 1</b>	<b>72</b>
7.1	Граф, сеть. Схема: изоморфность, эквивалентность, классы, сложность . . . . .	72
7.2	Схема из функциональных элементов (СФЭ). Квазидерево. Переход от СФЭ к системе формул (квазидеревьев) . . . . .	73
7.3	Висячая вершина. Приведенная СФЭ. Функционалы сложности СФЭ . . . . .	76
7.4	Утверждение о связи функционалов сложности у СФЭ. (Аналогично случаю формулы) . . . . .	77
7.5	Оценки количеств формул. Квазиизоморфность формул . . . . .	77
7.6	Оценка количества попарно неэквивалентных СФЭ . . . . .	80
<b>8</b>	<b>Лекция 8. Эквивалентные преобразования СФЭ</b>	<b>81</b>
8.1	Пример эквивалентных СФЭ . . . . .	81
8.2	Задача эквивалентных преобразований (ЭП) для СФЭ. Модификация тождества . . . . .	83
8.3	Подсхема СФЭ. Многократное ЭП. Полнота системы тождеств . .	84
8.4	Сопоставление ЭП формул и ЭП СФЭ. Примеры . . . . .	85
8.5	Утверждение о получении полной системы тождеств для ЭП СФЭ	89
<b>9</b>	<b>Лекция 9. Контактная схема</b>	<b>93</b>
9.1	Типы проводящих контактов, их соответствие с МОП-транзисторами	93
9.2	Контактная схема (КС). Пример. Матрица проводимости, функция проводимости. Примеры . . . . .	94
9.3	Цепь контактов. Проводящая цепь. Сопоставление цепи функции проводимости. Примеры . . . . .	97
9.4	Сечения КС, их виды. ФАЛ отделимости . . . . .	99
9.5	Пи-схема. Последовательное и параллельное соединения . . . . .	100
9.6	Сопоставление пи-схеме формулы с поднятыми отрицаниями . . .	101
9.7	Контактное дерево . . . . .	102
9.8	Оценка числа попарно неэквивалентных пи-схем . . . . .	102
9.9	Неприводимая КС. Утверждение об оценке числа попарно неэквивалентных КС . . . . .	103

<b>10 Лекция 10. Нижние оценки сложности ФАЛ</b>	<b>106</b>
10.1 Задача синтеза. Простейшие верхние оценки . . . . .	106
10.2 Нижние оценки сложности системы ФАЛ . . . . .	106
10.3 Нижние оценки сложности отдельной ФАЛ . . . . .	107
10.4 "Более высокие" нижние оценки сложности отдельной ФАЛ. При- мер . . . . .	109
10.5 Нижняя оценка сложности вектор-функции. Пример . . . . .	111
10.6 Верхние оценки некоторых ФАЛ . . . . .	113
<b>11 Лекция 11. Суперпозиции различных схем</b>	<b>117</b>
11.1 Суперпозиция систем ФАЛ. Суперпозиция схем . . . . .	117
11.2 Простейшая схема. Частные случаи суперпозиции . . . . .	118
11.3 Правильная и корректная суперпозиции . . . . .	118
11.4 Суперпозиция СФЭ. Примеры . . . . .	119
11.5 Суперпозиция КС. Пример . . . . .	120
11.6 Лемма Шеннона и её следствие. Разделительность КС. Пример . .	122
11.7 Операция присоединения контактов к выходу КС. Каскадные кон- тактные схемы (ККС). Пример . . . . .	124
11.8 Утверждение о существовании ККС, реализующей систему из от- рицаний ФАЛ . . . . .	124
<b>12 Лекция 12. Каскадная СФЭ</b>	<b>126</b>
12.1 Свойства ККС. Полная ККС. Частный случай . . . . .	126
12.2 Утверждение о существовании ККС, реализующей систему из от- рицаний ФАЛ . . . . .	128
12.3 Пример с конъюнктивным и дизъюнктивным контактными дере- вьями . . . . .	129
12.4 Каскадная СФЭ. Примеры . . . . .	130
12.5 Метод каскадов в общем виде. Пример . . . . .	131
12.6 Оценки сложности некоторых каскадных схем . . . . .	133
12.7 Метод Шеннона . . . . .	134
<b>13 Лекция 13. Асимптотически наилучший метод синтеза СФЭ</b>	<b>136</b>
13.1 Нижние мощностные оценки функций Шеннона . . . . .	136

13.2	Задача синтеза схем для ФАЛ из специальных классов. Мощностные оценки в таких задачах . . . . .	139
13.3	Дизъюнктивно-универсальное множество. Метод Лупанова для синтеза СФЭ . . . . .	141
13.4	Асимптотически наилучшая верхняя оценка сложности для СФЭ	143
<b>14</b>	<b>Лекция 14. Асимптотически наилучший метод синтеза формул</b>	<b>145</b>
14.1	$m$ -регулярное множество единичного куба, его свойства. Пример	146
14.2	Утверждение о существовании регулярного разбиения единичного куба. Продолжение примера . . . . .	147
14.3	Асимптотически наилучшая верхняя оценка сложности для формул	149
14.4	Асимптотически наилучшая верхняя оценка сложности для КС	151
<b>15</b>	<b>Лекция 15. Асимптотически наилучший метод синтеза КС 1</b>	<b>153</b>
15.1	Необходимые разложения для формулы . . . . .	154
15.2	Построение КС по разложению и её сложность . . . . .	154
15.3	Замечание о верхней оценке разбиения данной КС . . . . .	156
15.4	Задача синтеза схем для ФАЛ из специальных классов. Нижние оценки и их асимптотическая точность. Примеры . . . . .	157
15.5	Синтез схем для дешифраторов, мультиплексоров и некоторых других ФАЛ, встречающихся в приложениях, оценки их сложности	159
<b>16</b>	<b>Лекция 16. Самокорректирующаяся КС</b>	<b>161</b>
16.1	Верхние оценки сложности для некоторых дешифраторов. Каскадная пи-схема . . . . .	161
16.2	Верхние оценки сложности для некоторых мультиплексоров . . .	164
16.3	Самокорректирующаяся КС. Обрыв и замыкание контакта . . . .	166
16.4	Тривиальное решение задачи повышения надёжности схемы . . .	167
16.5	Однородная связная подсхема. Пример. Коррекция с помощью разбиения . . . . .	168
16.6	Асимптотика функции Шеннона для самокорректирующихся КС	169
16.7	Случай схемы Кардо . . . . .	170
<b>17</b>	<b>Лекция 17. Задача контроля схем</b>	<b>172</b>
17.1	Постановка задачи контроля схем. Область определённости . . . .	172

17.2	Таблица контроля. Цель контроля: проверка, диагностика . . . . .	172
17.3	Тест для таблицы и тест для схемы. Тупиковый, минимальный, проверяющий и диагностический тесты . . . . .	174
17.4	Пример задачи. Способы построения тестов . . . . .	174
17.5	Утверждение о задании ФАЛ теста с помощью КНФ и его след- ствие. Продолжение примера . . . . .	177
17.6	Оценки длин тестов . . . . .	178
<b>18</b>	<b>Лекция 18. Эквивалентные преобразования КС</b>	<b>180</b>
18.1	КС с неразделёнными полюсами . . . . .	180
18.2	Тождество КС. Примеры . . . . .	182
18.3	Подстановка (модификация) тождества КС. Виды модификаций. Примеры . . . . .	183
18.4	Понятие подсхемы, выполняющее принцип эквивалентной замены	184
18.5	Примеры эквивалентных преобразований КС . . . . .	185
18.6	Основные тождества. Вывод вспомогательных тождеств . . . . .	186
18.7	Полная система тождеств. Обобщённые тождества . . . . .	188
<b>19</b>	<b>Лекция 19. Полнота системы основных тождеств</b>	<b>192</b>
19.1	Вывод обобщённых тождеств . . . . .	192
19.2	Каноническая форма КС. Переход к канонической форме через ЭП	192
19.3	ЭП эквивалентных КС . . . . .	196
<b>20</b>	<b>Лекция 20. Цикломатическое число КС</b>	<b>197</b>
20.1	Суммарное цикломатическое число КС. Пример . . . . .	197
20.2	Утверждение об изменении суммарного цикломатического числа при ЭП. Пример . . . . .	198
20.3	Доказательство отсутствия конечной полной системы тождеств в классе КС . . . . .	200
20.4	Примеры задач на синтез схем ФАЛ из специальных классов . . .	201
<b>21</b>	<b>Лекция 21. Модификации классов схем</b>	<b>205</b>
21.1	Вычисляющая программа. Команды ввода, вычисляющие коман- ды, вывод . . . . .	205
21.2	Ширина программы. Пример . . . . .	207

---

21.3	Двоичная решающая диаграмма ( <i>BDD</i> ) . . . . .	208
21.4	Схема на КМОП-транзисторах. Пример . . . . .	209
21.5	Автомат. Скалярные и векторные канонические уравнения. Схем- ная реализация автомата. Элементы задержки . . . . .	210
21.6	Реализация элементов задержки через схемы с "мгновенными" обратными связями . . . . .	213



## Лекция 1. ДНФ функции алгебры логики

### Введение: о кибернетике, системах управления, курсе, его разделах и сферах применения

Кибернетика — наука об общих законах хранения, получения, преобразования и передачи информации в сложных системах управления (С.В. Яблонский, 1959 г.). Математическая кибернетика — математические модели и методы исследования сложных систем управления. Основным объектом исследования в кибернетике является система управления. Система управления показывает взаимодействие между двумя объектами, такие как управляющая система и объект управления.

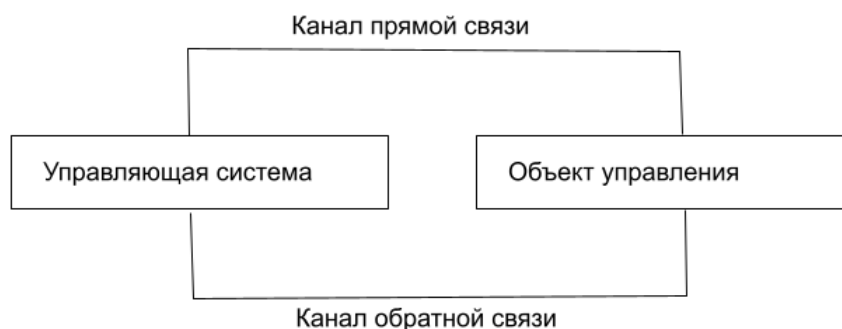


Рис. 1.1. Система управления

Функционирование системы управления — это круговорот информации. Информация о состоянии объекта управления поступает в управляющую систему, в управляющей системе эта информация обрабатывается и выдается команда на выполнения тех или иных действий. В рамках такого подхода решаются 2 основные задачи: задача анализа и синтеза. Анализ предполагает, что необходимо найти, как устроена управляющая система. При построении системы управления известны закон поведения и управляемости объекта управления, цель управления, класс функций управления, класс управляющей системы и критерий качества. В зависимости от критерия качества формулируются следующие основные задачи:

- 1) Выбор оптимальной функции управления из данного класса (синтез управления).
- 2) Построение (синтез) оптимальной управляющей системы заданного типа.
- 3) Оценка качества построенной системы управления, а затем, возможно, коррекция классов и переход к пункту 1.

Курс посвящен основным моделям, методам и результатам математической кибернетики, связанным с теорией дискретных управляющих систем (ДУС), задачами их анализа и синтеза.

### Основные сферы применения результатов курса

- 1) Схемная и структурная реализация дискретных функций и алгоритмов, оценки ее сложности.
- 2) Различные задачи программно-аппаратной реализации алгоритмов. В большинстве объектов приходится вкладывать алгоритм. Эти задачи часто приводят к тому, что существенную часть алгоритма приходится вкладывать не в программу, а в схемы.
- 3) Разработка методов автоматизации проектирования заказных СБИС, программирование *FPGA* и др.

### Минимизация ДНФ и связанные с ней задачи

Рассматривается представление функций алгебры логики (ФАЛ) дизъюнктивными нормальными формами (ДНФ), его "геометрическая" интерпретация, совершенная ДНФ и критерий единственности ДНФ. Дискретная математика отличается от непрерывной математики тем, что в дискретной математике есть ограниченные множества, на основе которых строятся сложные объекты. В дискретной математике есть одна операция — операция суперпозиции функции, которая позволяет вместо переменных одной функции подставлять другие функции. Множество Булево состоит из 2 элементов, записываются объекты этого множества (единичный  $n$ - мерный куб):

$$B = \{0, 1\}, \quad B^n = B \times \dots \times B = \{\beta = (\beta_1, \dots, \beta_n) : \beta_i \in B\}$$

Записываются примеры этого объекта – двумерный единичный куб размерности 2 и трехмерный куб:

$$B^1 = B$$

$$B^2 = \{(00), (01), (10), (11)\}$$

$$B^3 = \{(000), (001), (010), (011), (100), (101), (110), (111)\}$$

Мощность записывается следующим образом:

$$|B^n| = 2^n$$

Наборы выписаны в лексикографическом порядке.

## Отношения полного/линейного и частичного монотонного порядков. Отношение соседства. Вес набора

Рассматривается следующее отображение:

$$v : B^n \rightarrow [0, 2^n)$$

$$v((\beta_1, \dots, \beta_n)) = \beta_1 2^{n-1} + \dots + \beta_{n-1} \times 2^1 + \beta_n \times 2^0$$

Наборы  $B^n$  записаны в порядке возрастания их лексикографического номера. Один набор предшествует другому, если его лексикографический номер меньше, чем лексикографический номер другого набора. Это называется отношением полного или линейного порядка на единичном кубе  $B^n$ .

Отношение частичного монотонного порядка записывается следующим образом:

$$\forall \beta = (\beta_1, \dots, \beta_n) \in B^n, \quad \forall \gamma = (\gamma_1, \dots, \gamma_n) \in B^n \quad \beta \leq \gamma \Leftrightarrow \beta_1 \leq \gamma_1, \dots, \beta_n \leq \gamma_n$$

Любые 2 набора сравнимы.

$$(01) \not\leq (10)$$

$$(10) \leq (01)$$

Рассматривается отношение соседства.  $\beta$  и  $\gamma$  – соседние, если отличаются только в одном разряде. Это означает, что расстояние Хэмминга между ними равна 1. Если 2 набора соседние, то они сравнимы между собой по отношению частичного порядка.

Вес набора  $\beta$  записывается как:

$$\|\beta\| = \beta_1 + \dots + \beta_n$$

$$B_i^n = \{\beta \in B^n; \|\beta\| = i\}$$

### Геометрическая интерпретация: рёбра, слои, грани (их представления, мощность и количество в кубе)

При геометрическом изображении единичного куба соседние наборы соединяются ребром.

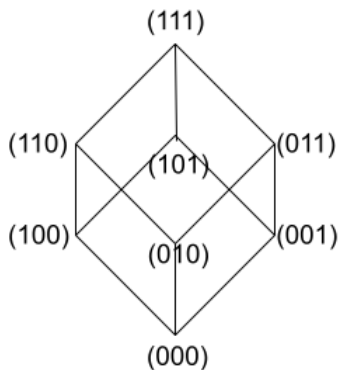


Рис. 1.2. Геометрическая интерпретация

В слой  $B_1^3$  входят наборы  $\{(100), (010), (001)\}$ . В слой  $B_0^3$  входит набор  $\{(000)\}$ . Если 2 набора сравнимы и один набор больше другого, то они расположены в разных слоях. Чем больше единиц в наборе, тем выше расположен соответствующий слой единичного куба. В слой  $B_2^3$  входят наборы  $\{(110), (101), (011)\}$ . В слой  $B_3^3$  входит набор  $\{(111)\}$ .

#### Грани в $B^n$

Записываются фиксированные элементы грани:

$$\forall 1 \leq i_1 < \dots < i_l \leq n$$

$$\forall \sigma_1, \dots, \sigma_r \in B \Rightarrow B_{\sigma_1, \dots, \sigma_r}^{n; i_1, \dots, i_r} = \left\{ \beta = (\beta_1 \dots \beta_n) : \beta_{i_1} = \sigma_1, \dots, \beta_{i_r} = \sigma_r \right\}$$

Это есть грань ранга  $r$  или размерности  $(n - r)$  в кубе  $B^n$ . Объединением этих множеств можно представлять другие подмножества в единичном кубе. Чтобы задать грань, также можно ввести троичный набор  $\gamma$ . Тогда грань будет обозначаться как  $\Gamma_\gamma$ .

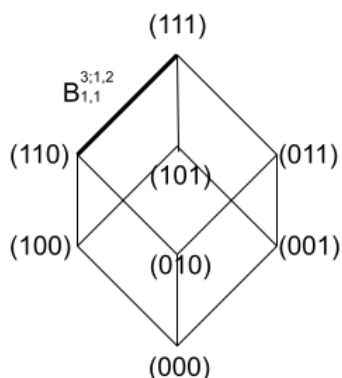


Рис. 1.3. Грань размерности 1 в трехмерном кубе

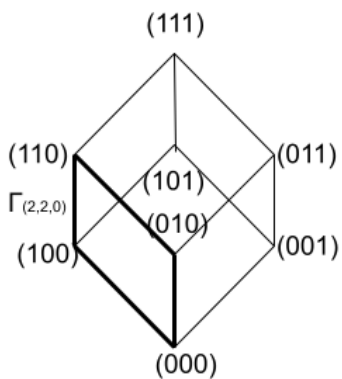


Рис. 1.4. Грань размерности 2 в трехмерном кубе

Мощность грани записывается следующим образом:

$$|\Gamma_\gamma| = 2^{n-r}$$

$n - r$  — число свободных переменных.

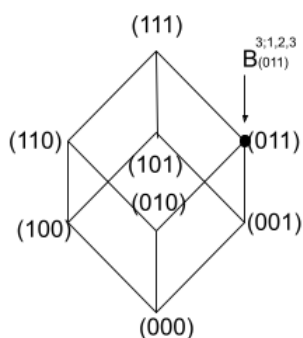


Рис. 1.5. Грань размерности 0 в трехмерном кубе

$n$  — длина набора. Чтобы задать грань, сперва необходимо выбрать переменные, которые фиксированы. Далее, каждому из этих фиксированных переменных нужно придать значение, в котором они зафиксированы. Таким образом, число граней ранга  $r$  в  $B^n$  кубе записывается следующим образом:

$$C_n^r \times 2^r$$

## Определение функции алгебры логики (ФАЛ)

Рассматриваются функции над наборами — счетный упорядоченный алфавит Булевских переменных:

$$\tilde{X} = \{x_1, x_2, \dots, x_n, \dots\}$$

Все функции будут зависеть от переменных, выбранных из этого множества.

$$X \subseteq \tilde{X}$$

Множество Булевских переменных функции алгебры логики  $f(x)$  записывается как:

$$|X| = n$$

Переменные  $X$  сопоставляются  $B^n(X)$ . Каждому разряду куба сопоставляется переменная из набора мощности  $n$ . Таким образом получается следующий алфавит:

$$X(n) = \{x_1, \dots, x_n\}$$

Функция алгебры логики  $f(X)$  — это отображение:

$$B^n(X) \rightarrow B$$

Каждому двоичному набору куба сопоставляется Булевская константа и записывается таблица значений  $\alpha_f$ .

$x_1 \dots x_n$	$f$
0...00	$\alpha_0$
0...01	$\alpha_1$
1...1	$\alpha_{2^n-1}$

Рис. 1.6. Таблица значений для отображения

## Существенная и несущественная зависимости ФАЛ от переменных. Равенство ФАЛ. Множества ФАЛ

Функция  $f$  — существенная и зависит от переменной  $X_i$ , когда существуют два набора такие, что:

$$\exists (\beta_1, \dots, \beta_{i-1}^0, \beta_i + 1, \dots, \beta_n) = \beta$$

$$\exists (\beta_1, \dots, \beta_{i-1}^1, \dots, \beta_n) = \gamma$$

$$f(\gamma) \neq f(\beta)$$

Две функции совпадают, если все фиктивные переменные в каждом наборе изъяты. Таким образом, получится функция, существенно зависящая от одних и тех же переменных и совпадающая на любом наборе своих значений. Равенство функций  $f = g$  с точностью добавления или изъятия фиктивных Булевских

переменных. ФАЛ от Булевских переменных из  $X$  будет  $P_2(X)$ . Мощность множества  $P_2(n)$  записывается как:

$$|P_2(n)| = 2^{2^n}$$

$\tilde{\alpha}_f$  — это стандартный вариант табличного задания функции, но еще и вариант задания характеристического множества функции  $N_f$ .

$$N_f = \{\beta \in B^n : f(\beta) = 1\}$$

Также есть дополнение характеристического множества функции:

$$\bar{N}_f = B^n \setminus N_f$$

## Примеры со стандартными ФАЛ

Записываются функции множества  $P_2(1)$ :  $x_1, \bar{x}_1, 0, 1$ . Далее записываются функции двух переменных. В таблице также записываются функции конъюнкции, дизъюнкции, сложения по модулю 2, эквивалентности, отрицание конъюнкции, отрицание дизъюнкции и импликация.

$x_1x_2$	$\&$	$\vee$	$\oplus$	$\sim$	$\downarrow$	$\mid$	$\rightarrow$
00	0	0	0	1	1	1	1
01	0	1	1	0	0	1	1
10	0	1	1	0	0	1	0
11	1	1	0	1	0	0	1

Рис. 1.7. Функция двух переменных

Мощность множества записывается как:

$$|P_2(2)| = 16$$

В это множество входят 6 функций ( $0, 1, x_1, \bar{x}_1, x_2, \bar{x}_2$ ), существенно зависящие не более, чем от одной переменной. Остальные 10 функций существенно зависят от двух переменных.



$x_1x_2x_3$	$H$	$I_3$
000	0	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0
110	1	0
111	1	1

Рис. 1.8. Функция трех переменных

Рассматривается функция трех переменных. В таблице значений записываются функции голосований и линейная функция.

Характеристическое множество функции голосования записывается следующим образом:

$$N_H = \{(011), (110), (101), (111)\}$$

Столбец значения функции  $I_3$  записывается как:

$$\tilde{\alpha}_{I_3} = (01101001)$$

## Представление ФАЛ дизъюнктивными нормальными формами (ДНФ)

Функции представляются с помощью либо формул, либо схем. Формулы называются дизъюнктивными нормальными формами (ДНФ). ДНФ и КНФ — это специальные виды формул в базисе  $B_0 = \{\&, \vee, \bar{\phantom{x}}\}$ , который состоит из конъюнкции, дизъюнкции и отрицания. Определяется первый уровень ДНФ ( $x_i, \bar{x}_i$  — буквы Булевой переменной  $x_i$ ):

$$x_i = x_i^1, \quad \bar{x}_i = x_i^0$$

$$x_i^\sigma = 1 \Leftrightarrow x_i = \sigma$$

Дальше определяются элементарные конъюнкции букв различных переменных:

$$K = x_{i_1}^{\sigma_1} \dots x_{i_r}^{\sigma_r} \quad (1 \leq i_1 < i_2 < i_r \leq n, \sigma_1, \dots, \sigma_r \in B)$$

Операция конъюнкции ассоциативная и коммутативная.

Следующий уровень определения ДНФ от  $X(n)$  записывается как:

$$a = K_1 \vee \dots \vee K_s$$

ДНФ — это дизъюнкция различных элементарных конъюнкций от множества переменных. Характеристическое множество конъюнкции записывается следующим образом:

$$N_k = B_{\sigma_1, \dots, \sigma_r}^{n; i_1, \dots, i_r}$$

Характеристическое множество конъюнкции — это грань в кубе, где фиксированы переменные своими степенями.

ДНФ реализует функцию  $f$ . Характеристическое множество этой функции записывается законом дизъюнкции:

$$N_f = N_{k_1} \cup \dots \cup N_{k_s}$$

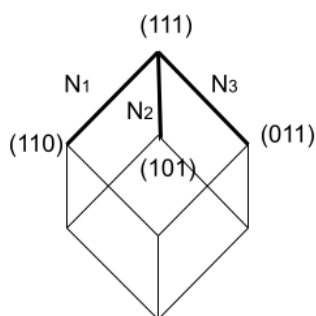


Рис. 1.9. Функция голосования в трехмерном кубе

$$N_H = B_3^3 \cup B_2^3$$

Записывается значение граней:

$$N_1 = B_{1,1}^{3;1,2}$$

$$K_1 = x_1x_2$$

$$K_2 = x_1x_3$$

$$K_3 = x_2x_3$$

Таким образом, ДНФ имеет значение:

$$x_1x_2 \vee x_1x_3 \vee x_2x_3 \rightarrow H$$

ДНФ можно также представить другой функцией:

$$x_1x_2 \vee x_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \rightarrow H$$

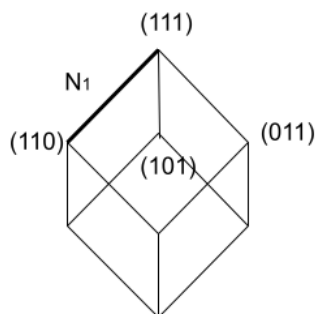


Рис. 1.10. Функция голосования в трехмерном кубе с гранью  $N_1$

Для любой функции можно записать совершенную ДНФ. Совершенная ДНФ — это ДНФ, у которой все элементарные конъюнкции имеют ранг, равный числу переменных. Если есть функция  $f$  от переменных, то совершенная ДНФ для произвольной функции записывается следующим образом:

$$\forall f(x_1, \dots, x_n) f \neq 0$$

$$a \rightarrow f = \vee x_1^{\sigma_1}, \dots, x_n^{\sigma_n}, \quad (\sigma_1 \dots \sigma_n) \in N_f$$

Совершенная ДНФ для функции голосования записывается как:

$$\bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$$

**Утверждение 1.1.** *Совершенная ДНФ ФАЛ  $f$ ,  $f \neq \bar{f} \in P_2(n)$ , является единственной ДНФ от БП (Булевских переменных)  $X(n)$ , которая реализует эту ФАЛ, тогда и только тогда, когда во множестве  $N_f$  нет соседних наборов.*

Совершенная ДНФ ФАЛ  $l, \bar{l}$ , является единственной ДНФ этой ФАЛ от БП  $X(n)$ .

## Лекция 2. Сокращённая ДНФ

Рассматриваются простейшие формулы над множеством Булевских переменных, которые называются дизъюнктивными нормальными формами (ДНФ).

Сначала определяется самая простейшая формула: либо переменная, либо ее отрицание. Отрицание переменной есть ее нулевая степень, а сама переменная есть ее первая степень. Степени называются буквами переменной  $X_i$ .

$$X(n) = \{x_1, \dots, x_n\}$$

$$x_i^0 = \bar{x}_i \quad x_i^1 = x_i$$

$$x^\sigma = 1 \leftrightarrow x = \sigma$$

$$K = x_{i_1}^{\sigma_1} \cdot \dots \cdot x_{i_r}^{\sigma_r}$$

$$N_K = B_{\sigma_1 \dots \sigma_r}^{n; i_1 \dots i_r}$$

Записывается конъюнкция переменных или их отрицаний, взятых из множества в степенях. Конъюнкция — это произведение букв различных переменных в количестве  $r$  штук. Эти конъюнкции называются элементарными конъюнкциями ранга  $r = R(K)$  от Булевских переменных (БП)  $X(n)$ .

$$1 \leq i_1 < \dots < i_r \leq n, \quad \sigma_j \in B$$

Дальше записываются дизъюнкции различных элементарных конъюнкций.

$$a = K_1 \vee \dots \vee K_s$$

Ранг ДНФ — это сумма рангов всех ее элементарных конъюнкций:

$$R(a) = R(K_1) + \dots + R(K_s)$$

Каждая ДНФ реализует Булевскую функцию. Характеристическое множество функции  $f$  — это множество тех наборов, на которых эта функция обращается в 1.

$$f \in P_2(n)$$

$$a = f \leftrightarrow N_f = N_{K_1} \cup \dots \cup N_{K_s}$$

$$N_f = \{f^{-1}(1)\}$$

ДНФ совершенная только тогда, когда все элементарные конъюнкции имеют ранг  $n$ .

$$\forall j \in [1, n] \quad R = (K_j) = n$$

Мощность характеристического множества записывается следующим образом:

$$|N_{K_j}| = 1$$

Таким образом, совершенная форма ДНФ записывается как:

$$\forall f \in P_2(n) \quad f \neq 0 \rightarrow a_f = \bigvee_{\sigma_1 \dots \sigma_n \in N_f} x_1^{\sigma_1} \dots x_n^{\sigma_n}$$

Для функции голосования можно построить много ДНФ, для линейной функции от трех переменных существует единственная ДНФ от этих же переменных, которая эту функции представляет. Совершенная ДНФ является единственной для функции  $f$ , когда в ее характеристическом множестве нет соседних наборов. Линейная функция имеет следующую формулу:

$$l_n = x_1 \oplus \dots \oplus x_n$$

$$\bar{l}_n = x_1 \oplus \dots \oplus x_n \oplus 1$$

$$N_{l_n} = B_{\text{неч}}^n = B_n^1 \cup B_n^3 \cup \dots$$

$$N_{\bar{l}_n} = B_{\text{чет}}^n = B_n^0 \cup B_n^2 \cup \dots$$

Если брать все нечетные слои, то в этом множестве нет никаких соседних наборов.

## Конъюнктивная нормальная форма (КНФ). Совершенная КНФ (СКНФ)

Конъюнктивная нормальная форма (КНФ) — это формулы, двойственные к ДНФ. В КНФ порядок конъюнкции и дизъюнкции меняется местами.

$$X(n) = \{x_1, \dots, x_n\}$$

$$x_i^0 = \bar{x}_i \quad x_i^1 = x_i$$

$$x^\sigma = 1 \leftrightarrow x = \sigma$$

$$N_J = B_{\sigma_1 \dots \sigma_r}^{n; i_1 \dots i_r}$$

Элементарная дизъюнкция записывается следующим образом:

$$J = x_{i_1}^{\sigma_1} \vee \dots \vee x_{i_r}^{\sigma_r}$$

Элементарная дизъюнкция ранга  $r = R(K)$  от БП  $X(n)$ :

$$1 \leq i_1 < \dots < i_r \leq n, \quad \sigma_j \in B$$

Дальше записывается конъюнкция различных элементарных дизъюнкций:

$$b = J_1 \& \dots \& J_s$$

Ранг КНФ — это сумма рангов всех ее элементарных дизъюнкций:

$$R(b) = R(J_1) + \dots + R(J_s)$$

Берется функция  $f \in P_2(n)$ :

$$b = f \rightarrow \bar{N}_f = \bar{N}_{J_1} \cup \dots \cup \bar{N}_{J_s}$$

$\bar{N}_f$  — дополнение характеристического множества покрывается гранями:

$$\bar{N}_{J_1}, \dots, \bar{N}_{J_s}$$

$$\bar{N}_f = \{f^{-1}(0)\}$$

КНФ совершенная только тогда, когда все элементарные конъюнкции имеют ранг  $n$ .

$$\forall j \in [1, n] \quad R = (J_j) = n$$

Мощность характеристического множества записывается следующим образом:

$$|\bar{N}_{J_j}| = 1$$

Таким образом, совершенная форма КНФ записывается как:

$$\forall f \in P_2(n) \quad f \neq 1 \rightarrow b_f = \bigg\&_{\sigma_1 \dots \sigma_n \in N_f} x_1^{\bar{\sigma}_1} \vee \dots \vee x_n^{\bar{\sigma}_n}$$

## Пример построения СКНФ. Сравнение с СДНФ

Берется функция трех переменных:  $x_1x_2x_3$ .

$x_1x_2x_3$	$g$
000	0
001	1
010	1
011	1
100	1
101	1
110	1
111	0

Рис. 2.1. Функция трех переменных

Совершенная ДНФ функции  $g$  записывается как:

$$a_g = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3$$

Совершенная КНФ функции  $g$  или сокращенная ДНФ записывается как:

$$b_g = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) = x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee \bar{x}_1x_2 \vee x_2\bar{x}_3 \vee \bar{x}_1x_3 \vee \bar{x}_2x_3$$

При раскрытии скобок используются следующие тождества:

$$x \times \bar{x} = 0$$

$$x \vee 0 = x$$

## Некоторые тождества алгебры логики, необходимые в дальнейшем

Функция  $f_1$  имплицирует в функцию  $f_2$  или функция  $f_2$  поглощает функцию  $f_1$  тогда, когда:

$$N_{f_1} \subseteq N_{f_2}$$



Импликация из функции  $f_1$  в функцию  $f_2$  тождественна равна 1.

$$f_1 = f_1 \cdot f_2 \leftrightarrow f_2 = f_1 \vee f_2$$

$$\vec{\alpha}_{f_1} \leq \vec{\alpha}_{f_2}$$

$$f_1 \leq f_2$$

Вводится такое понятие, как тождество поглощения:

$$t^{\Pi} : x_1 \vee x_1 x_2 = x_1$$

$t_{\&}^k$  — тождество коммутативности для конъюнкции.

$t_{\vee}^k$  — тождество коммутативности для дизъюнкции.

$$x_1 \circ x_2 = x_2 \circ x_1$$

$t_{\&}^A$  — тождество ассоциативности конъюнкции.

$t_{\vee}^A$  — тождество ассоциативности дизъюнкции.

$$(x_1 \circ x_2) \circ x_3 = x_1 \circ (x_2 \circ x_3)$$

$t_{\&,\vee}^D$  — тождество дистрибутивности конъюнкции по отношению к дизъюнкции.

$$x_1(x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3$$

$t_{\vee,\&}^D$  — тождество дистрибутивности дизъюнкции по отношению к конъюнкции.

Тождество приведения подобных:

$$x \cdot \bar{x} = 0$$

$$x \vee \bar{x} = 1$$

$$0 \cdot x = 0$$

$$1 \vee x = 1$$

$$x \cdot x = x \vee x = x$$

## Импликанта. Простая импликанта

Рассматривается отношение имплицируемости, где в качестве функции  $f_1$  выступает элементарная конъюнкция. Любая элементарная конъюнкция  $K$ , имплицирующая функцию  $f$ , называется импликантой функции  $f$ .

$$N_K \subseteq N_f$$

$N_K$  — грань функции  $f$ .

Элементарная конъюнкция  $K$  — это простая импликанта функции  $f$ , когда  $K$  является импликантой функции  $f$  и не существует элементарной конъюнкции  $K'$  такой, что:

$$N_K \subset N_{K'} \subseteq N_f$$

$N_K$  — максимальная грань функции  $f$ .

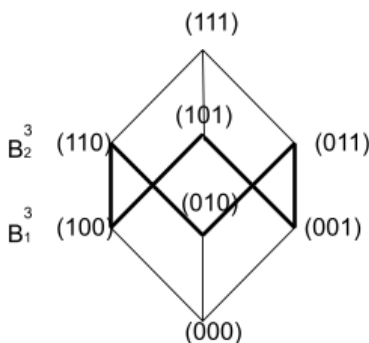


Рис. 2.2. Геометрическая интерпретация сокращенной ДНФ

Сокращенная ДНФ функции  $f$  — это ДНФ, которая представляет собой дизъюнкцию всех ее простых импликант. С геометрической точки зрения это представление характеристического множества функции  $f$  в виде объединения всех максимальных граней этой функции.

Сокращенную ДНФ можно строить от геометрической. Для этого используется карта Карно. Эту карту можно реально использовать только для функции трех или четырех переменных. Геометрическая интерпретация может служить основой для построения сокращенной ДНФ только для функции не более четырех переменных.

## Утверждение о построении сокращённой ДНФ и его следствие

**Утверждение 2.1.** Пусть  $a'$  и  $a''$  — сокращённые ДНФ ФАЛ  $f'$  и  $f''$  соответственно, а ДНФ  $a$  без поглощений ЭК получается из формулы  $a' \cdot a''$  в результате раскрытия скобок и приведения подобных. Тогда  $a$  — сокращённая ДНФ ФАЛ:  $f = f' \cdot f''$

Если ДНФ  $a$  без поглощений ЭК получается из КНФ  $b$  ФАЛ  $f$  в результате раскрытия скобок и приведения подобных, то  $a$  — сокращённая ДНФ ФАЛ  $f$ .

$$N_K \subset N_{K'} \leftrightarrow \{K'\} \subseteq \{K\}$$

**Доказательство.**

Достаточно доказать, что в ДНФ  $a$  присутствуют все простые импликанты функции  $f$ . Пусть  $K$  — простая импликанта функции  $f$ . Тогда  $K$  — импликанта функций  $f'$  и  $f''$ . Следовательно, существует простая импликанта  $K'$  функции  $f'$ , поглощающая  $K$ .

$$\{K'\} \subseteq \{K\}$$

Аналогично существует простая импликанта  $K''$  функции  $f''$ , поглощающая  $K$ .

$$\{K''\} \subseteq \{K\}$$

Следовательно:

$$a' = K' \vee \dots$$

$$a'' = K'' \vee \dots$$

$$a' \cdot a'' = K' \cdot K'' \vee \dots$$

$$\tilde{K} = K' \cdot K''$$

$$\{\tilde{K}\} = \{K'\} \cup \{K''\} \subseteq \{K\}$$

$$K = \tilde{K}$$

■

## Тождество обобщённого склеивания. Расширение и строгое расширение ДНФ

Тождество обобщённого склеивания представляет собой:

$$t^{\text{OC}} : \bar{x}_1x_2 \vee x_1x_3 = \bar{x}_1x_2 \vee x_1x_3 \vee x_2x_3$$

Это тождество можно применять к элементарным конъюнкциям. Если расширять исходную ДНФ до тех пор, пока расширение возможно, то получится сокращенная ДНФ функции. Пусть есть ДНФ:

$$a = \bar{x}_iK' \vee x_iK'' \vee \tilde{a} \xrightarrow{t^{\text{OC}}} \bar{x}_iK' \vee x_iK'' \vee K'K'' \vee \tilde{a} = a'$$

Таким образом,  $a'$  – это расширение ДНФ  $a$  и строгое расширение тогда, когда  $K'K''$  не поглощается элементарными конъюнкциями  $a'$ .

## Метод Блейка построения сокращённой ДНФ и его следствие

**Утверждение 2.2.** *ДНФ без поглощений ЭК является сокращенной ДНФ тогда только тогда, когда она не имеет строгих расширений.*

Как следствие можно сформулировать алгоритм построению сокращенной ДНФ из какой-либо ДНФ. Из любой ДНФ  $a$  ФАЛ  $f$  можно получить сокращенную ДНФ этой ФАЛ в результате построения последовательных строгих расширений и приведения подобных до получения ДНФ без поглощений ЭК, не имеющей строгих расширений. Этот метод построения сокращенной ДНФ называется методом Блейка.

**Доказательство.**

Пусть  $a$  – это ДНФ без поглощений ЭК и без строгих расширений.

$$a = a(x_1, \dots, x_n)$$

Пусть ДНФ  $a$  реализует функцию  $f$ :

$$f \in P_2(n)$$

Необходимо доказать, что ДНФ  $a$  — это сокращенная ДНФ функции  $f$ .

Достаточно доказать, что в ДНФ  $a$  входят все простые импликанты функции  $f$ . Доказывается от противного. Пусть утверждение неверно и существует ЭК  $K$  — простая импликанта функции  $f$  такая, что:

$$K \notin a$$

Рассматривается множество  $(k)$  всех тех импликант множества ФАЛ  $f$ , которые не являются импликантами ни одной ЭК из ДНФ  $a$ .

$$k \neq \emptyset \quad K \in k$$

В множестве  $k$  нет элементарных конъюнкций ранга  $n$ . Предполагается, что в множестве  $k$  есть конъюнкция:

$$x_1^{\sigma_1} \dots x_n^{\sigma_n}$$
$$f(\sigma_1 \dots \sigma_n) = 1$$

Но при этом ДНФ  $a$  реализует функцию  $f$ . Следовательно, существует конъюнкция  $\tilde{K}$  из  $a$ , которая поглощает конъюнкцию из ЭК:

$$\tilde{K} \in a : \tilde{K}(\sigma_1 \dots \sigma_n) = 1$$

Тогда получается противоречие с выбранным условием, что в множество входят только те импликанты функции, которые не являются импликантами ни одной ЭК из множества  $a$ .

Существует  $\tilde{k}$  — ЭК максимального ранга из множества  $k$ .

$$R(\tilde{k}) < n \quad 1 \leq i \leq n$$

$$x_i, \bar{x}_i \notin \{\tilde{k}\}$$

Рассматриваются следующие 2 элементарные конъюнкции:

$$K' = x_i \tilde{k}$$

$$K'' = \bar{x}_i \tilde{k}$$

Любая из этих элементарных конъюнкций должна войти в ДНФ  $a$ . Следовательно, существует  $\tilde{K}' \in a$ , которая поглощает  $K'$  и существует  $\tilde{K}'' \in a$ , которая поглощает  $K''$ .

$$\tilde{K}' = x_i \tilde{k}'$$

$$\tilde{K}'' = \bar{x}_i \tilde{k}''$$

Применяя тождество обобщенного склеивания, получится конъюнкция, которая должна будет входить в ДНФ  $a$  или поглощать какую-нибудь конъюнкцию из ДНФ  $a$ :

$$\tilde{k}' \cdot \tilde{k}''$$

Тогда конъюнкция  $\tilde{k}$  является импликантой элементарной конъюнкции из  $a$ . Таким образом, это противоречие доказывает, что заданное предположение неверно. ■

## Лекция 3. Особенности сокращённых ДНФ

**Утверждение 3.1.** Дизъюнктивная нормальная форма  $\cap T$  ФАЛ  $f$  состоит из тех простых импликант ФАЛ  $f$ , которые соответствуют ядровым граням этой ФАЛ.

В следствии, сокращенная ДНФ ФАЛ  $f$  является ее единственной тушиковой ДНФ тогда и только тогда, когда  $f$  — ядровая ФАЛ, т.е. все ее максимальные грани входят в ядро.

**Утверждение 3.2.** Простая импликанта  $K$  ФАЛ  $f$  входит в ДНФ  $\sum T$  тогда и только тогда, когда грань  $N_K$  не является регулярной гранью этой ФАЛ.

### Окрестность грани, степень локальности

Пусть есть функция  $f \in P_2(n)$  и  $N_K$  является максимальной гранью функции  $f$ . Тогда индукции по целому неотрицательному параметру  $r = 0, 1, \dots$  будут определять окрестность  $(S_r(N_K, f))$  порядка  $r$  грани  $N_K$  функции  $f$ . Окрестность порядка 0 записывается следующим образом:

$$S_0(N_K, f) = \{N_K\}$$

Если окрестность порядка  $r$  определена, то определяется окрестность следующего порядка:

$$S_r(N_K, f) \rightarrow S_{r+1}(N_K, f)$$

Окрестность следующего порядка является множеством тех максимальных граней функции  $f$ , которые имеют непустое пересечение хотя бы с одной гранью из окрестности предыдущего порядка. Порядок окрестности, в которую попадает грань функции, — это максимальная длина цепи из граней.

### Локальность критериев вхождения простой импликанты

Необходимо найти окрестность грани, по которой можно определить является грань ядровой или регулярной. Известно, что простая импликанта входит в ДНФ  $\cap T$  ФАЛ  $f$  тогда и только тогда, когда грань  $N_K$  — ядровая грань.

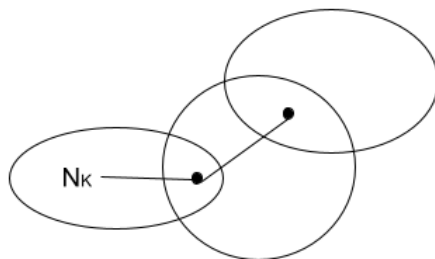


Рис. 3.1. Окрестность грани

Это происходит тогда, когда существует точка  $\alpha$  на грани  $N_K$  такая, что пучок функции  $f$  через точку  $\alpha$  состоит в точности из граней  $N_K$ .

$$\alpha \in N_K : \Pi_\alpha(f) = \{N_K\}$$

Чтобы проверить выше написанное равенство, достаточно знать окрестность первого порядка.

Известно, что простая импликанта входит в ДНФ  $\sum T$  ФАЛ  $f$  тогда и только тогда, когда грань  $N_K$  не является регулярной гранью. Это происходит тогда, когда существует нерегулярная точка  $\alpha$  на грани  $N_K$  такая, что при  $\forall \beta \in N_f \setminus N_K$  пучок функции  $f$  через точку  $\beta$  имеет следующую форму:

$$\Pi_\beta(f) \not\subseteq \Pi_\alpha(f)$$

Чтобы проверить грань, которая не проходит через точку  $\alpha$ , достаточно знать окрестность грани порядка 2 ( $S_2(N_K, f)$ ). Таким образом обосновывается локальность доказанных критериев. Вхождение простой импликанты в ДНФ суммы минимальных не является локальным критерием.

### Задача данной лекции. Линейные ФАЛ и их свойства

Совершенная ДНФ ФАЛ  $f$  является единственной ДНФ тогда, когда во множестве  $N_f$  нет соседних наборов. Этим свойством обладают 2 линейные функции:  $l_n$  и  $\bar{l}_n$ .

$$l_n(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n$$

$$\bar{l}_n(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n \oplus 1$$



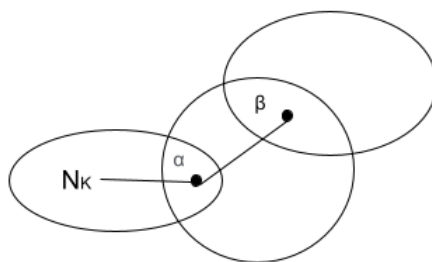


Рис. 3.2. Окрестность грани по второму критерию

Некоторые функции имеют линейную зависимость от переменных. Рассматриваются функции, у которых во множестве  $N_f$  нет соседних наборов, но только по одной переменной. Пусть есть функция  $f \in P_2(n)$ , в характеристическом множестве которого нет соседних наборов по переменной  $x_i$  при  $i \in [1, n]$ . Тогда в каждую импликанту  $K$  функции  $f$  входит либо  $x_i$ , либо  $\bar{x}_i$ .

Если в  $K$  нет переменных  $x_i$  и  $\bar{x}_i$ , точка  $\alpha \in N_K$  и точка  $\beta$  с ней соседняя по переменной  $x_i$ , то в точке  $\alpha$  и  $\beta$  функция и конъюнкция равны 1. Таким образом точка  $\beta$  тоже попадет в грань  $N_K$ . В характеристическом множестве функции оказываются две соседние точки. Это доказывает, что любая импликанта функции, у которой нет соседних наборов по переменной  $x_i$  всегда входит либо  $x_i$ , либо  $\bar{x}_i$ .

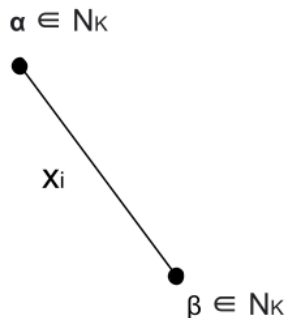


Рис. 3.3. ФАЛ  $f$  с характеристическим множеством  $N_f$

Функция  $f$  линейно зависит от переменной  $x_i$ .

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

Функция  $f$  — линейная ФАЛ тогда и только тогда, когда  $f$  линейно зависит

от любой существенной БП.

$$f = x_{i_1} \oplus \dots \oplus x_{i_2} \oplus \alpha_0$$

### Функции, монотонно зависящие от переменной и их свойство

Функция  $f$  монотонно зависит от функции  $x_i$  при  $i \in [1, n]$  тогда и только тогда, когда для любых двух соседних наборов по  $x_i$  справедливо равенство:

$$\alpha \leq \beta, \quad f(\alpha) \leq f(\beta)$$

В этом случае, любая простая импликанта монотонной функции по переменной  $x_i$  не может содержать отрицание переменной  $x_i$ .

**Доказательство.**

Пусть простая импликанта функции  $f$  записывается следующим образом:

$$K' = \bar{x}_i \tilde{K}$$

Грань  $N'_K$  входит во множество  $N_f$  и у всех наборов в этой грани переменная  $x_i = 0$ . Грань  $N'_K$  параллельно переносится в грань  $N''_K$  по переменной  $x_i$ . Рассматривается грань  $N''_K$ , где:

$$K'' = x_i \tilde{K}$$

Эти грани не пересекаются, потому что  $x_i = 1$  на грани  $N''_K$ .

$$N_{K'} \cup N_{K''} = \tilde{K}$$

В итоге получается противоречие, что  $K'$  — это простая импликанта. Это доказывает, что предположение верное и любая простая импликанта функции монотонной по переменной  $x_i$  не может содержать отрицание этой переменной. ■

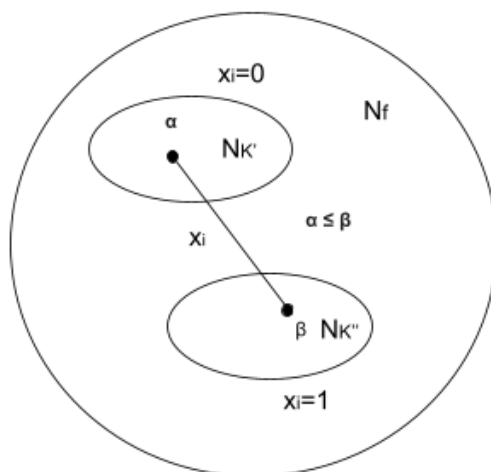


Рис. 3.4. Монотонная функция

## Монотонные ФАЛ, особенность их простых импликант.

### Имонотонность ФАЛ

Пусть функция  $f$  — монотонная функция, когда:

$$\forall \alpha \leq \beta : f(\alpha) \leq f(\beta)$$

Функция  $f$  монотонно зависит от переменной  $x_i$ . Любая импликанта функции  $f$  состоит только из конъюнкций чистых переменных без отрицаний (монотонная элементарная конъюнкция).

Имонотонность функции  $f$  по переменной  $x_i$  тогда и только тогда, когда  $\forall \alpha \leq \beta$  соседние наборы по  $x_i$ :

$$f(\alpha) \geq f(\beta)$$

### Особенности сокращённых ДНФ монотонных ФАЛ

Сокращенная ДНФ монотонной функции является ее единственной тупиковой ДНФ. У монотонной функции 1 тупиковая ДНФ, которая совпадает с ее сокращенной ДНФ. Для этого вводится обозначение для элементарных конъюнкций:

$$\forall \beta = (\beta_1, \dots, \beta_n) \in B^n \rightarrow K_\beta(x_1, \dots, x_n) = x_1^{\beta_1} \dots x_n^{\beta_n}$$

Рассматривается также конъюнкция:

$$\beta_i = 1, \quad K_{\beta}^{+}(x_1, \dots, x_n) = \bigwedge_{1 \leq i \leq n} x_i$$

$$K_{(101)}(x_1 x_2 x_3) = x_1 \bar{x}_2 x_3$$

$$K_{(101)}^{+}(x_1 x_2 x_3) = x_1 x_3$$

Чтобы перейти из конъюнкции  $K$  в конъюнкцию  $K^{+}$ , необходимо вычеркнуть переменные с отрицанием. Далее записывается характеристическое множество:

$$N_{K_{\beta}} = \{\beta\}$$

$N_{K_{\beta}^{+}}$  — это грань ранга  $r$ , где:

$$R = \|\beta\|$$

$$N_{K_{\beta}^{+}} = \{\beta' \geq \beta\}$$

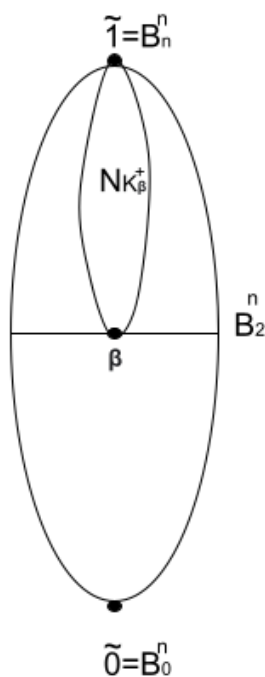


Рис. 3.5. Монотонная ФАЛ

Если  $f$  — монотонная функция, то  $\beta \in N_f$  тогда и только тогда, когда  $K_\beta^+$  — грань и импликанта функции  $f$ . При этом  $K_\beta^+$  — это простая импликанта функции  $f$  тогда и только тогда, когда  $\beta$  — нижняя "1" функции  $f$ .

$$\forall \gamma \leq \beta, \gamma \neq \beta \quad \gamma \in \bar{N}_f$$

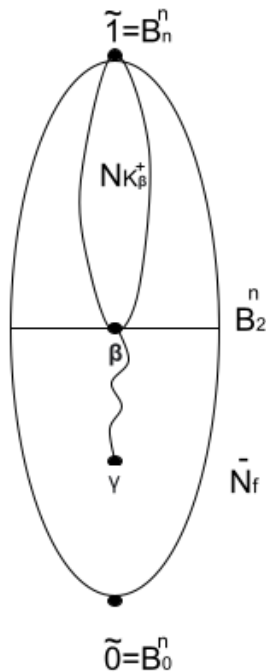


Рис. 3.6.  $\beta$  — нижняя "1" функции  $f$

$N_f^+$  — это множество всех нижних "1" функции  $f$ .

**Утверждение 3.3.** Сокращенная ДНФ а монотонной ФАЛ  $f$ ,  $f \in P_2(n)$ , является единственной тупиковой ДНФ этой ФАЛ и имеет вид:

$$a(x_1, \dots, x_n) = \bigvee_{\beta \in N_f^+} K_\beta^+(x_1, \dots, x_n)$$

При этом все наборы из  $N_f^+$  являются ядровыми точками ФАЛ  $f$ .

В следствии, монотонная ФАЛ является ядровой ФАЛ. Необходимо доказать, что любая простая импликанта монотонной функции может быть представлена как  $K_\beta^+$ , где  $\beta$  — нижняя "1" функции  $f$ .

### Доказательство.

Пусть  $K$  — это монотонная элементарная конъюнкция. Следовательно, можно записать следующее:

$$K = K_{\beta}^+, \quad \beta \in N_f$$

В силу простоты импликанты  $K$ :

$$\beta \in N_f^+$$

Это доказывает то, что  $\beta$  — нижняя "1" функции  $f$ . ■

### Доказательство.

Необходимо доказать, что любая точка  $\beta$  из множества  $N_f^+$  является ядровой точкой грани  $N_{K_{\beta}^+}$ . Если точка  $\beta$  не является ядровой точкой, то есть более широкая грань, которая ее покрывает. Пусть точка  $\beta$  покрывается гранью  $N_{K_{\beta}^+}$ , то следовательно:

$$\gamma \leq \beta$$

$$\gamma \neq \beta$$

Это означает, что  $\beta$  не является нижней "1" функции ( $\beta \notin N_f^+$ ), а это противоречит выбору точки  $\beta$  из множества нижних "1" функции. ■

## Покрывание, задача покрывания, функция покрывания и её свойства

Монотонные функции оказались очень удобным инструментом для описания способ решения целого ряда задач, в частности задач построения всех тупиковых ДНФ. Эта задача сначала должна быть сформулирована как задача покрывания и нахождение всех тупиковых подпокрываний.

Пусть  $N$  — это конечное множество, которое состоит из следующих элементов:

$$N = \{\alpha_1, \dots, \alpha_s\}$$

$$n = \{N_1, \dots, N_p\}$$

$n$  — это покрытие множества  $N$ .

$$N_i \subseteq N$$

$$\bigcup_{i=1}^p N_i = N$$

$(N, n)$  сопоставляются в булевскую матрицу  $M$ .

$$(N, n) \rightarrow M \in B^{p,s}$$

$$M \langle i, j \rangle = 1 \leftrightarrow \alpha_j \in N_i$$

	$\alpha_1$	...	$\alpha_j$	...	$\alpha_s$
$N_1$					
...					
$N_i$			$M \langle i, j \rangle$		
...					
$N_p$					

Рис. 3.7. Матрица  $M$

Предполагается, что матрица не имеет нулевых столбцов. Строка матрицы  $M$  с номером  $i$  покрывает столбец с номером  $j$  тогда и только тогда, когда  $M \langle i, j \rangle = 1$ . Система строк матрицы  $M$  с номерами из множества  $I \subseteq [1, p]$  покрывает матрицу  $M$  тогда и только тогда, когда любой столбец  $M$  покрывается строкой с номером из этого множества  $I$ .

Система множеств  $\{N_i; i \in I\}$  образует покрытие множества  $N$ . При этом оба покрытия считаются тупиковыми тогда и только тогда, когда в этих покрытиях нет "лишних" строк и подмножеств. Необходимо выделить все тупиковые покрытия матрицы  $M$ . Для этого используются монотонные функции, которые называются монотонными функциями покрытия.

Вводятся вспомогательные переменные  $y$ . Если переменная принимает значение 1, то соответствующая ей строка включается в покрытие.

$$\forall \beta \in B^p \rightarrow I = \beta = \{i \in [1, p] : \beta_i = 1\}$$

	$\alpha_1$	...	$\alpha_j$	...	$\alpha_s$	$\beta = (\beta_1, \dots, \beta_n)$
$N_1$						$y_1$
...						
$N_i$			$M_{\langle i, j \rangle}$			$y_i$
...						
$N_p$						$y_p$

Рис. 3.8. Матрица  $M$  с вспомогательными переменными

Пусть  $F(y_1, \dots, y_p)$  — это ФАЛ покрытия для матрицы  $M$  тогда и только тогда, когда:

$$\forall \beta : F(\beta) = 1$$

Это происходит тогда, когда  $I_\beta$  — это покрытие матрицы  $M$ .

### Свойства функции

- 1) Функция  $F$  — монотонная функция и монотонно зависит от своих переменных  $y$ .
- 2)  $\beta \in N_F^+$  тогда и только тогда, когда  $I_\beta$  — это тупиковое покрытие матрицы  $M$ .

## Построение всех тупиковых ДНФ

**Утверждение 3.4.** Функция покрытия  $F(y_1, \dots, y_p)$  матрицы  $M$ ,  $M \in B^{p \times s}$ , без нулевых столбцов задает КНФ вида:

$$F(y_1, \dots, y_p) = \bigwedge_{j=1}^s \left( \bigvee_{\substack{i \leq y \leq p \\ M(i,j)=1}} y_i \right)$$



В результате раскрытия скобок и приведения подобных из этой КНФ можно получить сокращенную ДНФ ФАЛ  $F(y)$ , простые импликанты которой взаимно однозначно соответствуют тушиковым покрытиям матрицы  $M$ .

Рассматривается элементарная дизъюнкция:

$$\forall j = 1, \dots, s \quad J_j = \bigvee_{i: M_{\langle i, j \rangle} = 1} y_i$$

$J_j = 1$ , когда столбец  $j$  покрывает  $I_\beta$ .

## Лекция 4. Задача минимизации ДНФ

### Функционал сложности ДНФ. Примеры

Пусть  $\psi$  — функционал сложности (ДНФ, схемы функциональных элементов (СФЭ), контактные схемы (КС)), т.е. для любой ДНФ  $a$  определяется ее  $\psi(a)$  сложность. Предполагается, что есть число, которое имеет 2 свойства:

- 1) Неотрицательность:

$$\psi(a) \geq 0$$

- 2) Монотонность. Для любой ДНФ  $a'$ , получающийся из ДНФ  $a$  удалением букв или элементарных конъюнкций:

$$\psi(a') \leq \psi(a)$$

Рассматриваются следующие функционалы:

- $R(a)$  — ранг ДНФ  $a$ .
- $\lambda(a)$  — длина ДНФ  $a$ .
- $L_1(a)$  — число букв, входящий в ДНФ  $a$  с отрицаниями.

### Пси-сложность, Пси-оптимальность

Для каждой ДНФ есть ее  $\psi$  сложность.  $\psi$  сложность записывается следующим образом:

$$\forall f \in P_2(n) : \psi(f) = \min \psi(a')$$

Минимум берется по ДНФ  $a'$ , реализующим функцию  $f$ .  $\psi$  сложность достигается на ДНФ  $a$ , реализующая функцию  $f$ . Таким образом  $\psi(f)$  —  $\psi$  сложность функции  $f$ . ДНФ  $a$  — это  $\psi$ -оптимальная ДНФ. Из свойств монотонности вытекает, что  $\psi$ -оптимальную ДНФ всегда можно найти среди тупиковых ДНФ.

$R$ -оптимальная ДНФ по рангу называется минимальной ДНФ. Минимальная ДНФ обязательно является тупиковой ДНФ.

$\lambda$ -оптимальную ДНФ по длине тоже можно найти среди тупиковых ДНФ.

## Задача минимизации ДНФ в общем виде. Общая схема её решения. Пример

Дана функция  $f$  и функционал сложности ДНФ, и требуется найти  $\psi$ -оптимальную ДНФ  $a$ , реализующая данную функцию  $f$ . Следовательно, достаточно просмотреть все тупиковые ДНФ, реализующая функцию  $f$ , и найти среди них минимальную ( $\psi$ -оптимальную ДНФ  $a$ ).

Любая тупиковая ДНФ состоит только из простых импликант. Поэтому сначала строится сокращенная ДНФ, в которой содержатся все простые импликанты. Далее необходимо построить ДНФ суммы тупиковых ( $\sum T$ ). В сумму тупиковых входят только те импликанты, которые встречаются хотя бы в одной тупиковой ДНФ. Из суммы тупиковых ДНФ можно построить перебор  $(a_1, \dots, a_\mu)$ , среди которых можно найти оптимальные ДНФ.

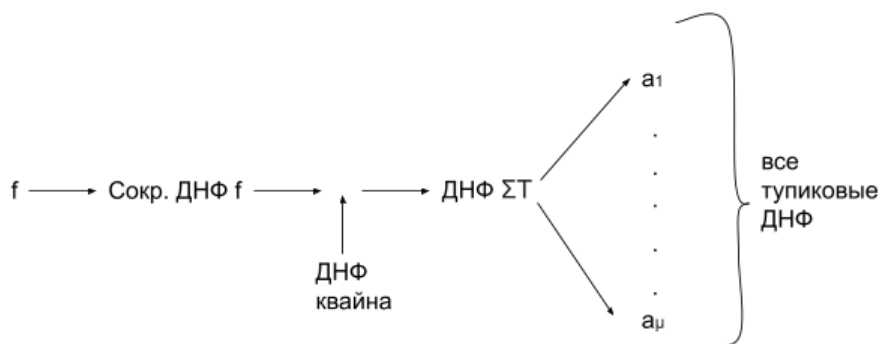


Рис. 4.1. Общая схема решения задачи минимизации ДНФ

## Функция Шеннона. Примеры для длины и ранга, их точные значения. Разложение Шеннона

Рассматривается не произвольная функция, а функция Шеннона. Вводится  $\psi(n)$  — максимальное значение  $psi$  сложности по всем функция от  $n$  переменных.

$$\psi(n) = \max_{f \in P_2(n)} \psi(f)$$

Это есть функция Шеннона для  $psi$  сложности ДНФ. Такие функции также

вводятся для других моделей вычислений.

$$\lambda(n) = \max_{f \in P_2(n)} \lambda(f)$$

$$R(n) = \max_{f \in P_2(n)} R(f)$$

**Утверждение 4.1.**

$$\forall n = 1, 2, \dots \quad \lambda(n) = 2^{n-1}$$

$$\forall n = 1, 2, \dots \quad R(n) = n \cdot 2^{n-1}$$

Это единственная модель ДНФ, где можно легко найти точные значения функции Шеннона.

Известно, что у функции  $g(x_1x_2x_3)$  5 тупиковых ДНФ. Ранги всех элементарных конъюнкций, входящих в сокращенную ДНФ функции, записывается как:

$$R(g) = 6$$

А длина:

$$\lambda(g) = 3$$

$$\lambda(3) \geq 3$$

$$R(3) \geq 6$$

**Доказательство.**

Сначала необходимо получить нижние оценки. Для этого вводится функция, которая была бы сложной по функционалу длина и ранга. Поэтому берется линейная функция от  $n$  переменных:  $l_n = x_1 \oplus \dots \oplus x_n$ . У этой функции есть единственный ДНФ. Следовательно:

$$\lambda(l_n) = 2^{n-1}$$

$$R(l_n) = n \cdot 2^{n-1}$$

Совершенная ДНФ — это единственная ДНФ для линейной функции  $l_n$ . Так как была найдена функция, у которой значение длины и ранга достигает двух

значений  $\lambda(n)$  и  $R(n)$ , то можно утверждать, что эти значения  $\lambda(l_n)$  и  $R(l_n)$  являются нижней оценкой функции Шеннона.

$$\lambda(l_n) \geq 2^{n-1}$$

$$R(l_n) \geq n \cdot 2^{n-1}$$

■

### Доказательство.

Необходимо доказать, что нижние оценки можно обеспечить для любой функции  $n$  переменной. Тем самым можно получить соответствующие верхние оценки.  $\forall f \in P_2(n)$  существует ДНФ  $a$ , реализующая функцию  $f$  и такая, что:

$$\lambda(a) \leq 2^{n-1}$$

$$R(a) \leq n \cdot 2^{n-1}$$

Чтобы доказать это утверждение, необходимо использовать разложение Шеннона.

$$f(x_1, \dots, x_q, x_{q+1}, \dots, x_n) = \bigvee_{\sigma'' \in B^{n-q}(x'')} K_{\sigma''}(x'') \cdot f_{\sigma''}(x'')$$

$$q = 1, \quad f(x_1, \dots, x_n) = \bigvee_{\sigma'' = (\sigma_2, \dots, \sigma_n) \in B^{n-1}} x_2^{\sigma_2} \dots x_n^{\sigma_n} \times f_{\sigma''}(x_1)$$

В качестве набора  $x'$  берется только одна первая переменная. Функция от одной переменной может принимать 4 значений.

$$f_{\sigma''}(x_1) \in \{0, 1, x_1, \bar{x}_1 + 1\}$$

При значениях 0 и 1 получаются элементарные конъюнкции ранга  $n - 1$ . При значениях  $x_1$  и  $\bar{x}_1$  получаются элементарные конъюнкции ранга  $n$ .

$$\hat{a}_f : \lambda(\hat{a}_f) \leq 2^{n-1}$$

Тем самым была построена ДНФ:

$$\lambda(a) \leq 2^{n-1}$$

Следовательно, для любой функции и для функции Шеннона было доказано:

$$\lambda(f) \leq 2^{n-1} \leftrightarrow \lambda(n) \leq 2^{n-1}$$

■

## Отображение набора $k$ -мерного единичного куба на набор независимых случайных величин

Рассматриваются типичные значения. Записывается набор из  $k$  независимых случайных величин, принимающих значение 0 и 1 с вероятностью  $\frac{1}{2}$ :

$$B^k \leftrightarrow \xi = (\xi_1, \dots, \xi_k)$$

$\forall A \subseteq B^n$  вероятность того, что реализация случайной величины  $\xi$  входит в множество  $A$  равна:

$$\frac{|A|}{2^n}$$

Рассматривается сумма этих независимых величин, математическое ожидание ( $E(\theta)$ ) и дисперсия ( $D(\theta)$ ):

$$\theta = \xi_1 + \dots + x_k$$

$$E(\theta) = \frac{k}{2}$$

$$E(\xi_i) = \frac{1}{2}$$

$$D(\theta) = \frac{k}{4}$$

$$D(\xi_i) = \frac{1}{4}$$

Поскольку эти величины независимы, то дисперсия суммы независимых случайных величин совпадает сумме дисперсии этих величин.

Неравенство Чебышева дает оценку вероятности больших отклонений.

$$P\{|\theta - E\theta| > t\} \leq \frac{D(\theta)}{t^2}$$

Если применить его к случайным величинам, то можно написать следующее соотношение:

$$t = m\sqrt{k}, \quad P\{|\theta - E\theta| > t\} \leq \frac{1}{4m^2} \xrightarrow{m \rightarrow \infty} 0$$
$$m = m(k)$$

## Лекция 5. Сложность задачи минимизации ДНФ

$\lambda(f)$  — это минимальная длина, где минимум берется по всем ДНФ, реализующим функцию  $f$ .

$$\lambda(f) = \min \lambda(a)$$

$\lambda(n)$  — это максимальная длина, где максимум берется по всем функциям, зависящим от  $n$ -переменных.

$$\lambda(n) = \max \lambda(f), \quad f \in P_2(n)$$

Ранг функции  $f$  — это минимальный ранг ДНФ, где минимум берется по всем ДНФ, реализующим функцию  $f$ .

$$R(f) = \min R(a)$$

Ранг функции  $n$  — это максимальный ранг, где максимум берется по всем функциям, зависящим от  $n$ -переменных.

$$R(n) = \max R(f), \quad f \in P_2(n)$$

**Утверждение 5.1.** Для любого  $n, n \in N$ , имеют место соотношения:

$$\lambda(n) = 2^{n-1}, \quad R(n) = n \cdot 2^{n-1}$$

Это утверждение доказывается с помощью нахождения нижней оценки. Для этого берется пример функции, для которой ДНФ  $\lambda \leq 2^{n-1}$  для данной функции не существует.

$$f = x_1 \oplus \dots \oplus x_n$$

У этой функции только одна ДНФ — совершенная ДНФ. Нижняя оценка для этой функции записывается следующим образом:

$$\lambda(l_n) = 2^{n-1} \rightarrow \lambda(n) \geq 2^{n-1}$$

Ранг совершенной ДНФ записывается как:

$$R(l_n) = n \cdot 2^{n-1} \rightarrow R(n) \geq n \cdot 2^{n-1}$$



Верхние оценки рассматриваются при разложении функции Шеннона. Для каждой функции  $f$  строится ее специальная ДНФ:

$$\forall f \in P_2(n) : \hat{a}_f = \bigvee_{\sigma'' = (\sigma_2 \dots \sigma_n) \in B^{n-1}} x_2^{\sigma_2} \dots x_n^{\sigma_n} \cdot f_{\sigma''}(x_1)$$

$$f_{\sigma''}(x_1) \in \{0, 1, x_1, \bar{x}_1\}$$

В результате получается, что  $\lambda(\hat{a}_f)$  — число тех наборов  $\sigma'' \in B^{n-1}$  таких, что:

$$f_{\sigma''}(x_1) \neq 0 \rightarrow \lambda(\hat{a}_f) \leq 2^{n-1}$$

Таким образом, были получены верхние оценки.

Рассматривается типичная функция при использовании неравенства Чебышева, которое применимо к наборам независимых случайных величин, принимающих значение 0 и 1 с вероятностью  $\frac{1}{2}$ .

$$B^k \leftrightarrow \xi = (\xi_1, \dots, \xi_k)$$

Такие векторные случайные величины сопоставляются наборам единичного куба, считая, что каждый набор единичного куба есть некоторая реализация случайной величины, и вероятность этой реализации равна  $\frac{|A|}{2^n}$ . Из этого делается следующий вывод: чтобы оценить мощность множество наборов в единичном кубе, достаточно знать вероятность того, что эта случайная величина примет свое значение из этого множества. Математическое ожидание и дисперсия этих величин записывается следующим образом:

$$E\xi_i = \frac{1}{2} \rightarrow E\theta = \frac{k}{2}$$

$$D\xi_i = \frac{1}{4} \rightarrow D\theta = \frac{k}{4}$$

$$P\{|\theta - E\theta| > t\} \leq \frac{D\theta}{t^2}$$

Следовательно, доля тех наборов  $\gamma$  из единичного куба  $\gamma \in B^n$ , для которых число единиц — это вес набора:

$$\frac{k}{2} - m\sqrt{k} \leq \|\gamma\| \leq \frac{k}{2} + m\sqrt{k} \leq \frac{1}{4m^2} \xrightarrow{m \rightarrow \infty} 0$$

## Неравенства для длины и ранга и их доказательства. Эффект Шеннона

**Утверждение 5.2.** Для почти всех ФАЛ  $f \in P_2(n)$ , выполняются неравенства:

$$\lambda(f) \leq \frac{3}{4} 2^{n-1} \left(1 + O\left(n \cdot 2^{-\frac{n}{2}}\right)\right) \quad (5.1)$$

$$R(f) \leq \frac{3}{4} n \cdot 2^{n-1} \left(1 + O\left(n \cdot 2^{-\frac{n}{2}}\right)\right) \quad (5.2)$$

**Доказательство.**

Достаточно доказать, что доля тех функций  $f$ , у которых:

$$\lambda(\hat{a})_f > \frac{3}{4} 2^{n-1} + m \cdot 2^{\frac{1}{2}} \xrightarrow{m \rightarrow \infty} 0 \quad m = n$$

удовлетворяет неравенству (5.1). Отсюда будет следовать, что ранг удовлетворяет неравенству (5.2).

Для этого, необходимо рассматривать  $\hat{a}_f$  как сумму некоторых случайных величин. Вводятся случайные величины, связанные с единичным кубом размерности  $2^n$ .

$$P_2(n) \leftrightarrow B^{2^n} \leftrightarrow \xi = (\xi_{(0\dots 0)}, \xi_{(0\dots 01)}, \dots, \xi_{(1\dots 1)})$$

Вместо случайных величин рассматриваются следующие величины:

$$\forall \sigma'' \in B^{n-1} \rightarrow \eta_{\sigma''} = \xi_{(0, \sigma'')} \vee \xi_{(1, \sigma'')} = 0 \leftrightarrow f_{\sigma''}(x_1) \equiv 0$$

Записывается математическое ожидание и дисперсия:

$$E\eta_{\sigma''} = \frac{3}{4}$$

$$D\eta_{\sigma''} = \frac{3}{16}$$

Сумма случайных величин имеет следующую формулу:

$$\Lambda = \sum_{\sigma''} \eta_{\sigma''} \leftrightarrow \lambda(\hat{a}_f)$$

Сумма математического ожидания и дисперсии записывается как:

$$E\Lambda = \frac{3}{4}2^{n-1}$$

$$D\Lambda = \frac{3}{16}2^{n-1}$$

Применяется неравенство Чебышева:

$$t = n2^{n-1}$$

$$P\{|-E\Lambda| > t\} \leq \frac{D}{t^2} \leq \frac{3}{16n^2} \xrightarrow{n \rightarrow \infty} 0$$

Это означает, что для почти всех функций верно обратное неравенство.

$$\lambda(\hat{a}_f) \leq \frac{3}{4}2^{n-1} + n \cdot 2^{\frac{n}{2}} \leq \frac{3}{16n^2} \rightarrow \lambda(f) \leq \frac{3}{4}2^{n-1} + n \cdot 2^{\frac{n}{2}}$$

В результате, получается, что для почти всех функций верно неравенство (5.1). Выяснилось, что у типичной функции значение этих параметров не имеет эффект Шеннона. Типичное значение асимптотически меньше, чем сама функция Шеннона. ■

**Задача:** Необходимо доказать, что для почти всех функций  $f$ ,  $f \in P_2(n)$ :

$$\lambda(f) = \bar{o}(2^n)$$

Доля тех ФАЛ  $f$ ,  $f \in P_2(n)$ , для которых неверное неравенство (5.1), стремится к 0 с вероятностью  $\frac{3}{16n^2}$ .

## Параметры трудоёмкости задачи минимизации ДНФ

Задача минимизации ДНФ ставится следующим образом: по заданной функции  $f$  и функционалу сложности  $\Psi$  построить  $\Psi$  минимальную ДНФ, реализующую функцию  $f$ . Сначала строится сокращенная ДНФ, строится функция  $f$  и ДНФ Квайна. Потом строится сумма тупиковых ДНФ. Дальше начинается перебор, связанный с построением тупиковых ДНФ, среди которых всегда есть  $\Psi$  минимальная ДНФ. Достаточно перебрать все тупиковые и найти среди них нужную  $\Psi$  минимальную ДНФ, реализующую функцию  $f$ .

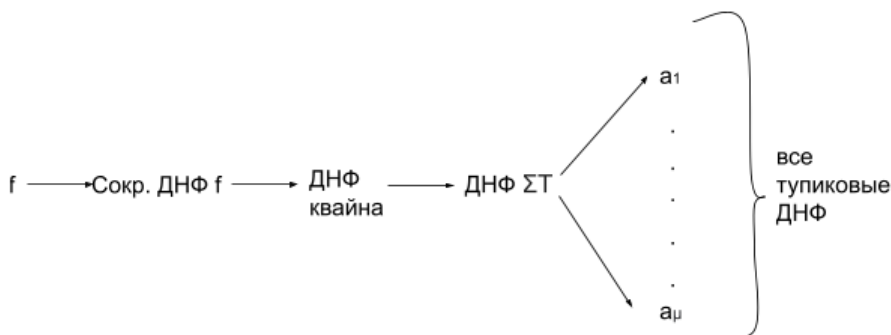


Рис. 5.1. Задача минимизации

Можно рассматривать разные параметры, характеризующие трудоемкость задачи. Одним из этих параметров является ранг окрестности данной грани, которую нужно просмотреть, чтобы посмотреть входит ли она в нужную категорию грани. Чтобы построить ДНФ суммы тупиковых, достаточно рассматривать грань окрестности порядка 2. Один из критериев оптимальности — это порядок окрестности грани. Критерий вхождения в ДНФ суммы минимальных не может ограничиваться каким-то конечным порядком.

Длину сокращенной ДНФ ( $\lambda_{\text{сокp}}(f)$ ), число тупиковых ДНФ ( $\tau(f)$ ) и число минимальных ДНФ ( $\mu(f)$ ) можно рассматривать как параметры, характеризующие трудоемкость задачи минимизации. Также можно рассматривать функции Шеннона, как параметры, характеризующие трудоемкость задачи минимизации функции из  $P_2(n)$ :

$$\lambda_{\text{сокp}}(n) = \max_{f \in P_2(n)} \lambda(f)$$

$$\tau(n) = \max_{f \in P_2(n)} \tau(f)$$

$$\mu(n) = \max_{f \in P_2(n)} \mu(f)$$

### Утверждение о числе тупиковых (минимальных) ДНФ и его следствие

**Утверждение 5.3.** Число тупиковых (минимальных) ДНФ у ФАЛ  $f$  из  $P_2(n)$ ,  $n \geq 4$ , вида:

$$f(x_1, \dots, x_n) = g(x_1, x_2, x_3) \cdot (x_4 \oplus \dots \oplus x_n)$$

где  $\bar{N}_g = \{(000), (111)\}$ , равно  $5^{2^{n-4}}$  (соответственно  $2^{2^{n-4}}$ ).

В следствии, получаются следующие неравенства:

$$\tau(n) \geq 5^{2^{n-4}}$$

$$\mu(n) \geq 2^{2^{n-4}}$$

**Доказательство.**

Сокращенная ДНФ функции  $g$  устроена как цикл из 6 ребер, проходящих по двум средним слоям четырехмерного куба.

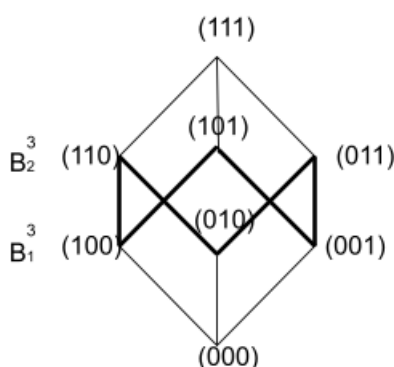


Рис. 5.2. Сокращенная ДНФ функции  $g$

Линейная функция обращается в 1, когда число слагаемых в этой сумме нечетное. Следовательно, сокращенная ДНФ функции  $f$  имеет следующий вид:

$$a = \bigvee_{(\sigma_4 \dots \sigma_n) \in B_{\text{неч.}}^n} x_4^{\sigma_4} \dots x_n^{\sigma_n} \left( \bigvee_{i=1}^6 K_i \right) = \bigvee_{i=1}^6 \bigvee_{(\sigma_4 \dots \sigma_n) \in B_{\text{неч.}}^n} x_4^{\sigma_4} \dots x_n^{\sigma_n} \cdot K_i$$

Чтобы взять тупиковую ДНФ, то надо в каждом из этих циклов взять свою тупиковую ДНФ (5 вариантов выбора для каждого цикла).

$$\tau(f) = 5^{2^{n-4}}$$

В каждом таком цикле можно выбирать 2 минимальные ДНФ:

$$\mu(f) = 2^{2^{n-4}}$$

■



Рис. 5.3. Геометрическое представление сокращенной ДНФ функции  $f$

## Нижняя оценка длины сокращённой ДНФ.

### Симметрическая ФАЛ, примеры

**Утверждение 5.4.** Нижняя оценка длины сокращенной ДНФ записывается следующим образом:

$$\lambda_{\text{сокр}}(n) \geq e_1 \frac{3^n}{n}$$

где  $e_1$  — некоторая константа.

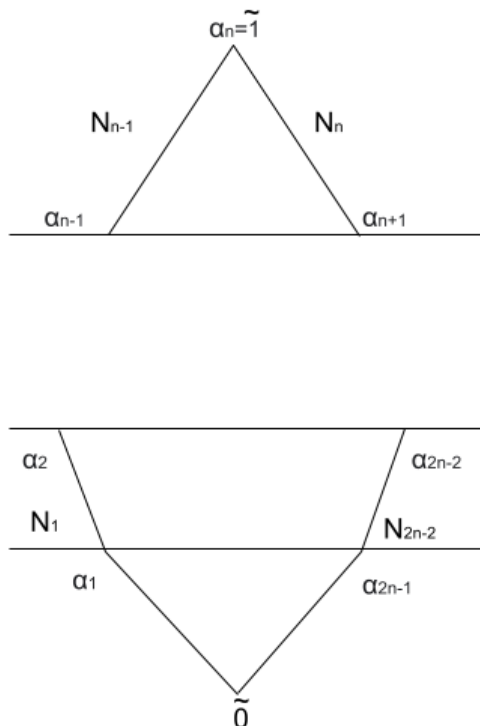


Рис. 5.4. Цепная ФАЛ длины  $(2n - 2)$  в кубе  $B^n$

### Доказательство.

Пусть  $f \in P_2(n)$  — симметрическая функция тогда и только тогда, когда значение функции  $f(\beta) = f(\gamma)$ , если  $\beta$  и  $\gamma$  получаются друг из друга перестановками Булевских переменных. Число 1 в этих наборах одинаковое. Это происходит тогда, когда функция  $f$  принимает одинаковое значение на любых наборах с равным числом 1.

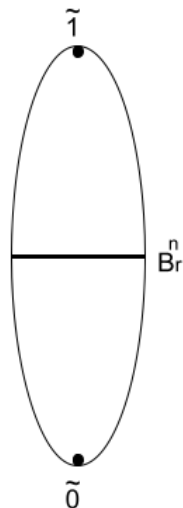


Рис. 5.5. Симметрическая функция

Число симметрических функций в  $P_2(n)$  равно  $2^{n+1}$ . Далее необходимо показать слои, на которых симметрическая функция равна 1. Номера этих слоев называются рабочими числами симметрической функции. Следовательно, характеристическое множество функции можно представить как объединение нескольких слоев.

$$N_f = B_{i_1}^n \cup B_{i_2}^n \cup \dots \cup B_{i_s}^n, \quad 0 \leq i_1 < i_2 < \dots < i_s \leq n$$

Таким образом, симметрическая функция записывается как:

$$f = S_n^I, \quad I = \{i_s, \dots, i_r\}$$

■

**Пример 5.1.** Пусть функция  $g$  — симметрическая функция:

$$g = S_3^{\{1,2\}}$$

Функция голосования записывается следующим образом:

$$H = S_3^{\{2,3\}}$$

Линейная функция — это есть:

$$l_n = S_n^{\{1,3,5\dots\}}$$

$$\bar{l}_n = S_n^{\{0,2,4\dots\}}$$

Рассматриваются поясковые симметрические функции:

$$0 \leq r \leq p \leq n$$

$$N_f = S_n^{[r,p]}$$

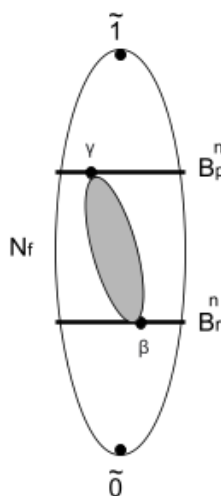


Рис. 5.6. Поясковая симметрическая функция

Любая максимальная грань функции задается двумя наборами  $\beta$  и  $\gamma$ , которые сравнимы между собой.

$$\beta \in B_p^n \quad \gamma \in B_r^n \quad \beta \leq \gamma$$



Сокращенная ДНФ функции  $f$  устроена следующим образом:

$$a = \bigvee_{1 \leq i_1 < \dots < i_{n-p+r}} x_{i_1}^{\sigma_1} \dots x_{i_{n-p+r}}^{\sigma_{n-p+r}}, \quad \sigma_1 \dots \sigma_{n-p+r} \in B_r^{n-p+r}$$

Дальше необходимо найти длину сокращенной ДНФ:

$$\lambda(a) = C_n^{n-p+r} \cdot C_{n-p+r}^r$$

Таким образом, чтобы найти точную нижнюю оценку длины, надо найти значение  $r$  и  $p$ .

$$r = n - p = \left\lfloor \frac{n}{3} \right\rfloor$$

$$p - r \approx \frac{n}{3}$$

Записывается формула Стирлинга:

$$n! \approx \frac{n^n e^{-n}}{\sqrt{2\pi n}}$$

В результате получается:

$$\lambda_{\text{сокp}}(n) \geq e_1 \frac{3^n}{n}$$

## Теорема Журавлёва о ДНФ сумма минимальных. Цепная ФАЛ, примеры

Необходимо найти размерность окрестности грани, чтобы найти входит ли она в ДНФ суммы минимальных. Эта размерность окрестности линейно растет с ростом  $n$ .

**Утверждение 5.5.** При любом  $n \in \mathbb{N}$ ,  $n \geq 3$  в  $P_2(n)$  существуют ФАЛ  $f'$ ,  $f''$ , имеющие общую простую импликанту  $K$ , которая входит в ДНФ  $\sum M$  одной, но не входит в ДНФ  $\sum M$  другой из этих ФАЛ и для которой:

$$S_{n-3}(N_K, f') = S_{n-3}(N_K, f'')$$

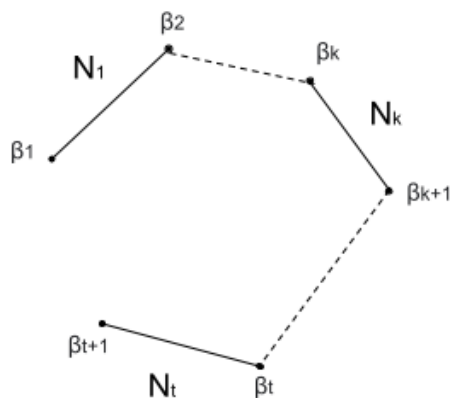


Рис. 5.7. Сокращенная ДНФ функции  $f$

$f \in P_2(n)$  — цепная функция длины  $t$  тогда и только тогда, когда сокращенная ДНФ функции  $f$  (с геометрической точки зрения) — это цепь из  $t$  ребер куба  $B^n$ .

$$N_f = \{\beta_1 \dots \beta_{t+1}\}$$

$$N_i = \{\beta_i \dots \beta_{i+1}\}$$

$K_i$  — это соответствующая элементарная конъюнкция  $N_i$ .

$$a_{\text{сокр}} = K_1 \vee \dots \vee K_t$$

Если  $t$  — нечетное число, то функция имеет единственную минимальную ДНФ (дизъюнкция всех ребер с нечетными номерами):

$$t = 2k - 1 \geq 5 \rightarrow f = K_1 \vee K_3 \vee K_5 \vee \dots \vee K_t = \sum M$$

**Доказательство.**

Необходимо построить в  $B^n$  цепную ФАЛ  $f$  четной длины  $2k$ . Вводятся 2 функции:

$$f' : N_{f'} = N_f \setminus \{\beta_1\}$$

$$f'' : N_{f''} = N_f \setminus \{\beta_{2k+1}\}$$

Для одной из этих функций среднее ребро  $N_k$  будет четным, а для другой функции нечетным. Окрестности у функций  $f'$  и  $f''$  будут одинаковыми.

Необходимо найти цепную функцию нужной длины в кубе  $B^n$  и параметр  $K$ .

$$2k = 2n - 2$$

Окрестности порядка  $(n - 3) N_K$  относительно  $f'$  и  $f''$  будут совпадать. ■

Из теоремы следует, что критерий вхождения ЭК в ДНФ  $\Sigma M$  не имеет такого локального характера, как критерий вхождения ЭК в ДНФ  $\Sigma T$ .

Известно, что при  $n \geq 14$  в  $P_2(n)$  имеется цепная ФАЛ четной длины  $t$ ,  $t \geq 2^{n-11} - 4$ , на основе которой справедливость теоремы можно установить для окрестности порядка  $(\frac{t}{2} - 2)$ .

## Лекция 6. Формула алгебры логики

### Базис, функциональный элемент, входной и выходной алфавиты

Пусть есть счетный упорядоченный алфавит входных Булевых переменных (БП):

$$X = \{x_1, \dots, x_n, \dots\}$$

и базисная система, т.е. полная в  $P_2$  системе ФАЛ.

$$B = \{\varphi_1, \dots, \varphi_\sigma\}$$

Под базисом подразумевается полная система, в которой нет лишних с точки зрения полноты функций.

$$B_0 = \{x_1 \cdot x_2, x_1 \vee x_2, \bar{x}_1\}$$

Предполагается, что  $\varphi_i$  существенно зависит от всех своих Булевских переменных при условии, что  $k_i \geq 2$ :

$$\varphi_i = \varphi_i(x_1, \dots, x_{k_i})$$

$\varphi_i$  рассматривается как функциональный символ. В схемных представлениях  $\varphi_i$  рассматривается как функциональный элемент  $\varepsilon_i$ . Удобнее всего будет рассматривать это представление как вершину графа  $\varphi_i$ .

Записывается счетный упорядоченный алфавит выходных БП, с помощью которых можно помечать те вершины, значение которых будет выходить из системы на другие схемы:

$$Z = \{z_1, \dots, z_{m_1} \dots\}$$

### Индуктивное (по глубине) определение формулы

В алгебре логики считалось, что сама переменная не считается формулой, а из схемных содержательных соображений удобно считать, что сама переменная

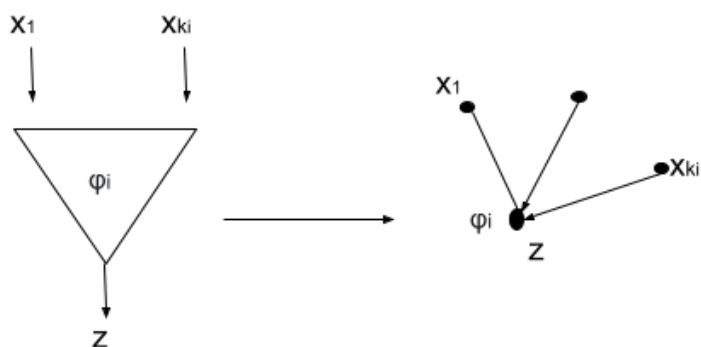


Рис. 6.1. Представление функционального элемента

тоже есть простейшая формула глубины 0. Формула — это запись из функциональных символов функции  $\varphi_i$  и из входных переменных  $X$ , которая показывает как какие функции подставляются в какие другие функции вместо каких других переменных. Формула глубины 0 записывается следующим образом:

$$F = x_j$$

Формула глубины 0 реализует ФАЛ равной своей переменной  $x_j$ . Таким образом, это базис индукции. Далее записывается индуктивный переход. Пусть  $i \in [1, 6]$  — номер функционального элемента и  $F_1, \dots, F_{k_i}$  — формулы глубины  $q_1, \dots, q_{k_i}$ , реализующие функции  $f_1, \dots, f_{k_i}$ . Тогда запись формулы глубины  $q$  имеет следующий вид:

$$F = \varphi_i(F_1, \dots, F_{k_i})$$

$$q = \max\{q_1, \dots, q_{k_i}\} + 1$$

Глубина формулы реализует следующую функцию:

$$\varphi_i(f_1, \dots, f_{k_i})$$

Все полученные таким образом записи называются формулами, и каждая из них реализует индуктивным образом определенную ФАЛ.

## Формула как граф. Пример

Базису индукции сопоставляется формула как граф. Вершина имеет переменную  $x_j$  в качестве своей пометки и переменную  $z_m$  как выходную переменную. Это есть простейшая формула, которая в привычном виде записывается как формула:

$$F = x_j$$

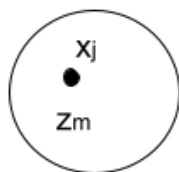


Рис. 6.2. Дерево формулы  $x_j$

При индуктивном переходе есть дерево формулы  $F_1$  и  $F_{k_i}$  с выходными переменными  $z_1$  и  $z_{k_i}$ , из которых можно построить дерево формулы  $F$  с выходной переменной  $z$  и пометкой  $\varphi_i$ . Далее убираются выходные переменные  $z_1$  и  $z_{k_i}$  и добавляется дуга  $1, \dots, k_i$ .

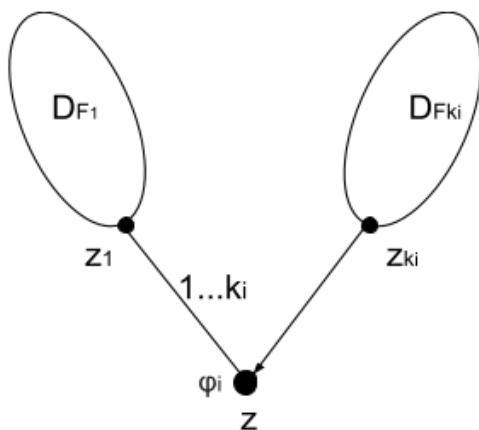


Рис. 6.3. Дерево формулы  $F$

Формула как запись и формула как граф взаимно однозначно соответствуют друг другу. Каждая формула представляется корневым помеченным деревом. Листья дерева имеют пометки из алфавита  $X$ . Каждая дуга, входящая в какую-то вершину, имеет числовую пометку, соответствующую номеру того входа, на которой она поступает.

**Пример 6.1.**

$$\overline{(x_1 \vee x_2)} \vee x_3 \vee (x_3(x_1 \vee x_2) \vee x_1x_2)$$

*Сопоставляются пометки листьев и строятся элементы дизъюнкции и конъюнкции.*

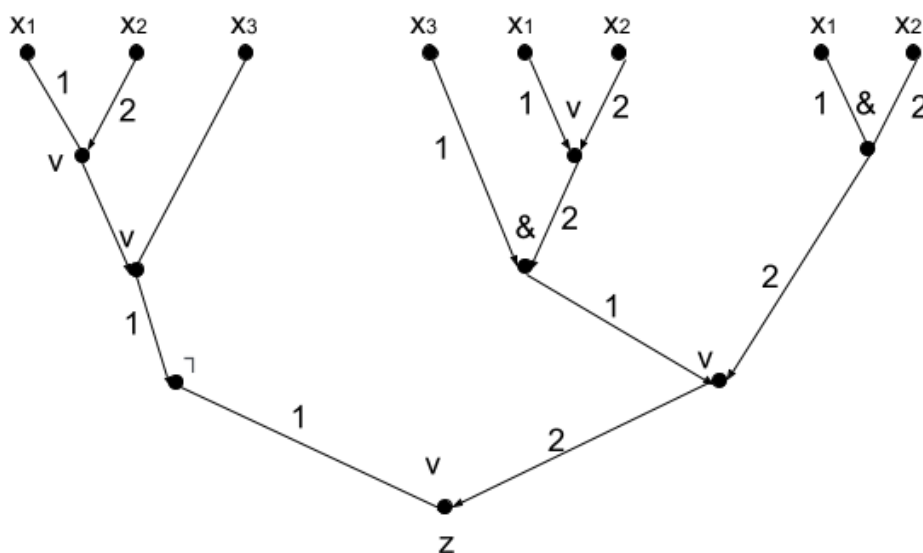


Рис. 6.4. Дерево формулы

## Функционалы сложности формул.

Множество всех формул в базисе  $B$  обозначается как  $U_B^F$ . Функционалы сложности формул бывают следующими:

- 1)  $R(F)$  — ранг  $F$  (число вхождений БП в  $F$ ) равен числу листьев  $D_F$ .
- 2)  $D(F)$  — глубина формулы  $F$  — это максимальное число вложенных друг в друга функциональных символов. На языке дерева это максимальная длина цепи в  $D_F$ .

3)  $L(f)$  — число функциональных символов в  $F$ . При интерпретации с помощью дерева, то это — число нелистовых вершин в  $D_F$ .

$L_{\&,\vee}(F)$  — рассматривается число только функциональных элементов конъюнкции и дизъюнкции в  $F$ .

$L_{\neg}(F)$  — рассматривается число функциональных символов отрицания в  $F$ .

Глубина — это время срабатывания схемы или время вычисления формулы с использованием параллельных процессоров для вычисления независимых друг от друга операций, входящих в запись формулы.

### Утверждение о связи функционалов сложности

Формулы в стандартном базисе обозначаются как:

$$U^{\Phi} = U_{B_0}^{\Phi}$$

**Утверждение 6.1.** Для формулы  $F$ ,  $F \in U^{\Phi}$ , выполняются неравенства

$$R(F) = L_{\&,\vee}(F) + 1 \leq L(F) + 1 \leq 2^{D(F)}$$

где  $L_{\&,\vee}(F)$  — число ФС  $\&$  и  $\vee$  в формуле  $F$ .

В следствии получается следующее:

$$D(F) \geq \lceil \log(L(F) + 1) \rceil$$

### Доказательство.

Необходимо доказать, что число элементов конъюнкции и дизъюнкции на единицу меньше, чем ранг формулы. Таким образом, удобно будет воспользоваться графовой моделью. Пусть  $G$  — это граф.

$$G = (V, E)$$

$V$  — это множество вершин графа.

$$V = V(G)$$



$E$  — это множество ребер графа.

$$E(G) = E$$

Пусть формуле  $F$  сопоставлено дерево  $D_F$ . Необходимо посчитать число ребер в этом дереве:

$$|E(D_F)| = R(F) + L_{\&, \vee}(F) + L_{\neg}(F) - 1 = 2L_{\&, \vee}(F) + L_{\neg}(F) \implies R(F) = L_{\&, \vee}(F) + 1$$

Второе неравенство доказывается индукцией по глубине формулы.

$$L(F) + 1 \leq 2^{D(F)}$$

■

## Эквивалентность и изоморфизм формул. Подформула.

### Принцип эквивалентной замены

Эквивалентность двух формул записывается следующим образом:

$$F'(x_{i_1}, \dots, x_{i_n}) = F''(x_{j_1}, \dots, x_{j_m}) \Leftrightarrow f_{F'} = f_{F''} \Leftrightarrow t : F' = F''$$

Две формулы называются эквивалентными, если они реализуют равные функции, у которых одни и те же существенные переменные, от которых задаются одинаковые отображения соответствующего куба значения этих переменных в значение функции.

Изоморфизм формул записывается следующим образом:

$$F' \approx F''$$

Это означает, что либо эти формулы одинаковые с точки зрения их записи  $F' = F''$ , либо соответствующие им деревья изоморфны как помеченные графы  $D_{F'} \approx D_{F''}$ . Если формулы изоморфны, то они эквивалентны.

$F'$  — подформула  $F$  тогда и только тогда, когда запись формулы  $F'$  представляет собой часть записи  $F$ , когда  $F'$  получается как промежуточная формула при индуктивном построении  $F$ , и когда  $D_{F'}$  — входное поддереве  $F$ .

$$F' \subseteq F$$

Принцип эквивалентной замены заключается в том, что если в формуле  $F$  найти подформулу  $F'$  и вставить ее в эквивалентную ей  $F''$ , то получится формула, эквивалентная исходной формуле. Следовательно,  $\hat{F}$ , получающаяся из  $F$  заменой  $F'$  на  $F''$  эквивалентна исходной формуле  $F$ .

$$F \mapsto \hat{F}$$

## Известные тождества формул. Преобразования подобия

Тождество коммутативности для конъюнкции:

$$t_{\&}^k : x_1 x_2 = x_2 x_1, \quad t_{\vee}^k = \dots$$

Тождество ассоциативности для конъюнкции:

$$t_{\&}^A : (x_1 x_2) x_3 = x_1 (x_2 x_3), \quad t_{\vee}^A = \dots$$

Тождество де Моргана для конъюнкции:

$$t_{\&}^M : \overline{x_1 \cdot x_2} = \bar{x}_1 \vee \bar{x}_2$$

Тождество де Моргана для дизъюнкции:

$$t_{\vee}^M : \overline{x_1 \vee x_2} = \bar{x}_1 \times \bar{x}_2$$

Тождество де Моргана для отрицания:

$$t_{\neg}^M : \bar{\bar{x}} = x$$

Операция отрицания — самая сильная операция. Потом идет операция конъюнкции, а дальше идут остальные операции, которые между собой одинаковы. Порядок их выполнения должен регулироваться скобками.

$$(|x_1) \cdot x_2) \vee x_3$$

С помощью этих тождеств можно проводить преобразование подобия. Две формулы называются подобными, если они получаются друг из друга с помощью преобразований на основе тождеств коммутативности ( $\tau^k$ ) и ассоциативности ( $\tau^A$ ).  $F'$  подобна  $F''$  тогда и только тогда, когда существует эквивалентное

преобразование  $F'$  в  $F''$  с помощью тождеств коммутативности ( $\tau^k$ ) и ассоциативности ( $\tau^A$ ).

$$F' \xrightarrow{\tau^k, \tau^A} F''$$

$$R(F') = R(F'')$$

$$L(F') = L(F'')$$

$$D(F') = D(F'')$$

Ни ранг, ни сложность заведомо не изменится. Только глубина может меняться. Поэтому с помощью преобразования подобия можно оптимизировать формулу по глубине.

### Формула с поднятыми отрицаниями

$F$  — формула с поднятыми отрицаниями тогда и только тогда, когда любой ФС в  $F$  располагается только над Булевыми переменными. Отрицание не может висеть над конъюнкцией или дизъюнкцией.

Предполагается, что есть произвольная формула не с поднятыми отрицаниями, тогда от этой формулы с помощью тождеств де Моргана ( $\tau^M$ ) можно перейти в формулу с поднятыми отрицаниями ( $\hat{F}$ ). При этом проводится эквивалентное преобразование.

Если переходить от формулы  $F$  в формулу  $\hat{F}$ , то ранг, сложность и глубина не меняются.

$$R(\hat{F}) = R(F)$$

$$L_{\&, \vee}(F) = L_{\&, \vee}(\hat{F})$$

$$D(\hat{F}) = D(F)$$

### Оптимизация подобных формул по глубине.

#### Альтернирование

**Утверждение 6.2.** Для любой формулы  $F$  с поднятыми отрицаниями из  $U^\Phi$  существует подобная ей формула  $\hat{F}$  такая, что:

$$D(\hat{F}) \leq [\log(L(F) + 1)] + Alt(F)$$

Альтернирование формулы — это максимально возможное число изменений типов конъюнкции и дизъюнкции в цепях формулы  $D_F$ .

Пусть  $F$  — это ЭК или ЭД, то его альтернирование записывается следующим образом:

$$Alt(F) = 0$$

Пусть  $F$  — это ДНФ или КНФ отличная от ЭД и ЭК, то альтернирование:

$$Alt(F) = 1$$

Для любой ЭК или ЭД  $K$  существует подобная ей формула  $\hat{K}$  такая, что:

$$D(\hat{K}) = [\log(L(K) + 1)]$$

которая минимальна по глубине.

Для любой ДНФ или КНФ  $U$  существует подобная ей формула  $\hat{U}$  такая, что:

$$D(\hat{U}) \leq [\log(L(U) + 1)] + 1$$

### Доказательство.

Индукция по  $Alt(F)$ . Если  $Alt(F) = 0$ , то  $F$  — это ЭК или ЭД.

$$F = x_1 \circ \dots \circ x_k \circ \bar{x}_{k+1} \circ \dots \circ \bar{x}_t$$

В результате получится следующая формула, показывающая минимальную возможную глубину:

$$\hat{F} : D(\hat{F}) \leq [\log(L(F) + 1)]$$

Следовательно,  $\hat{F}$  — оптимальная по глубине формула эквивалентная формуле  $F$ .

$$F = F_1 \circ \dots \circ F_t$$

$$Alt(F_j) \leq (a - 1)$$

Каждая из этих формул оптимизируется по индуктивному предположению по глубине.

$$\hat{F} = \tilde{F}(\hat{F}_1, \dots, \hat{F}_t)$$

$\tilde{F}$  оптимизируется по глубине с помощью расстановки скобок.

$$\tilde{F} = y_1 \circ \dots \circ y_t$$

■

В обоих случаях используется вспомогательное утверждение.

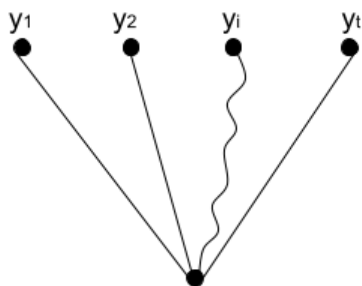


Рис. 6.5. Двоичное корневое дерево

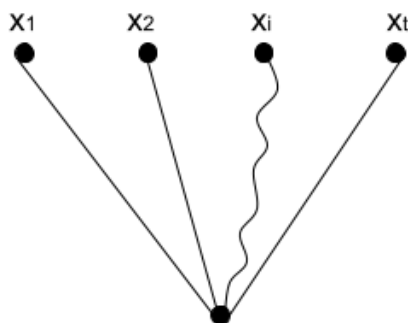


Рис. 6.6. Двоичное корневое дерево  $\hat{F}$

**Утверждение 6.3.** Пусть для любых целых неотрицательных чисел  $d \geq d_1 \geq \dots \geq d_t$  таких, что:

$$2^d \geq 2^{d_1} + \dots + 2^{d_t}$$

Тогда существует двоичное корневое дерево  $D$  с  $t$  листьями, в котором глубина:

$$y_i \leq d - d_i$$

Это утверждение применяется при индукции по  $Alt(F)$ .

$$L(F) + 1 = t + k = 2^1 + \dots + 2^1 + 2^0 + \dots + 2^0$$

$$d = [\log(t + k)]$$

$$d_1 = \dots = d_k = 1, d_{k+1} = d_t = 0$$

Таким образом, получается формула  $\hat{F}$ .

## Лекция 7. Схема из функциональных элементов 1

### Граф, сеть. Схема: изоморфность, эквивалентность, классы, сложность

Пусть задан граф  $G$ :

$$G = (V, E)$$

$$V = V(G)$$

$$E = E(G)$$

При этом ребра могут быть либо ориентированными — дуги, либо неориентированными.

Рассматривается сеть  $S$ . Определяются входные выходные вершины из вершин графа.  $S$  — это  $(p, q)$  сеть, если  $V'$  — это выборка без повторений, а  $V''$  — это выборка мощности  $q$ .

$$S = (G, V', V'')$$

$$V' = (v_1, \dots, v_p)$$

$$V'' = (v_1, \dots, v_q)$$

Граф  $G$  имеет входные вершины  $V'$  и выходные вершины  $V''$ . При построении различных схем рассматриваются два варианта графов: с ориентированными ребрами и с неориентированными ребрами.

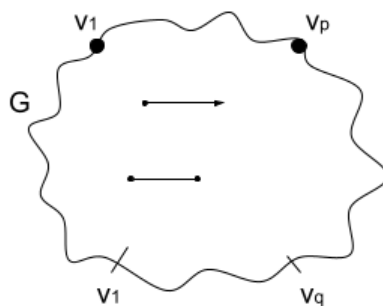


Рис. 7.1. Граф  $G$

Чтобы из графа получить схемы, помимо этих двух выборок необходимо пометить дуги и вершины графа некоторыми символами. Вершины входной выборки помечаются символами входных переменных:

$$V' \longrightarrow \{x_1, \dots, x_n\} = X$$

$$V'' \longrightarrow \{z_1, \dots, z_n\} = z$$

При этом такая схема реализует систему функций  $F$ , реализуемых на ее выходных вершинах.

$$F = (f_1, \dots, f_q, \dots, f_m) \quad m \geq q$$

Каждая входная вершина помечается в точности одним символом входной переменной  $X$ . А выходные вершины могут получать пометки несколькими символами. Поэтому система функций зависит от  $f_m$ .

Рассматриваются изоморфные схемы и эквивалентные схемы. Две схемы называются изоморфными, если они изоморфны как помеченные графы.

$$\Sigma' \approx \Sigma''$$

Схемы называются эквивалентными, если они реализуют одни и те же системы функций.

$$\Sigma' \sim \Sigma'' \Rightarrow F_{\Sigma'} = F_{\Sigma''}$$

Любые две изоморфные схемы всегда будут эквивалентны. Классы схем обозначаются через  $U$ . Тогда, количество попарно не изоморфных схем обозначается как  $|U|$ . Количество попарно не эквивалентных схем обозначается как  $||U||$ . В каждом из классов схем вводится понятие сложности —  $L(\varepsilon)$ . Тогда  $U(L, n)$  — это множество всех схем в классе  $U$ , сложность которых не превосходит  $L$  и который зависит от переменных  $x_1, \dots, x_n$ .

## Схема из функциональных элементов (СФЭ).

### Квазидерево. Переход от СФЭ к системе формул

#### (квазидеревьев)

Схема из функциональных элементов (СФЭ) — это граф с ориентированными дугами. Каждая внутренняя вершина СФЭ получает пометку некоторым



функциональным символом. Пусть есть базис — множество функциональных элементов:

$$B = \varepsilon_1, \dots, \varepsilon_b$$

Тогда, СФЭ -  $(n, m)$  сеть, у которой входы и выходы обозначены переменными.

$$V' - X = \{x_1, \dots, x_n\}$$

$$V' - Z = \{z_1, \dots, z_m\}$$

А внутренние вершины сети помечены функциональными символами функций, реализуемых элементами базиса.

$$\varphi_1 = \varphi_1(x_1, \dots, x_{k_1})$$

$$\varphi_b = \varphi_b(x_1, \dots, x_{k_b})$$

Внутренняя вершина  $v$  с полу степенью захода ( $d^+(v) = k$ ) помечена функциональным символом  $\varphi_i$ .

$$k_i = k$$

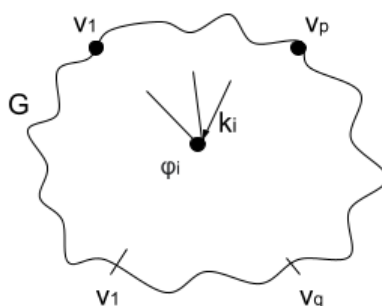


Рис. 7.2. Внутренняя вершина  $v$

Входами схемы являются истоки сетей. В дереве можно отождествить все листовые вершины с одинаковыми пометками. Тогда получится квазидерево. Граф такого вида будет схемой из функциональных элементов.

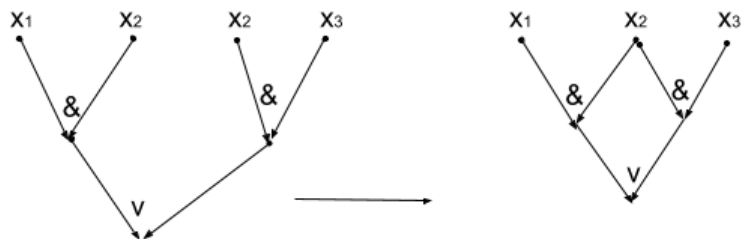


Рис. 7.3. Отождествление листовых вершин с одинаковыми пометками

$$U_B^\Phi \leq U_B^e$$

Таким образом, каждой формуле соответствует эквивалентная схема. Также можно сделать обратное преобразование. Пусть есть схема из функциональных элементов, по которой можно построить систему формул, реализующих функции, которые были на выходах схемы.

$$\Sigma \rightarrow F = (f_1, \dots, f_m)$$

### Функционирование СФЭ

- 1) Если входная вершина  $v_i$  имеет пометку  $x_i$ , то она реализует функцию  $x_i$ .
- 2) Если в вершинах  $v'_1, \dots, v'_{k_i}$  реализуются ФАЛ  $f'_1, \dots, f'_{k_i}$ , то вершина  $v$  реализует функцию  $\varphi_i(f'_1, \dots, f'_{k_i})$ .

При этом выходными вершинами схемы считаются вершины, помеченные выходными переменными.

Можно построить систему формул над базисом  $B$ , который реализует ту же систему функций.

$$\vec{f} = (F_1, \dots, F_n)$$

Этой системе формул соответствует множество деревьев.

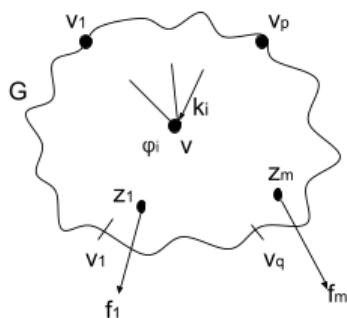


Рис. 7.4. Функционирование СФЭ

## Висячая вершина. Приведенная СФЭ. Функционалы сложности СФЭ

Висячая вершина — это вершина, из которой не выходит ни одна дуга.

$$d^-(v) = 0$$

При этом СФЭ называется приведенной, если она не содержит висячих вершин, отличных от выходов схемы. Любую СФЭ переделать в приведенную, последовательно удаляя висячие вершины, которые не являются выходами схемы.

Для схемы в базисе  $(\Sigma \in U_B^C)$  функционал сложности схемы  $L(\Sigma)$  — это число внутренних вершин.

$$\Sigma = \Sigma(X, Z) = \Sigma(x_1, \dots, x_n; z_1, \dots, z_m)$$

Функционал глубины  $D(\Sigma)$  — это наибольшая глубина ее главных цепей. Цепь называется главной, если ее первая вершина входная, а последняя вершины выходная. Тогда глубина цепи — это количество вершин в ней.

$$v_{i_1} \in V'$$

$$v_{i_s} \in V''$$

$$D(C) = S$$

Ранг схемы  $(R(\Sigma))$  определяется как количество дуг, исходящих из входных вершин.

## Утверждение о связи функционалов сложности у СФЭ. (Аналогично случаю формулы)

Стандартный базис — это:

$$B_0 = \{\&, \vee, \}\}$$

$$U_{B_0}^\Phi = U^\Phi$$

**Утверждение 7.1.** Для любой приведенной СФЭ  $\Sigma \in U^C$  с одним выходом справедливо следующее:

$$R(\Sigma) \leq L_{\&, \vee}(\Sigma) + 1 \leq L(\Sigma) + 1 \leq 2^{D(\Sigma)}$$

Это утверждение доказывается аналогично утверждению 6.1. Необходимо посчитать, сколько всего дуг входит во все вершины схемы.

$$2 \times L_{\&, \vee}(\Sigma) + L_{\lceil}(\Sigma) \leq R(\Sigma) + (L(\Sigma) - 1)$$

## Оценки количеств формул. Квазиизоморфность формул

**Утверждение 7.2.** Для любых натуральных чисел  $(\forall n, L, D \in \mathbb{N})$  выполняются неравенства:

- Количество попарно не изоморфных формул в стандартном базисе:

$$|U^\Phi(L, n)| \leq (10n)^{L+1}$$

- Количество попарно неэквивалентных формул:

$$\|U^\Phi(L, n)\| \leq (8n)^{L+1}$$

$$\|U^\Phi[D, n]\| \leq (8n)^{2^D}$$

$$\|U^\Phi[D, n]\| \subseteq U^\Phi(2^D - 1, n)$$

## Доказательство.

Чтобы задать количество попарно не изоморфных формул, достаточно выбрать двоичное корневое дерево ( $D'$ ), у которого вершин ( $q$ ) не больше, чем  $L$ . Далее происходит перенумерация вершин таким образом, чтобы номер вершины, который находится ниже был больше номера вершины, который выше по цепи. Если эти вершины не находятся на одной цепи, то номер той вершины справа больше, чем номер вершины слева.

Необходимо знать какие вершины соединяются с внутренними вершинами, а какие с входными вершинами. Поэтому к каждой внутренней вершине сопоставляется двоичный набор.

$$d_i = d^+(v_i)$$

$$\alpha^{(i)} = (\alpha_1^{(i)}, \dots, \alpha_{d_i}^{(i)})$$

Дуга с номером  $j$ , входящая в вершину  $v_i$  начинается со входа:

$$\alpha^{(i)}_j = 1$$

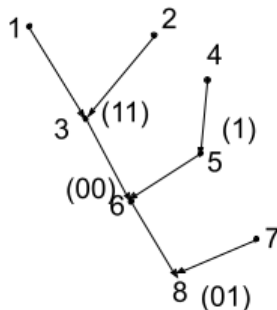


Рис. 7.5. Двоичное корневое дерево ( $D'$ )

$$\gamma = (\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(q)}, \dots, \beta^{(q+1)}, \dots, \beta^{(G)})$$

По произвольному набору  $\gamma$  и по числу  $q$  восстанавливается двоичное дерево. Помимо структуры дерева в форму еще есть функциональные элементы. Поэтому рассматриваются функциональные элементы, которые можно приписать каждой внутренней вершине. Рассматривается вектор из функциональных

элементов:

$$\delta = (\delta_1, \dots, \delta_q, \dots, \delta_L)$$
$$\delta_i = \begin{cases} \{\&, \vee\} & d_i = 2; \\ | & d_i = 1 \end{cases}$$

Таким образом, формула задается набором  $\gamma$ , который задает ее структуру, и набором  $\delta$ , который определяет какие функциональные элементы приписываются вершинам. Но если структурно один вход вершины, то ему приписывается функциональный элемент отрицания. Количество комбинаций выбора не превосходит  $10^L$ .

После того как было выбрано дерево, необходимо приписать переменные к входным вершинам. Количество вариантов приписывания листьям дерева различных символов входных переменных не превосходит  $n^{R(F)}$ .

$$R(F) \leq L + 1$$

Таким образом, доказывается следующее:

$$|U^\Phi(L, n)| \leq (10n)^{L+1}$$

■

### Доказательство.

Чтобы посчитать количество попарно неэквивалентных формул, необходимо сделать следующее:

$$(01) = (10)$$

В итоге получается оценка  $8^n$ .

■

Схемы — квазиизоморфны, если они изоморфны с точностью до порядка дуг, входящие в коммутативные операции. В следствии получается, что число попарно не квазиизоморфных формул в стандартном базисе с поднятыми отрицаниями ранга, не превосходящего  $R$ , зависящих от переменных  $x_1, \dots, x_n$ , не превосходит величины  $(12n)^R$ .

$$||U^{\Phi, |}\{R, n\}|| = ||U_{\{\&, \vee\}}^\Phi\{R, 2n\}|| \leq 6^R \cdot (2n)^R$$

## Оценка количества попарно неэквивалентных СФЭ

Утверждение 7.3.

$$||U^C(L, n)|| \leq (8(n + L))^{L+1} \quad \forall n, L \in \mathbb{N}$$

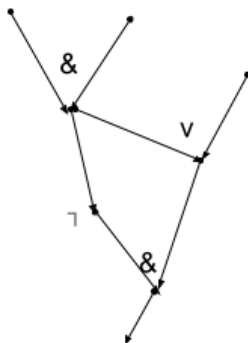


Рис. 7.6. Оценка количества попарно неэквивалентных СФЭ

## Лекция 8. Эквивалентные преобразования СФЭ

### Пример эквивалентных СФЭ

Пусть есть базис  $B$ , над которым строятся формулы и схемы из функциональных элементов. Каждый функциональный символ этого базиса связан с функциональным элементом  $\varepsilon_i$ , который реализует функцию  $\varphi_i$ , которая приписана функциональному символу.

$$B = \{\varphi_1, \dots, \varphi_b\}, \quad \varepsilon \leftrightarrow \varphi_i(x_1, \dots, x_{k_i})$$

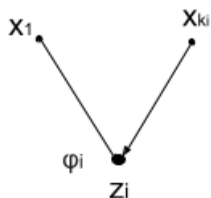


Рис. 8.1. Функциональный элемент  $\varepsilon_i$

$z_i$  — выходная переменная. Схему из функциональных элементов можно представить в виде графа. Схема реализует систему функций. Две схемы эквивалентны тогда и только тогда, две системы функций равны между собой.

$$\Sigma'(x_1, \dots, x_n, z_1, \dots, z_m) \sim \Sigma''(x_1, \dots, x_n, z_1, \dots, z_m)$$

$$\Sigma' \rightarrow F_{\Sigma'}(f_1, \dots, f_m)$$

$$\Sigma'' \rightarrow F_{\Sigma''}(x_1, \dots, x_m, z_1, \dots, z_m)$$

$$F_{\Sigma'} = F_{\Sigma''}$$

Они совпадают как наборы из функций. Рассматривается схема  $\Sigma'$  в стандартном базисе.

На выходе  $z_1$  реализуется функция, совершенная ДНФ сложение по модулю 2:

$$z_1 = \overline{x_1 \cdot x_2}(x_1 \vee x_2) = (\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_2) = \bar{x}_1 x_2 \vee x_1 \bar{x}_2$$



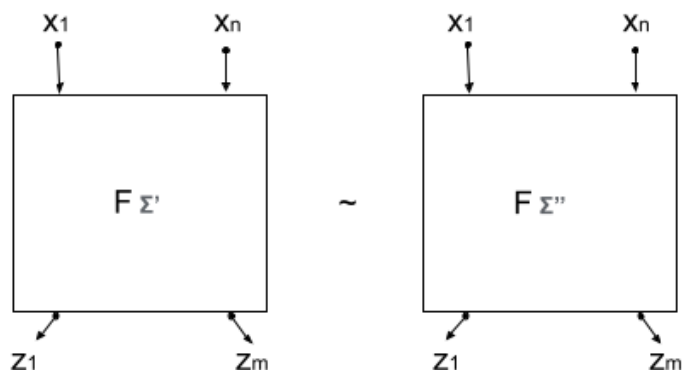


Рис. 8.2. Схема  $\Sigma'$  и  $\Sigma''$

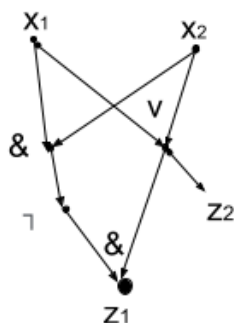


Рис. 8.3. Схема  $\Sigma'$  в стандартном базисе

$$F' = (x_1 \oplus x_2, x_1 \vee x_2)$$

Схема  $\Sigma'$  реализует систему из двух функций от двух переменных. Одна из этих функций — это сложение по модулю 2 этих переменных, вторая функция — это дизъюнкция этих двух переменных.

Рассматривается следующий пример схемы  $\Sigma''$ :

$$\overline{(\bar{x}_1 \cdot \bar{x}_2 \vee x_1 x_2)} = x_1 \oplus x_2$$

Эти схемы эквивалентны, поскольку каждая из них реализует одинаковую систему функций, которая представляет собой систему из равных функций.

$$F'' = (x_1 \oplus x_2, x_1 \vee x_2)$$

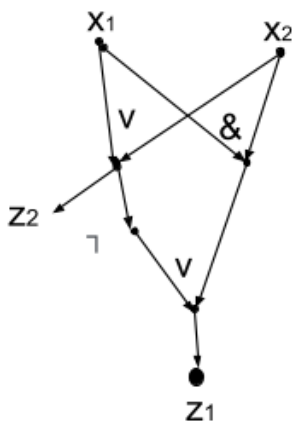


Рис. 8.4. Схема  $\Sigma''$

## Задача эквивалентных преобразований (ЭП) для СФЭ. Модификация тождества

Эквивалентное преобразование происходит, не изменяя функционирование схемы, а улучшая ее задержку или сложность. Тождество записывается следующим образом:

$$t : \Sigma'(x_1, \dots, x_n, z_1, \dots, z_m) \sim \Sigma''(x_1, \dots, x_n, z_1, \dots, z_m)$$

Чтобы говорить о задаче эквивалентных преобразований, необходимо определить тождество, модификации тождества и ввести понятие подсхемы. Модификация тождества  $\hat{t}$  — это тождество, получающееся из тождества  $t$  переименованием (с возможным отождествлением) входных БП в схемах  $\Sigma'$  и  $\Sigma''$ , и переименованием (с возможным дублированием или снятием) выходных БП. Дублирование — это добавление второй пометки  $z_k$  на выход с пометкой  $z_j$ .



Рис. 8.5. Дублирование выходных БП

Снятие — это удаление пометки выхода на вершине.



Рис. 8.6. Снятие выходных БП

Если две схемы эквивалентны, то при снятии одинаковых выходных переменных, снова получатся эквивалентные схемы.

Базовый элемент любой модели эквивалентных преобразований — это понятие подсхемы.

### Подсхема СФЭ. Многократное ЭП. Полнота системы тождеств

Предполагается, что если есть тождество, то подсхему необходимо искать в исходной схеме и вместо этой подсхемы подставить подсхему второй схемы.

**Определение 8.1.** СФЭ  $\Sigma'$  является подсхемой СФЭ  $\Sigma$  тогда и только тогда, когда:

•

$$V(\Sigma') \subseteq V(\Sigma)$$

$$E(\Sigma') \subseteq E(\Sigma)$$

- Любая входная вершина схемы  $\Sigma$  из множества вершин схемы  $\Sigma'$  — это вход  $\Sigma'$ . Любая вершина подсхемы  $\Sigma'$  без входящих дуг — это тоже вход  $\Sigma'$ .

Каждая выходная вершина  $\Sigma$  из множества  $V(\Sigma')$  — это выход  $\Sigma'$ . Любая вершина подсхемы  $\Sigma'$ , являющаяся начальной вершиной дуги из множества  $E(\Sigma) \setminus E(\Sigma')$  — это выход  $\Sigma'$ .

Таким образом, буде выполняться принцип эквивалентной замены для схемы.  $\tilde{\Sigma}$  получается из  $\Sigma$  заменой подсхемы  $\Sigma'$  на  $\Sigma''$ . Следовательно, получается, что схема  $\tilde{\Sigma}$  эквивалентна схеме  $\Sigma$ .

$$\tilde{\Sigma} : \tilde{\Sigma} \sim \Sigma$$

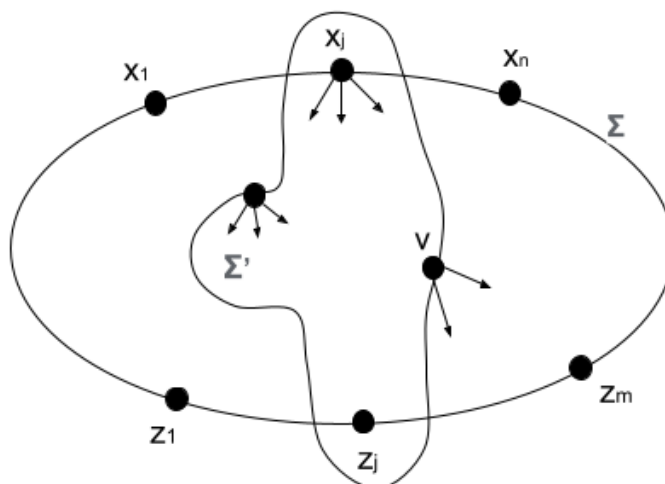


Рис. 8.7.  $\Sigma'$  является подсхемой  $\Sigma$

$$\begin{aligned} \Sigma &\xrightarrow[t]{} \tilde{\Sigma} \\ t &\mapsto \tilde{t} \end{aligned}$$

Если использовать несколько таких замен, то появляется многократное эквивалентное преобразование с помощью системы тождеств  $\tau$ .

$$\Sigma \xrightarrow[\tau]{} \hat{\Sigma}$$

Система тождеств является полной тогда и только тогда, когда две любые эквивалентные схемы из функциональных элементов можно преобразовать друг в друга с помощью системы тождеств  $\tau$ .

## Сопоставление ЭП формул и ЭП СФЭ. Примеры

**Утверждение 8.1.** Если  $\tau$  — конечная полная система тождеств для ЭП формул из  $U_B^\Phi$ , то  $\{\underline{\tau}, \tau^C, \tau^B\}$  — конечная полная система тождеств для ЭП СФЭ из  $U_B^C$ .

В следствии, система тождеств  $\{\underline{\tau}_{\text{ОСН}}, \tau^B, \tau^C\}$  — КПСТ для ЭП СФЭ из  $U^C$ .

Формул рассматриваются как частый случай схем. Поэтому любое тождество для формул является одновременно тождеством для тех схем, которые этим формулам сопоставлены. В модели СФЭ деревья не являются схемами. А

вот квазидеревья ( $\hat{D}_F$ ), графы, которые получаются из деревьев формул отождествлением одинаковых формул, являются схемами СФЭ. С их помощью можно задавать необходимые формулы.

$$\forall F \in U_B^\Phi \leftrightarrow \underline{F} = \hat{D}_F \in U_B^C$$

Пусть есть множество формул:

$$U \subseteq U_B^\Phi \leftrightarrow \underline{U} \subseteq U_B^C$$

$$t : F' = F'' \leftrightarrow \underline{t} : \underline{F}' \sim \underline{F}''$$

Если две формулы эквивалентны, то соответствующие квазидеревья тоже эквивалентны. Пусть есть система тождеств  $\tau$  для формул в базисе  $U_B^\Phi$ , то ей соответствует система тождеств  $\underline{\tau}$  для схемы СФЭ  $U_B^C$ .

**Пример 8.1.** Тождество подстановки константы 1 в конъюнкцию:

$$t_{1,\&}^{PK} : x_1(x_2 \vee \bar{x}_2) = x_1$$

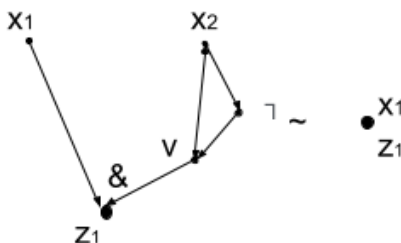


Рис. 8.8. Сопоставленная схема  $t_{1,\&}^{PK}$

**Пример 8.2.** Тождество де Моргана для конъюнкции:

$$t_{\&}^M : \overline{x_1 \cdot x_2} = \bar{x}_1 \vee \bar{x}_2$$

Для любого формульного тождества можно найти схемный аналог. Но есть схемы, которые не сводятся к формулам.

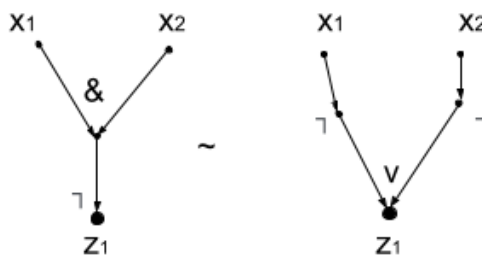


Рис. 8.9. Сопоставленная схема  $t_{\&}^M$

**Пример 8.3.** Тождество ветвления конъюнкции:  $t_{\&}^B$ . Выход имеет две пометки и поэтому выход является кратным выходом. Это ветвление можно поднять к входам самого элемента. Это тождество не имеет никакого формульного аналога.

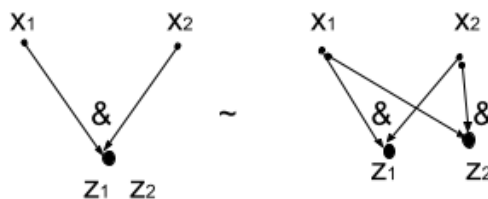


Рис. 8.10. Схема  $t_{\&}^B$

**Пример 8.4.** Тождество снятия висячих конъюнкций:  $t_{\&}^C$ . Выход этого элемента не является выходом этой схемы. Эту схему можно сбросить и оставить только висячие входы.

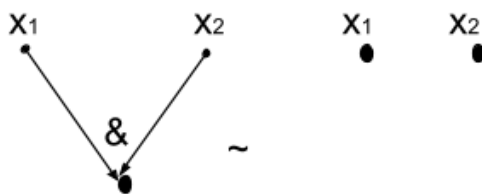


Рис. 8.11. Схема  $t_{\&}^C$



Рис. 8.12. Схема  $t_{\text{вх}}^C$

**Пример 8.5.** Тождество снятия висячего входа:  $t_{\text{ex}}^C$ .

Если такие тождества добавить к полной системе тождеств для формул, то получится полная система для схем.

Любое эквивалентное преобразование в классе формул можно промоделировать в новой модели эквивалентных преобразований СФЭ. Записывается эквивалентное преобразование для формул:

$$x_1(x_2x_3 \vee \bar{x}_2 \vee \bar{x}_3) \xrightarrow{t_{\&}^M} x_1(x_2x_3 \vee (\overline{x_2x_3})) \xrightarrow{t_{1,\&}^{\text{ПК}}}$$

Это преобразование необходимо представить в виде цепочки преобразований для СФЭ. Для этого понадобятся дополнительные тождества. Выделяется подсхема, входами которой являются  $x_2$  и  $x_3$ . Эта подсхема соответствует левой части схемы тождества де Моргана для конъюнкции. Эта подсхема заменяется правой частью тождества де Моргана.

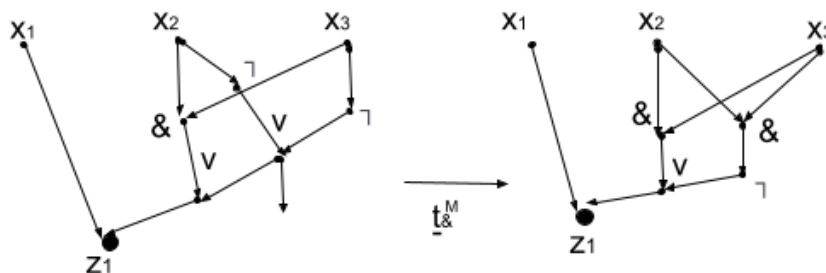


Рис. 8.13. Шаг 1: эквивалентное преобразование для формул

Дальше необходимо применить тождество постановки константы 1 в конъюнкцию. Чтобы появилась эквивалентная подсхема, нужно применить тождество ветвления элемента конъюнкции. Выделяется подсхема с входами  $x_2$  и  $x_3$  и с промежуточными выходами  $z_1$  и  $z_2$ .

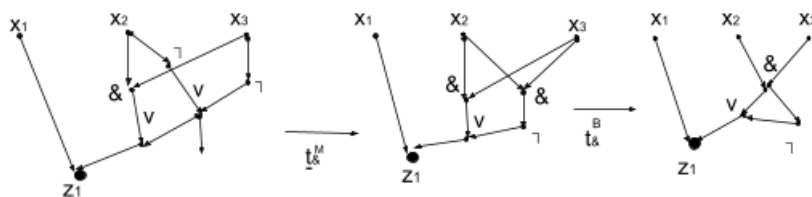


Рис. 8.14. Шаг 2: эквивалентное преобразование для формул

Таким образом, после этого шага можно применить тождество постановки константы 1 в конъюнкцию. Вместе с  $x_1$  также появляется висячая конъюнкция.

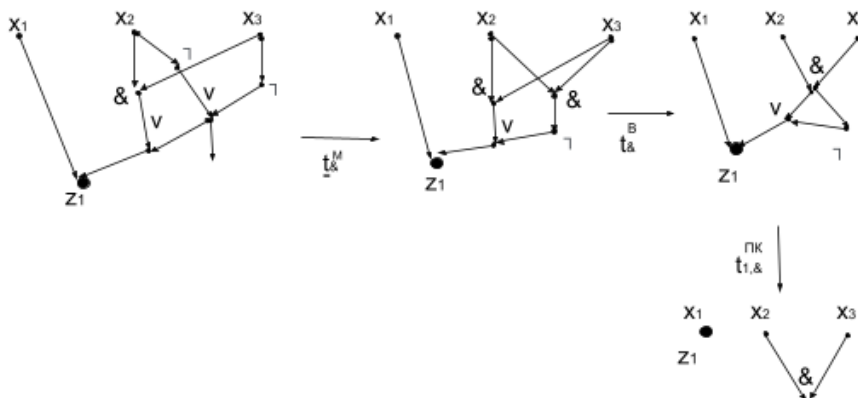


Рис. 8.15. Шаг 3: эквивалентное преобразование для формул

Висячие конъюнкции можно удалить с помощью тождества снятия висячей конъюнкции.

Эти два висячих входа можно удалить с помощью применения тождества снятия висячих входов дважды.

## Утверждение о получении полной системы тождеств для ЭП СФЭ

Определяется канонический вид схемы. Достаточно доказать, что любые две эквивалентные схемы можно преобразовать друг в друга.



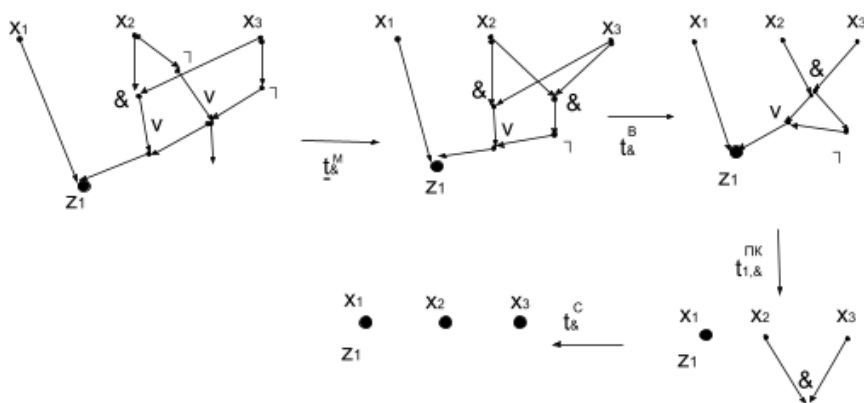


Рис. 8.16. Шаг 4: эквивалентное преобразование для формул

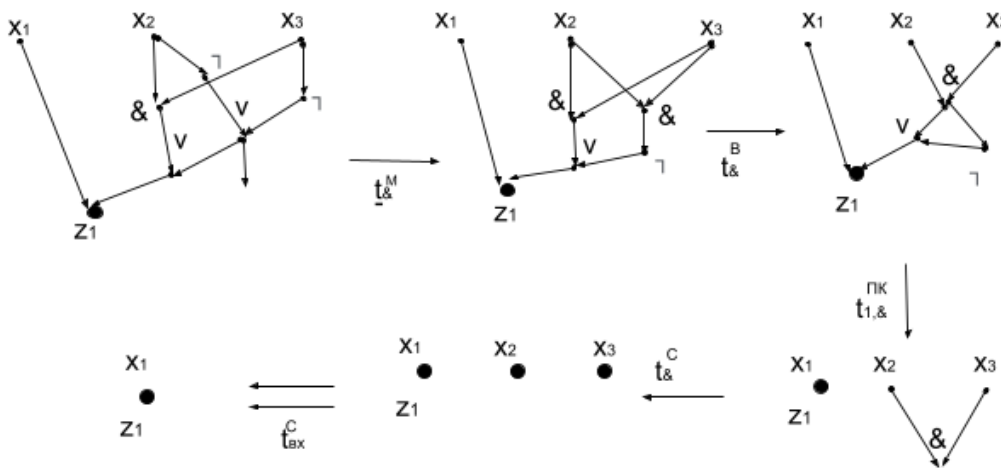


Рис. 8.17. Шаг 5: эквивалентное преобразование для формул

**Доказательство.**

Пусть есть СФЭ над базисом Б:

$$\Sigma(x_1, \dots, x_n, z_1, \dots, z_m)$$

Этой СФЭ сопоставляется канонический вид  $\hat{\Sigma}$  схемы  $\Sigma$ .

$$\hat{\Sigma} = (a_1, \dots, a_m)$$

$a_j$  — это формула, реализуемая на выходе  $z_j$ . Существует эквивалентное преобразование схемы  $\Sigma$  в схему  $\hat{\Sigma}$  с помощью тождеств ветвления и снятия в базисе  $B$ . Преобразование одного канонического вида в другой канонический вид делается с помощью однократного эквивалентного преобразования  $F \mapsto_t \hat{F}$ , которое можно промоделировать с помощью:

$$F \xrightarrow[t]{L, t_B^B, t_B^C}$$

■

**Пример 8.6.** Пусть есть формульное тождество:

$$t : F' = F''$$

Применяется подстановка этого тождества:

$$\hat{t} : F'(F_1, \dots, F_k) = F''(F_1, \dots, F_k)$$

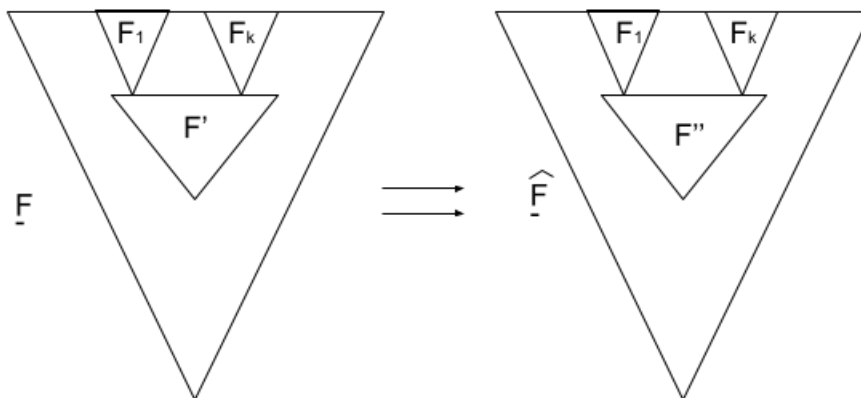


Рис. 8.18. Применение подстановки тождества

Если  $F'$  неповторная формула, то тогда достаточно одного эквивалентного преобразования. Если  $F'$  формула с повторениями, то тогда необходимо использовать тождество ветвления и снятия.

### Доказательство.

$\Sigma' \sim \Sigma''$  — эквивалентные СФЭ в  $U_B^C$ . Схема  $\Sigma'$  приводится к схеме  $\hat{\Sigma}'$ , а  $\Sigma''$  в  $\hat{\Sigma}''$ . Потом от одного канонического вида происходит переход в другой канонический вид. Для этого используются  $\tau$ ,  $\tau_B^B$ ,  $\tau_B^C$ . ■

## Лекция 9. Контактная схема

### Типы проводящих контактов, их соответствие с МОП-транзисторами

Преобразование информации осуществляется в вершинах, а ребра схемы служат для передачи информации. Это модель функционального типа. Вторая серия моделей — это модели передающего типа такие, как контактные схемы, в которых информация передается по ребрам схемы и при этом она обрабатывается на ребрах. Рассматриваются контактные схемы, построенные из нескольких типов контактов. В первом типе контактной схемы контакт управляется  $x_i$ . Контакт пропускает информацию тогда и только тогда, когда:

$$x_i = 1$$

Это замыкающий контакт.

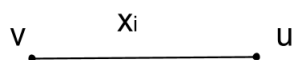


Рис. 9.1. Замыкающий контакт

Второй тип контакта — это размыкающий контакт тогда и только тогда, когда:

$$x_i = 0$$

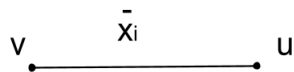


Рис. 9.2. Размыкающий контакт

Оба эти контакта являются неориентированными ребрами. Это означает, что передача контакта происходит как в одну сторону, так и в другую. Есть модификация этих контактов (ориентированный контакт), которые тоже бывают

двух типов. Такой контакт проводит информацию только в одном направлении и тогда и только тогда, когда:

$$x_i^\sigma = \sigma$$

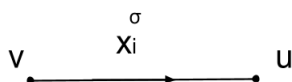


Рис. 9.3. Ориентированный контакт

Также есть чистый, но ориентированный проводник, который не зависит от управляющих переменных и проводит только от вершины  $v$  к  $u$ .

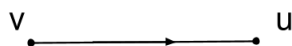


Рис. 9.4. Чистый ориентированный контакт

Замыкающий контакт соответствует  $N$  – МОП-транзистору. Размыкающий контакт соответствует  $P$  – МОП-транзистору. Ориентированные контакты соответствуют  $N$  или  $P$  – МОП-транзисторам с диодом Шоттки.

## Контактная схема (КС). Пример. Матрица проводимости, функция проводимости. Примеры

**Определение 9.1.**  $(p, q)$  – контактная схема (КС) от БП  $(x_1, \dots, x_n)$  с входами  $a_1, \dots, a_p$  и с выходами  $b_1, \dots, b_q$  т.е.:

$$\Sigma(x_1, \dots, x_n; a_1, \dots, a_p; b_1, \dots, b_q)$$

*Управляющие переменные управляют проводимостью контактов.*

*Это есть  $(p, q)$  – сеть из контактов.*

**Пример 9.1.**

$$\Sigma(x_1, x_2, x_3; a_1; b_2)$$

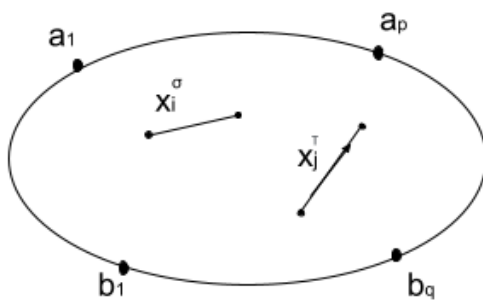


Рис. 9.5. Схема  $(p, q)$

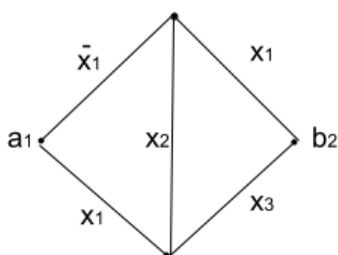


Рис. 9.6. Контактная схема  $\Sigma$

Контактная схема реализует матрицу проводимости  $F$ :

$$F = ||f_{ij}|| \quad 1 \leq i \leq p \quad 1 \leq j \leq q$$

$f_{ij}$  — функция проводимости от  $a_i$  к  $b_j$ .

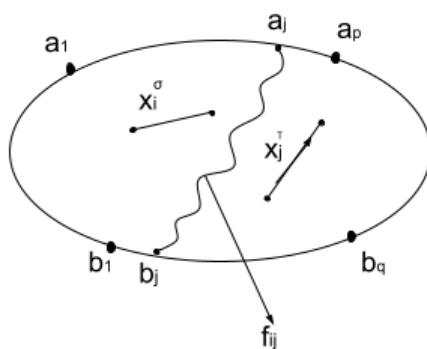


Рис. 9.7. Функция проводимости

Определяется функция проводимости для  $\forall(v, u) \in V(\Sigma)$  от вершины  $v$  к  $u$ .  $g_{v,u}(x_1, \dots, x_n)$  — это функция проводимости от  $v$  к  $u$  такая, что:

$$g_{v,u}(\alpha_1, \dots, \alpha_n) = 1$$

Тогда и только тогда, когда в схеме  $\Sigma$  существует  $(v - u)$  цепь, состоящая из контактов вида  $x_1^{\alpha_1}$  и  $x_n^{\alpha_n}$ , проводящих на наборе  $\alpha$ . Чтобы найти функцию проводимости в контактной схеме, необходимо построить таблицу значений переменных.

$x_1 x_2 x_3$	$g_{v,u}$
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

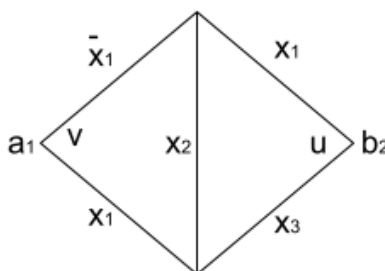


Рис. 9.8. Таблица значений переменных

Функция проводимости определяется независимо от того сколько входов и выходов в схеме.

$$\Sigma(x_1, x_2, x_3; a_1; b_2) = H(x_1, x_2, x_3)$$

Функция голосования равна 1, когда не менее двух переменных из трех приняли значение 1.

Схема Кардо имеет два входа и два выхода и реализует матрицу из четырех функций проводимости. Структура контактной схемы записывается следующим образом:

$$\Sigma = \Sigma(x_1, x_2, x_3; a_1, v; b_1, b_2)$$

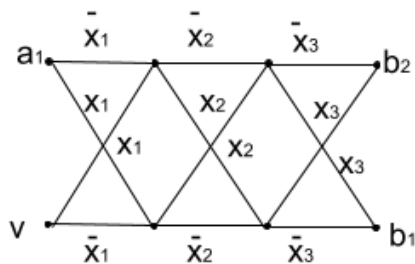


Рис. 9.9. (2,2) – контактная схема

Реализованная матрица функций записывается следующим образом:

$$F_{\Sigma} = \begin{bmatrix} l_3 & \bar{l}_3 \\ \bar{l}_3 & l_3 \end{bmatrix}$$

### Цепь контактов. Проводящая цепь. Сопоставление цепи функции проводимости. Примеры

Контактная схема — это несколько свернутых ДНФ. Каждая цепочка в контактной схеме — это элементарная конъюнкция, а параллельно соединенные цепочки — дизъюнкция. Таким образом, используя ДНФ, можно найти функцию проводимости контактной схемы. Пусть есть схема:

$$\Sigma = \Sigma(x_1, \dots, x_n; a_1, \dots, a_p; b_1, \dots, b_q)$$

Предполагается, что вершины  $v$  и  $u$  соединены цепью из контактов.

$$K = x_{i_1}^{\sigma_1} \cdot \dots \cdot x_{i_r}^{\sigma_r}$$

$C(u - v)$  — это цепь в схеме  $\Sigma$ , состоящая из контактов  $K$ . Множеству контактов сопоставляется функция проводимости:

$$K(C) = K = x_{i_1}^{\sigma_1} \cdot \dots \cdot x_{i_r}^{\sigma_r}$$

$C$  — проводящая цепь тогда и только тогда, когда:

$$K(C) \neq 0$$



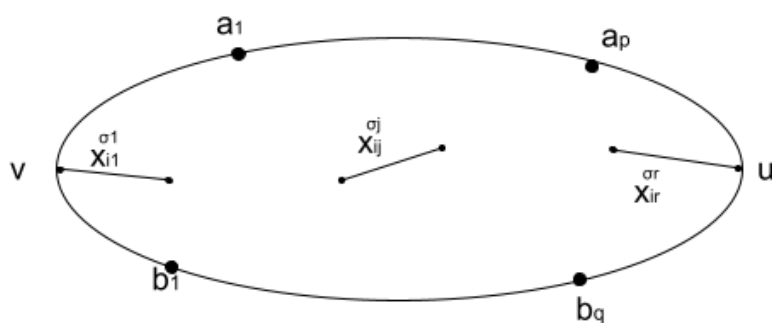


Рис. 9.10. Контактная схема как ДНФ

Пусть  $C_1, \dots, C_t$  — все проводящие цепи, соединяющие вершину  $v$  с вершиной  $u$ .

$$g_{v,u} = K(C_1) \vee \dots \vee K(C_t)$$

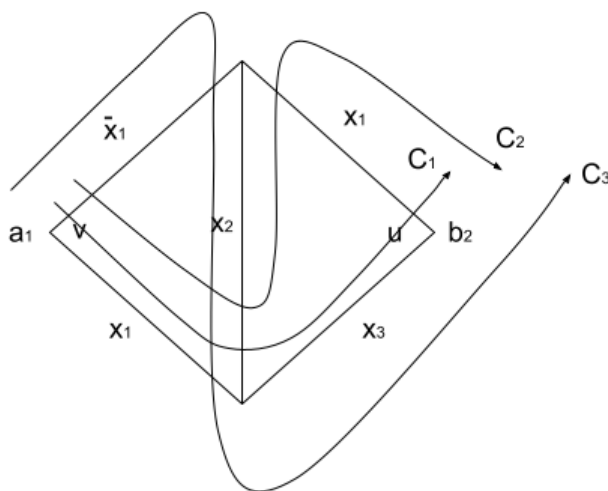


Рис. 9.11. Цепи в контактной схеме

$$K(C_1) = x_1 x_3$$

$$K(C_2) = x_1 x_2 x_1 = x_1 x_2$$

$$K(C_3) = \bar{x}_1 x_2 x_3$$

Получается ДНФ для функции голосования:

$$K(C_1) \vee K(C_2) \vee K(C_3) = H$$

Следовательно, это доказывает, что функция реализует функцию голосования.

## Сечения КС, их виды. ФАЛ отделимости

$S(v|u)$  — это сечение в схеме  $\Sigma$ . Это означает, что если оборвать все контакты, то исчезнут все цепи, идущие из  $v$  в  $u$ . Рассматривается функция отделимости множества контактов:

$$J(S) = x_{i_1}^{\sigma_1} \vee \dots \vee x_{i_r}^{\sigma_r}$$

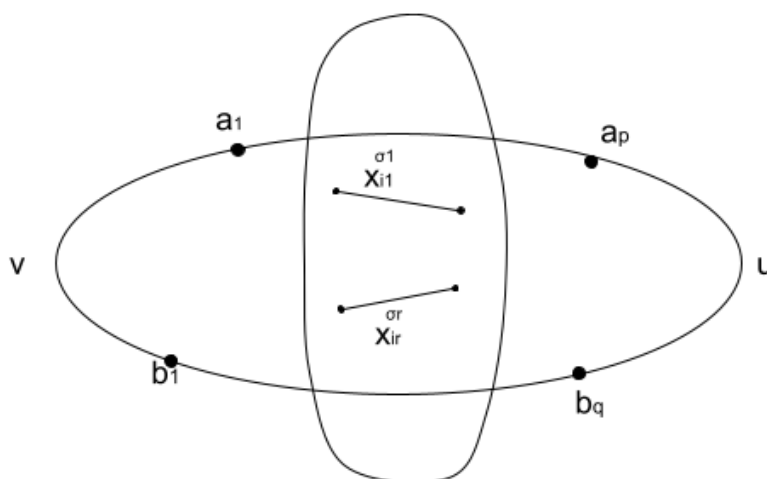


Рис. 9.12. Сечение КС

$S$  — не единичное сечение тогда и только тогда, когда:

$$J(S) \neq 1$$

Пусть  $S_1, \dots, S_t$  — все тупиковые и не единичные сечения, отделяющие  $v$  и  $u$ . Функцию  $g$  можно написать в виде КНФ:

$$g_{u,v} = J(S_1) \& \dots \& J(S_t)$$

## Пи-схема. Последовательное и параллельное соединения

$\pi$  - схема — (1,1) - КС, которая определяется индуктивно следующим образом:

- 1) Простейшая  $\pi$  - схема состоит из 1 контакта с одним входом и выходом. Простейшая  $\pi$  - схема реализует функцию равную той самой букве, которая приписана этому самому единственному контакту.
- 2) Если есть две контактные схемы  $\Sigma_1$  и  $\Sigma_2$ , реализующие ФАЛ  $f_1$  и  $f_2$ , то можно соединить эти 2 схемы последовательно или параллельно.

$$\Sigma' = \Sigma_1 \cdot \Sigma_2$$

$$\Sigma'' = \Sigma_1 \parallel \Sigma_2$$

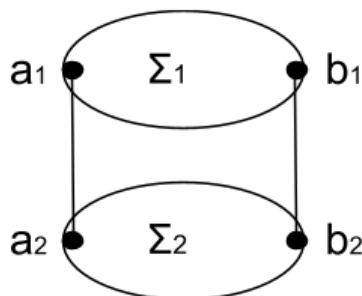


Рис. 9.13. Последовательное соединение

Рис. 9.14. Параллельное соединение

Функция проводимости при последовательное соединении — конъюнкция, а при параллельном — дизъюнкция.

$$\Sigma' \rightarrow f_1 \cdot f_2$$

$$\Sigma'' \rightarrow f_1 \vee f_2$$

Можно построить  $\pi$  - схему, реализующая функцию голосования.

$$H(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3$$

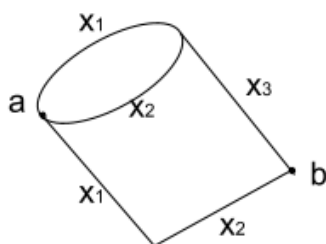


Рис. 9.15.  $\pi$  - схема, реализующая  $H$

### Сопоставление $\pi$ -схеме формулы с поднятыми отрицаниями

**Утверждение 9.1.** *Каждой  $\pi$  - схеме  $\Sigma$  можно сопоставить эквивалентную ей формулы  $F$  из  $U^\Phi$  с поднятыми отрицаниями, при этом ранг этой формулы будет равен сложности  $\pi$  - схемы.*

$$R(F) = L(\Sigma)$$

$$F = (x_1 \vee x_2)x_3 \vee x_1x_2$$

**Доказательство.**

Индукция по строению  $\pi$  - схемы  $\Sigma$ . Если  $F_1$  моделирует  $\Sigma_1$ , а  $F_2$  моделирует  $\Sigma_2$ , то тогда чтобы моделировать  $\Sigma'$  и  $\Sigma''$ :

$$F' = F_1 \cdot F_2$$

$$F'' = F_1 \vee F_2$$

$$R(F_1) = L(\Sigma_1) \rightarrow R(F') = L(\Sigma')$$

$$R(F_2) = L(\Sigma_2) \rightarrow R(F'') = L(\Sigma'')$$

■

## Контактное дерево

Функция проводимости от входа  $a_j$  до выхода представляет собой элементарную конъюнкцию. Схема реализует систему функций проводимости от входа к выходу — вектор, состоящий из различных ЭК.

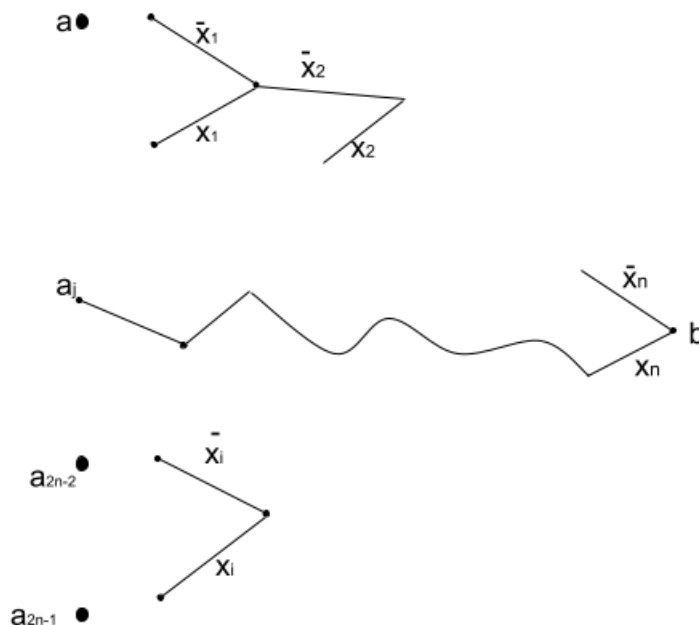


Рис. 9.16. Контактное дерево  $(2^n, 1)$

Схема  $(1,1)$  - КС реализует дизъюнкцию.

$$\bigvee_{(\sigma_1, \dots, \sigma_n) \in B^n} x_1^{\sigma_1} \dots x_n^{\sigma_n} \cdot y_V(\sigma_1, \dots, \sigma_n)$$

Реализуется мультиплексорная функция от адресных переменных и информационных переменных.

## Оценка числа попарно неэквивалентных пи-схем

**Утверждение 9.2.** Число попарно не эквивалентных  $\pi$  - схем сложности не более, чем  $L$  от  $n$  переменных не больше, чем  $(12n)^L$ .

$$\|U^\pi(L, n)\| \leq (12n)^L$$

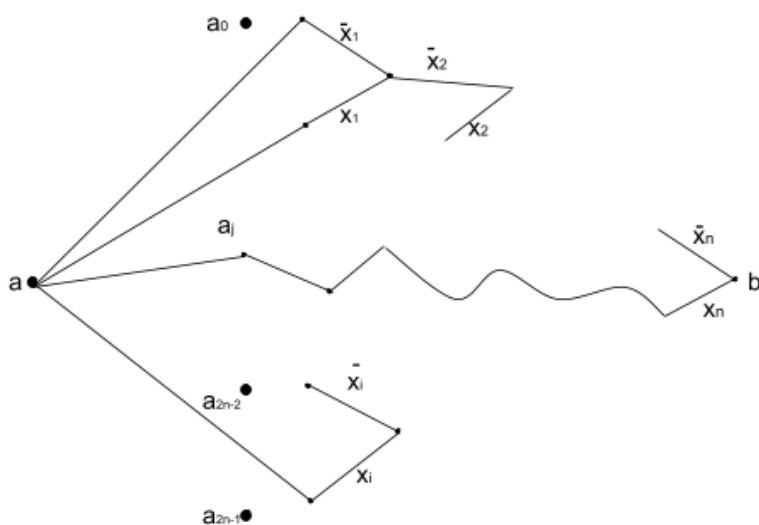


Рис. 9.17. Контактное дерево (1, 1)

Сложность контактной схемы — это сложность контактов.

**Доказательство.**

$$\Sigma \in U^\pi(L, n) \rightarrow F \in U^\Phi$$

Сопоставляется формула с поднятыми отрицаниями:

$$R(F) \leq L$$

■

## Неприводимая КС. Утверждение об оценке числа попарно неэквивалентных КС

**Утверждение 9.3.** При любых натуральных  $L$  и  $n$  выполняется неравенство:

$$||U^K(L, n)|| \leq (8nL)^L$$

$U^K$  — это класс контактных схем из неориентированных контактов.

$$U^K(L, n) = \{\Sigma \in U^K \mid \Sigma = \Sigma(x_1, \dots, x_n; a; b) : L(\Sigma) \leq L\}$$

$||U^K(L, n)||$  — число попарно не эквивалентных контактных схем из неориентированных контактов или число различных функций от переменных  $x_1, \dots, x_n$ , реализуемых  $(1,1)$  - КС сложности не более, чем  $L$ .

$$||U^K(n, 1)|| = 2n + 1$$

$(p, q)$  - КС  $\Sigma$  — неприводимая контактная схема тогда и только тогда, когда любая внутренняя вершина и любой контакт принадлежат некоторой простой проводящей цепи, соединяющая полюса схемы.

$$p = q = 1 \quad f_\Sigma \neq 0$$

Следовательно, схема  $\Sigma$  является связным графом.

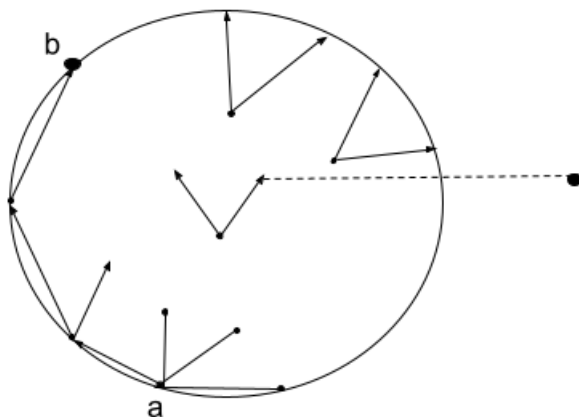


Рис. 9.18. Остовное дерево  $D$

**Доказательство.**

Пусть  $\Sigma \in U^K$  — неприводимая  $(1,1)$  КС сложности не более, чем  $L$ . Это связный граф.

$$\Sigma = \Sigma(x_1, \dots, x_n; a; b)$$

В любом связном графе можно выделить остовное дерево. Пусть  $D$  — остовное корневое дерево  $\Sigma$  с корнем  $a$ , содержащим все ребра инцидентные  $a$  и первым листом которого является  $a_2$ .

$$V(D) = V(\Sigma)$$

$\hat{D}$  — остовное наддерево  $\Sigma$ . Каждое ребро, которое было в исходном графе, но не вошло в остовное дерево, приклеиваться к одной из вершин и вершина этого ребра объявляется дополнительным листом дерева.  $\Sigma$  получается из  $\hat{D}$  присоединением некоторых листьев к внутренним вершинам того же дерева  $\hat{D}$ .

- 1) Выбор  $\hat{D}$  не больше, чем  $4^L$ .
- 2) Присоединение листьев  $\hat{D}$  к внутренним вершинам не больше, чем  $L^L$ .
- 3) Сопоставление  $x_i^\Sigma$  к каждому ребру не больше, чем  $(2n)^L$ .

Таким образом, получается оценка  $(8nL)^L$ . ■



## Лекция 10. Нижние оценки сложности ФАЛ

### Задача синтеза. Простейшие верхние оценки

Метод синтеза схем основан на ДНФ. Были получены верхние оценки сложности функций. Пусть  $U$  — класс схем и  $\psi$  - функционал сложности схем из  $U$ . Два основных функционала сложности: число элементов и глубина. Для любой  $F = (f_1, \dots, f_m)$  построить  $\psi$  - оптимальную схему  $\hat{\Sigma}$ , которая реализует эту систему функций и имеет минимальную  $\psi$  - сложность среди всех схем из данного класса, реализующих эту систему функций. Минимальное значение, которое совпадает со значением  $\psi$  - сложности схемы  $\Sigma$ , называется  $\psi$  - сложностью системы функций  $F$  в классе схем  $U$ .

$$\psi(\hat{\Sigma}) = \min_{\Sigma \in U} \psi(\Sigma),$$

Необходимо сравнить простейшие верхние оценки с нижними оценками. Задача синтеза решается следующим образом:

- 1) Строится схема  $\Sigma$  из класса  $U$ , которая реализует систему функций  $F$  и тем самым получается верхняя оценка  $\psi$  - сложности:

$$\psi(F) \leq \psi(\Sigma)$$

- 2) Доказывается следующее:

$$\psi(F) \geq \psi(\Sigma)$$

Если удастся доказать это, то можно получить нижние оценки, которые совпадают с верхними оценками.

### Нижние оценки сложности системы ФАЛ

Для системы  $F = (f_1, \dots, f_m)$ , состоящей из попарно различных ФАЛ отличных от констант, справедливо неравенство:

$$L^K(F) \geq m$$

Для системы  $F = (f_1, \dots, f_m)$ , состоящей из попарно различных ФАЛ отличных от БП, справедливо неравенство:

$$L_B^C(F) \geq m$$

С помощью этой оценки было доказано, что справедливо следствие:

$$L^K(\vec{Q}_n) \geq 2^n$$

$$L^C(\vec{Q}_n) \geq 2^n$$

$$L_B^C(\vec{P}_2(n)) \geq 2^{2n} - n$$

Это доказывает, что нижние и верхние оценки совпадают. Сложность реализации системы из всех функций от  $n$  переменных равна  $2^{2n} - n$ .

## Нижние оценки сложности отдельной ФАЛ

**Утверждение 10.1.** Если ФАЛ  $f(x_1, \dots, x_n)$  существенно зависит от все своих БП, то:

$$L_{\&, \vee}^C(f) \geq n - 1$$

$$L^K(f) \geq n$$

Если при этом ФАЛ  $f$  не является монотонной ФАЛ (каждая БП  $x_i$ ,  $i \in [1, k]$  не является ни монотонной, ни инмонотонной БП ФАЛ  $f$ ), то:

$$L^C(f) \geq n$$

$$L_1(\Sigma) \geq 1, \quad \forall \Sigma \in U^C$$

$$L^K(f) \geq n - k$$

$$L(\Sigma) \geq 2k + (n - k) \geq n + k$$

**Доказательство.**

Пусть  $\Sigma$  — приведенная СФЭ в  $B_0$ , реализующая функцию  $f$  с  $n$  входами.

$$R(\Sigma) \geq n$$

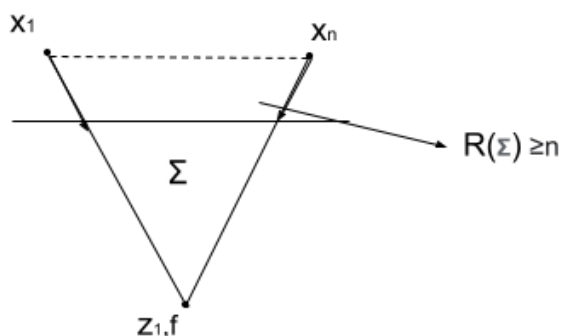


Рис. 10.1. приведенная СФЭ

$$L_{\&,v}(\Sigma) \geq (n - 1)$$

■

Пусть  $\Sigma$  — (1,1) КС, реализующая функцию  $f$ . Следовательно, в схеме  $\Sigma$  встречаются контакты любой переменной  $x_i$ .

$$L(\Sigma) \geq n$$

Оценки для основных индивидуальных функций:

$$L^C(l_n) \geq n$$

$$L^K(l_n) \geq 2n$$

$$L^C(\mu_n) \geq 2n + n$$

$$L^K(l_n) \geq 2^n + 2n$$

Мультиплексорная функция — это функция, которая по набору значений своих адресных переменных выдает ту из  $2^n$  информационных переменных, номер которой задается значениями адресных переменных.

$$\mu(x_1, \dots, x_n, y_0, \dots, y_{2^n-1}) = \bigvee_{\sigma_1, \dots, \sigma_n \in B^n} x_1^{\sigma_1} \dots x_n^{\sigma_n} y_{v(\sigma_1, \dots, \sigma_n)}$$

Таким образом, были получены нижние оценки.

## "Более высокие" нижние оценки сложности отдельной ФАЛ. Пример

**Утверждение 10.2.** Если для существенной БП  $x_n$  ФАЛ  $f \in P_2(n)$ , и для любого  $\sigma$ ,  $\sigma \in B$ , ФАЛ:

$$f|_{x_n=\sigma} = f(x_1 \dots x_{n-1}, \sigma) \neq 0, 1$$

то справедливо неравенство:

$$L_{\&, \vee}^C(f) \geq \min \{L_{\&, \vee}^C(f|_{x_n=0}), L_{\&, \vee}^C(f|_{x_n=1})\} + 2$$

**Доказательство.**

$$f_0(x_1, \dots, x_{n-1}) = f(x_1, \dots, x_{n-1}, 0) \neq 0, 1$$

$$f_1(x_1, \dots, x_{n-1}, 1) = f(x_1, \dots, x_{n-1}, 1) \neq 0, 1$$

Так как это существенные переменные, они отличаются друг от друга. Эта техника называется техника забивания переменных константами.

Пусть  $\Sigma$  — приведенная СФЭ, реализующая функцию  $f$ .

$$\Sigma = \Sigma(x_1, \dots, x_n; z_1)$$

Существует  $(x_n - z_1)$  цепь  $C$  в схеме  $\Sigma$ . Следовательно:

$$L_{\&, \vee}(C) \geq 2$$

Так как при:

$$L_{\&, \vee}(C) = 0 \rightarrow f \equiv x_n^\sigma$$

$$L_{\&, \vee}(C) = 1 \rightarrow f \equiv (x_n^\sigma \square g)^\tau$$

Существует  $\gamma$ , при котором:

$$f(x_1, \dots, x_n, \gamma) \equiv \text{const}$$

Это противоречит данным требованиям. Это означает, что имеется дело с 2 элементами либо конъюнкции, либо дизъюнкции.

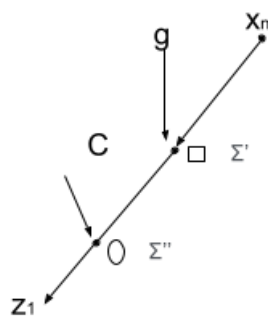


Рис. 10.2. Цепь  $C$

Необходимо доказать, что существует  $\delta \text{in} B$  такая, что для схемы  $\Sigma'$ , получившаяся из схемы  $\Sigma$  подстановкой константы  $\delta$  вместо переменной  $x_n$ , справедливо неравенство:

$$L_{\&,v}(\Sigma') \leq L_{\&,v}(\Sigma) - 2$$

■

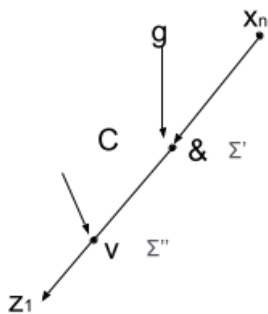


Рис. 10.3. Цепь  $C$  с конъюнкцией и дизъюнкцией

В следствии получается:

$$L^C(\mu_n) \geq 2^{n-1} + n - 1$$

$$D(\mu_n) \geq n + 2$$

**Доказательство.**

Необходимо подставить константу  $\sigma_0, \sigma_1, \dots, \sigma_{2^n-2}$  сначала в  $y_0, y_1, \dots, y_{2^n-2}$ .

$$g = \mu_n(x_1, \dots, x_n, \sigma_0, \sigma_1, \dots, \sigma_{2^n-2}, y_{2^n-1})$$

$$L_{\&, \vee}(\mu_n) \geq 2(2^n - 1) + L_{\&, \vee}(g) \geq 2^{n+1} + n - 2 \geq 2^{n+1} - n - 1$$

Такая нижняя оценка справедлива для мультиплексорной функции. ■

Мультиплексорная функция порядка 1 с одной адресной переменной:

$$n = 1 \quad (\mu_1(x_1, y_0, y_1)) = \bar{x}_1 y_0 \vee x_1 y_1$$

$$L^C(\mu_1) = 4$$

$$D(\mu_1) = 3$$

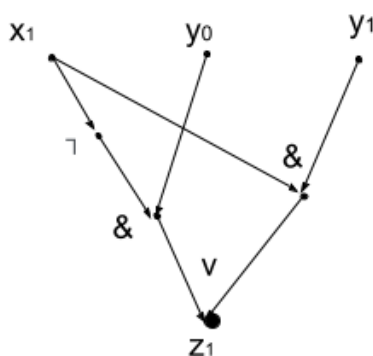


Рис. 10.4. Мультиплексорная функция порядка 1

## Нижняя оценка сложности вектор-функции. Пример

**Утверждение 10.3.**

$$F = (f_1, \dots, f_m) \in P_2^n(n)$$

$$f_i \neq f_j \quad f_1 \neq 0, 1$$

Следовательно справедливо неравенство:

$$L^K(F) \geq \frac{\sum_{j=1}^m |N_{f_j}|}{2^{n-1}}$$

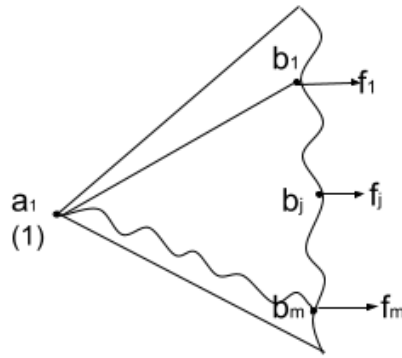


Рис. 10.5.  $\Sigma - (1, m)$  КС

**Доказательство.**

Пусть  $\Sigma - (1, m)$  КС, т.е. схема имеет следующий вид:

$$\Sigma = \Sigma(x_1, \dots, x_n; a_1; b_1, \dots, b_m)$$

$$L^K(F) = \min L(\Sigma)$$

$$L^K(F) \geq m$$

$$L^K(\vec{J}_n) \geq m$$

$$\vec{J}_n = (\bar{x}_1 \vee \dots \vee \bar{x}_n, \dots, x_1 \vee \dots \vee x_n)$$

$$|N_{f_j}| = 2^n - 1 \quad m = 2^n \rightarrow L^K(\vec{J}_n) \geq \frac{2^n(2^n - 1)}{2^{n-1}} = 2^{n+1} - 2$$

Для системы элементарных дизъюнкций оценка оказалась в 2 раза выше, чем оценка из утверждения.

$$n = 2 \quad L(\vec{J}_2) \geq 6$$

■

**Доказательство.**

Пусть  $\Sigma$  — приведенная КС. Следовательно схема  $\Sigma$  — связный граф. Определяется граф  $\Sigma|_\alpha$  для  $\alpha = (\alpha_1, \dots, \alpha_n) \in B^n$ , получившийся из  $\Sigma$  удалением контактов  $x_1^{\bar{\alpha}_1}, \dots, x_n^{\bar{\alpha}_n}$ . Записывается число тех функций, которые равны 1 на наборе  $\alpha$ .

$$M(\alpha) f_1(\alpha) + \dots + f_m(\alpha)$$

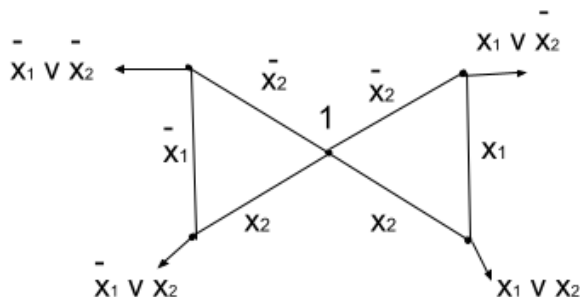


Рис. 10.6. Схема сложности 6

Следовательно, в  $\Sigma|_{\alpha}$  существует связный подграф, содержащий вход и те выходы, на которых функция равна 1.

$$f_j(\alpha) = 1$$

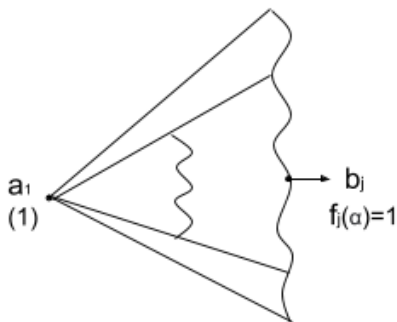


Рис. 10.7. Связный подграф

Число ребер записывается следующим образом:

$$|E(\Sigma|_{\alpha})| \geq M(\alpha)$$

$$\sum_{\alpha \in B^n} |E(\Sigma|_{\alpha})| \geq |N_{f_1}| + \dots + |N_{f_m}|$$

■

## Верхние оценки некоторых ФАЛ

**Утверждение 10.4.** Для любого натурального  $n$  выполняются неравенства:

$$L^C(I_n) \leq 4n - 4$$



$$L^C(\bar{l}_n) \leq 4n - 4 + \left\lceil \frac{1}{n} \right\rceil$$

$$L^\pi(\mu_n) \leq 3 \cdot 2^n - 2$$

$$L^\Phi(\mu_n) \leq 2^{n-2} - 3$$

$$L^C(\vec{Q}_n) \leq 2^n - O\left(n \cdot 2^{\frac{n}{2}}\right)$$

$$L^K(\vec{Q}_n) \leq 2^{n-1} - 2$$

$$L^C(\vec{J}_n) \leq 2^n - O\left(n \cdot 2^{\frac{n}{2}}\right)$$

Пусть  $f$  — линейная функция:

$$f = l_n(x_1, \dots, x_n), \quad f = \bar{l}_n$$

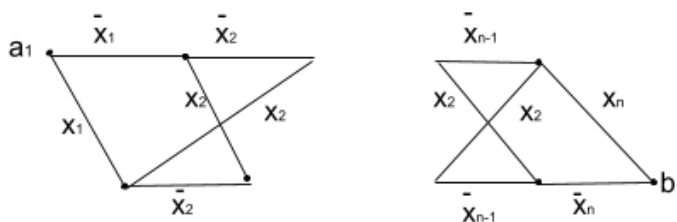


Рис. 10.8. Схема Кардо

Схема Кардо реализует сложение по модулю 2  $n$  переменных. Эта схема имеет сложность:

$$L = 4n - 4$$

Следовательно, получается контактная сложность для линейной функции:

$$L^K(l_n) \leq 4n - 4 \quad n \geq 2$$

Сложность СФЭ схемы равна 4. Таким образом, можно построить цепочку из таких схем. Следовательно, схемная сложность линейной функции записывается следующим образом:

$$L^C(l_n) \leq 4n - 4 \quad n \geq 2$$

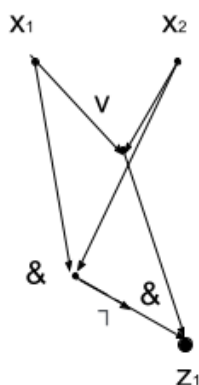


Рис. 10.9. СФЭ сложности линейной функции

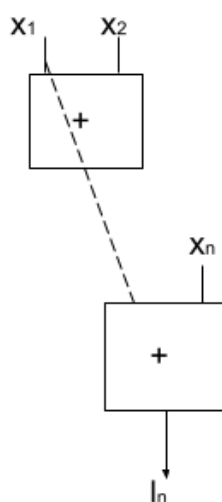


Рис. 10.10. Цепочка линейной функции

Сложность мультиплексорной функции записывается как:

$$L^{\pi}(\mu_n) \leq 3 \cdot 2^n - 2$$

Если промоделировать ее в классе формул, то получается, что формульная сложность мультиплексорной функции записывается как:

$$L^{\Phi}(\mu_n) \leq 4 \cdot 2^n - 3$$

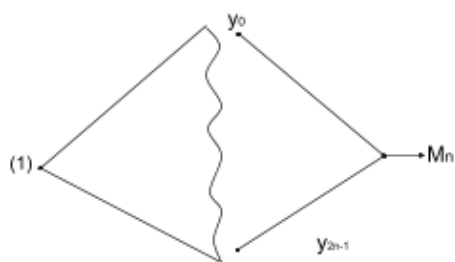


Рис. 10.11.  $\pi$  - схема мультиплексорной функции

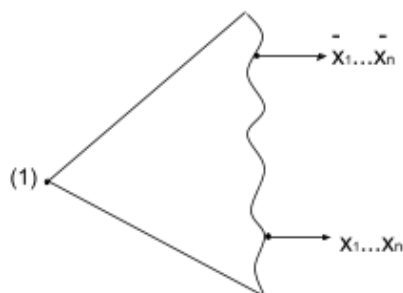


Рис. 10.12. Контактная сложность конъюнктивного дешифратора

В следствии, получается:

$$L^C(\vec{Q}_n) \sim L^C(\vec{J}_n) \sim 2^n$$

## Лекция 11. Суперпозиции различных схем

### Суперпозиция систем ФАЛ. Суперпозиция схем

В моделях есть не только формульные представления, но и представления в виде схем. Операцию суперпозиции можно записать следующим образом. Пусть есть система функций  $F$ , состоящая из функций  $f_1 \dots f_m$ , которая представляет из себя суперпозицию системы функций  $F''$ .

$$f_i = f_i''(f'_1, \dots, f'_q)$$

$$F(f_1, \dots, f_m) = F''(F')$$

$$F' = (f'_1, \dots, f'_q)$$

$$F'' = (f''_1, \dots, f''_m)$$

Каждая функция зависит от  $q$  переменных. Схема  $\Sigma''$  реализует систему функций  $F''$ , а схема  $\Sigma'$  реализует  $F'$ . Необходимо построить схему, которая реализует суперпозицию этих двух систем функций.

$$\Sigma = \Sigma''(\Sigma')$$

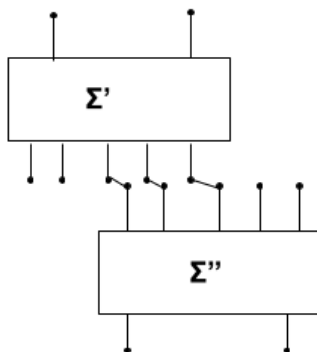


Рис. 11.1. Суперпозиция  $\Sigma$

Суперпозиция происходит тогда, когда часть выходов схемы  $\Sigma'$  присоединяется к входам  $\Sigma''$ . Эти вершины становятся внутренними схемы суперпозиции  $\Sigma$ . Входами суперпозиции являются все входы  $\Sigma'$  и свободные входы  $\Sigma''$ . Выходами суперпозиции являются все выходы  $\Sigma'$  и  $\Sigma''$ .

Для схем  $\Sigma$ ,  $\Sigma'$ ,  $\Sigma''$  из какого-то класса схем  $U$  необходимо определить схему  $\Sigma$  как суперпозицию схем  $\Sigma'$  и  $\Sigma''$  тогда и только тогда, когда  $\Sigma$  получается в результате присоединения с возможным отождествлением части входов схемы  $\Sigma''$  к некоторым выходам  $\Sigma'$ . При присоединении происходит снятие соответствующих вход-выходных пометок.

## Простейшая схема. Частные случаи суперпозиции

Есть простейшие схемы, которая состоит только из выходных вершин, с которых начинается операция суперпозиции. Вход-выходная вершина имеет 2 пометки.

### Частные случаи

- 1) Переименование с возможным отождествлением входов тогда и только тогда, когда схема  $\Sigma'$  — схема вход-выходных вершин.
- 2) Переименование с возможным дублированием и снятием выходов схемы  $\Sigma'$  тогда и только тогда, когда схема  $\Sigma''$  — схема вход-выходных вершин.
- 3) Объединение не пересекающихся схем.
- 4) Стыковка двух схем происходит, когда число выходов первой схемы совпадают с числом входов второй схемы. Стыковка может быть неповторной тогда и только тогда, когда все участвующие в суперпозиции входы и выходы соответствуют разным вершинам. Любая стыковка неповторна, если допускаются проводниковые элементы.

## Правильная и корректная суперпозиции

Пусть схема  $\Sigma$  является суперпозицией двух схем  $\Sigma''$  от  $\Sigma'$ , которые реализуют систему функций  $F$ ,  $F'$ ,  $F''$ .

$$\Sigma = \Sigma''(\Sigma')$$

Суперпозиция называется правильной тогда и только тогда, когда система функций  $F$  является суперпозицией соответствующих систем функций.

$$F = F''(F')$$

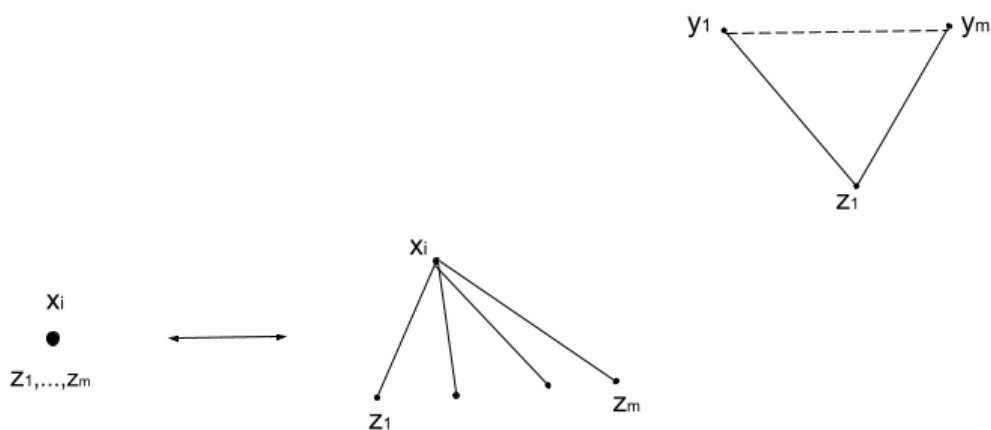


Рис. 11.2. Проводниковые элементы

Суперпозиция называется корректной тогда и только тогда, когда система функций  $F$  является суперпозицией соответствующих систем функций и на выходах подсхемы  $\Sigma'$  в схеме  $\Sigma$  реализуется та же самая система функций  $F'$ , что и в отдельно взятой схеме  $\Sigma'$ .

### Суперпозиция СФЭ. Примеры

Любая суперпозиция СФЭ является корректной. Вводятся 2 СФЭ, которые реализуют сложение по модулю 2 двух переменных и эквивалентность двух переменных. Схема, которая реализует сложение по модулю 2, обозначается как  $\Sigma_{\oplus}^{(2)}$ . Схема, реализующая эквивалентность двух переменных, обозначается как  $\Sigma_{\sim}^{(2)}$ .

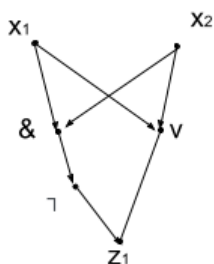


Рис. 11.3. Сложение по модулю 2

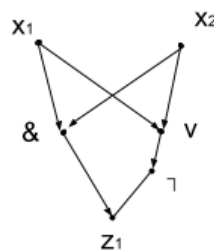


Рис. 11.4. Эквивалентность двух переменных

Если нужна схема  $\Sigma_{\oplus}^{(n)}$ , которая реализует линейную функцию, то можно взять и рассмотреть многократную суперпозицию схем  $\Sigma_{\oplus}^{(2)}$ .

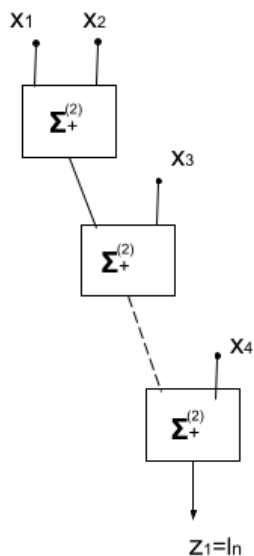


Рис. 11.5. Суперпозиция  $\Sigma_{\oplus}^{(n)}$  для линейной функции

Сложность этой функции записывается следующим образом:

$$L(\Sigma_{\oplus}^{(n)}) = 4n - 4, \quad n \geq 2$$

### Суперпозиция КС. Пример

Пусть есть суперпозиция с неповторной стыковкой  $\Sigma = \Sigma''(\Sigma')$ , контактная схема  $\Sigma'$  с одним входом и  $q$  выходами и контактная схема  $\Sigma''$  с  $q$  входами и одним выходом.

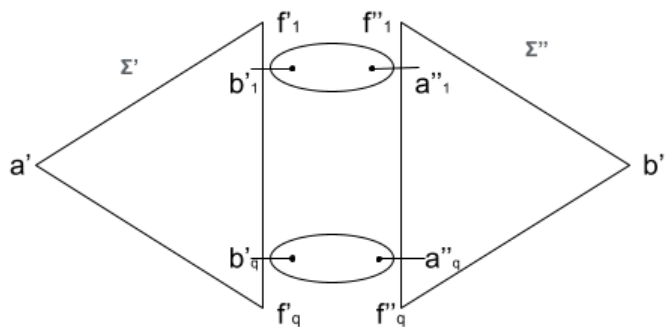


Рис. 11.6. Суперпозиция КС

Схема с  $p$  входами и  $q$  выходами реализует матрицу из функций. В этой

матрице  $p$  строк и  $q$  столбцов. Но в этом случае это будет система функций — вектор строка из функций, реализуемая схемой  $\Sigma'$ .

$$F' = (f'_1, \dots, f'_q)$$

Схема  $\Sigma''$  реализует матрицу — вектор столбец из функций:

$$F'' = \begin{pmatrix} f''_1 \\ \cdot \\ \cdot \\ \cdot \\ f''_q \end{pmatrix}$$

$f''_1$  — это функция проводимости от первого входа до к единственному выводу. Пометки входов схемы  $\Sigma''$  трактуются как Булевская переменная  $a''_1 \rightarrow y''_1$ . А на выходе реализуется 1 тогда и только тогда, когда есть проводящая цепь, идущая от одного из входов, которому приписана переменная  $y = 1$  на функцию проводимости от входа к выходу этой схемы. Вектору столбцу сопоставляется следующее выражение:

$$y''_1 f''_1 \vee \dots \vee y''_q f''_q$$

В случае правильности суперпозиции, схема  $\Sigma$  реализует функцию проводимости от входа  $a'$  к выходу  $b''$ .

$$f'_1 f''_1 \vee \dots \vee f'_q f''_q$$

В результате получается произведение матриц:  $F' \cdot F''$ .

Пусть есть суперпозиция с бесповторной стыковкой  $\Sigma = \Sigma''(\Sigma')$ , контактная схема  $\Sigma'$  с  $p$  входами и  $q$  выходами и контактная схема  $\Sigma''$  с  $q$  входами и  $s$  выходами. Тогда  $F'$  — это матрица с  $p$  строками и  $q$  столбцами, а  $F''$  — это матрица с  $q$  строками и  $s$  столбцами. Элементы матриц — это функции из  $P_2(n)$ .

Функция проводимости описывается произведением  $i$  строки матрицы  $F'$  на  $j$  столбец матрицы  $F''$ . В этом случае схема  $\Sigma$  будет реализовать такую же матрицу, но с размером  $p$  строк и  $s$  столбцов.

$$F' \cdot F''$$



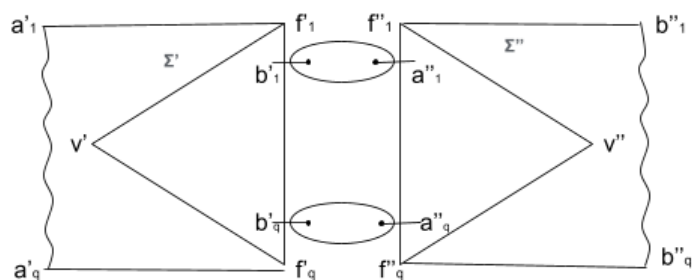


Рис. 11.7. Суперпозиция КС

## Лемма Шеннона и её следствие. Разделительность КС.

### Пример

Пусть матрица  $F$  реализуется контактной схемой  $\Sigma$ , которая описывается функцией проводимости от  $a'_i$  к  $b''_j$ . Необходимо доказать, что выполняется следующее неравенство:

$$F \geq F' \cdot F''$$

Также необходимо доказать, что схема правильная.

$$F = F' \cdot F''$$

Лемма Шеннона гласит о том, что эта суперпозиция действительно является правильной, если либо схема  $\Sigma'$  разделительна по выходам, либо схема  $\Sigma''$  разделительна по входам. Контактная схема  $\Sigma'$  разделительна по выходам, если функции проводимости между любыми ее двумя выходами равны 0. Контактная схема  $\Sigma''$  разделительна по входам, если функции проводимости между различными входами равна 0.

### Доказательство.

В всей схеме  $\Sigma$  берутся две произвольные вершины  $v'$  и  $v''$ . Необходимо найти функцию проводимости  $g$  от  $v'$  к  $v''$ .

$$v' \in V(\Sigma') \quad v'' \in V(\Sigma'')$$

$$g_{v',v''}(x_1, \dots, x_n)$$

Берутся функции проводимости от вершины  $v'$  к выходам схемы  $\Sigma'$  и функции проводимости от входов схемы  $\Sigma''$  к вершине  $v''$ .

$$g_{v',v''}(x_1, \dots, x_n) \geq g'_1 g''_1 \vee \dots \vee g'_q g''_q$$

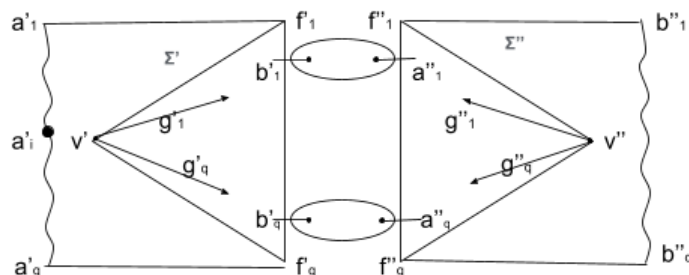


Рис. 11.8. Лемма Шеннона

Если  $\Sigma'$  разделительна по выходам или схема  $\Sigma''$  разделительна по входам, то справедливо равенство:

$$g_{v',v''}(x_1, \dots, x_n) = g'_1 g''_1 \vee \dots \vee g'_q g''_q$$

Это означает, что матрица  $F$  в точности совпадает с произведением двух матриц. Следовательно, суперпозиция является правильной. ■

Корректность суперпозиции требует, чтобы на выходах подсхемы  $\Sigma'$  реализовались те же самые функции, что и в самом начале. Это обеспечиваются разделительностью схемы. В качестве  $v'$  обозначается вход и один из входов схемы  $\Sigma''$  в качестве вершин  $v''$ . Необходимо найти функцию проводимости от  $v'$  к  $v''$ . Эта функция может увеличиться при суперпозиции за счет проведения цепи проводимости, которая обходит вход  $v''$ .

При общей суперпозиции у схемы  $\Sigma''$  могут склеиться какие-то входы. Тогда функция проводимости записывается следующим образом:

$$g_{a''b''} = f''_i \vee f''_j$$

Это правильная суперпозиция с отождествлением двух входов.

Отождествление входов (выходов) у разделительной по входам (выходам) КС дает разделительную по рассматриваемой группе полюсов КС.

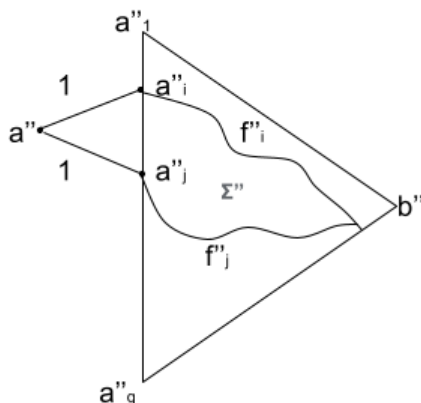


Рис. 11.9. Суперпозиция с отождествлением двух входов

### Операция присоединения контактов к выходу КС. Каскадные контактные схемы (ККС). Пример

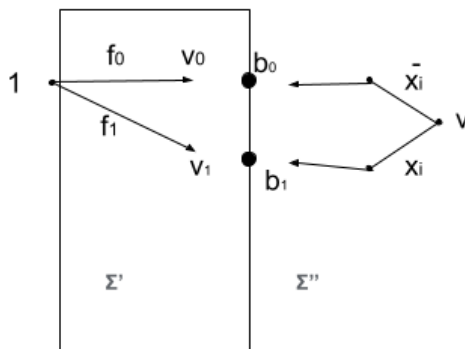


Рис. 11.10. Присоединение 2 контактов к КС

Каскадная контактная схема — это контактная схема, получающаяся из изолированных вход-выходных вершин с помощью операция присоединения и возможно снятия выхода.

### Утверждение о существовании ККС, реализующей систему из отрицаний ФАЛ

**Утверждение 11.1.** Если  $(1, m)$  - ККС  $\Sigma'$  реализует систему ФАЛ  $F' = (f'_1, \dots, f'_m)$ , то существует  $(1, m)$  - ККС  $\Sigma''$ , которая реализует систему ФАЛ  $\bar{F}' = (\bar{f}'_1, \dots, \bar{f}'_m)$

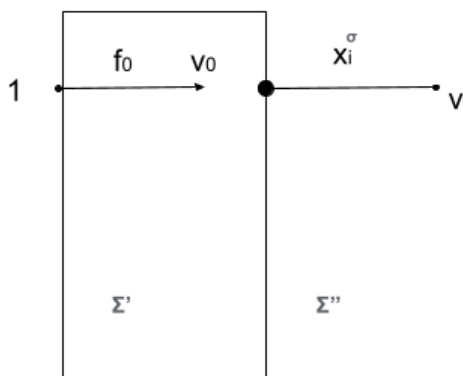


Рис. 11.11. Присоединение 1 входа к КС

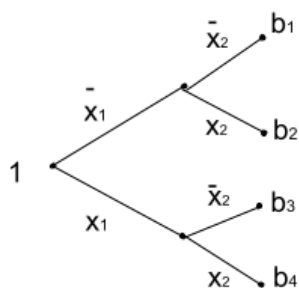


Рис. 11.12. (1,4) КД от  $x_1, x_2$

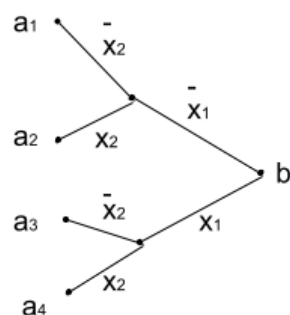


Рис. 11.13. (4,1) КД от  $x_1, x_2$

и для которой:

$$L(\Sigma'') \leq 2L(\Sigma')$$

## Лекция 12. Каскадная СФЭ

В основе каскадного метода лежит каскадная контактная схема. Пусть есть  $\Sigma - (p, q)$  — контактная схема от переменных  $X(n) = \{x_1, \dots, x_n\}$ . Эта схема называется каскадной контактной схемой тогда и только тогда, когда  $\Sigma$  получена в результате последовательного применения операций (корректная суперпозиция) присоединения двух противоположных или одного контакта к выходам контактной схемы к системе изолированных вход-выходных вершин.

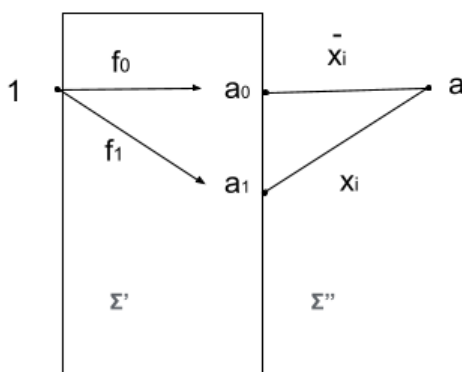


Рис. 12.1. Присоединение двух противоположных контактов

$$\Sigma = \Sigma''(\Sigma')$$

В вершине  $a$  реализуется следующая функция:

$$f = f_0 \bar{x}_2 \cdot f_1 x_2$$

В вершине  $v$  реализуется функция:

$$f = x_i^\sigma \cdot f_\sigma$$

### Свойства ККС. Полная ККС. Частный случай

Каскадная контактная схема обладает следующими свойствами:

- 1) Разделительна по входам. Функция проводимости между различными ее входами равна 0.

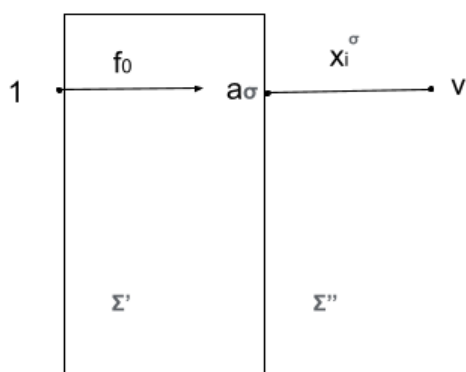


Рис. 12.2. Присоединение одного контакта

- 2) В каждой вершине  $v$  в схеме  $\Sigma$  реализуется столбец из  $p$  ортогональных ФАЛ от входов этой схемы к выходу.
- 3) Схема  $\Sigma$  является полной каскадной КС тогда и только тогда, когда при ее построении использовалась только операция присоединения двух противоположных контактов.

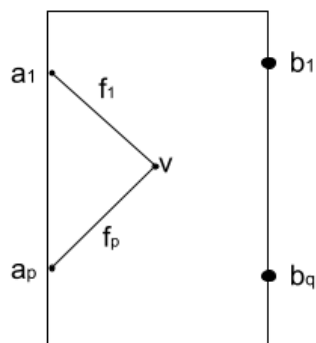


Рис. 12.3. Схема  $\Sigma$

$$f_1 \vee \dots \vee f_p \equiv 1$$

Если контактная схема имеет 2 входа, то в любой вершине  $v$  реализуются функция и ее отрицание.

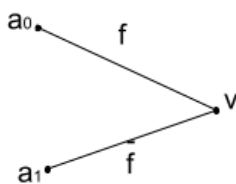


Рис. 12.4. КС  $\Sigma$  с  $p = 2$

### Утверждение о существовании ККС, реализующей систему из отрицаний ФАЛ

**Утверждение 12.1.** Если  $(1, m)$  — ККС  $\Sigma$  реализует систему функций  $F = (f_1, \dots, f_m)$ , то, следовательно, существует  $(1, m)$  — ККС  $\Sigma'$ , которая реализует систему функций  $F' = (\bar{f}_1, \dots, \bar{f}_m)$  причем сложность этой схемы имеет следующий вид:

$$L(\Sigma') \leq 2L(\Sigma)$$

**Доказательство.**

Необходимо построить  $(2, m)$  — ККС  $\hat{\Sigma}$ , которая является полной ККС.

$$\hat{\Sigma} \supseteq \Sigma$$

При построении схемы  $\Sigma$  в вершине  $v'$  происходит присоединение двух противоположных контактов, а в  $w$  происходит присоединение одного контакта.

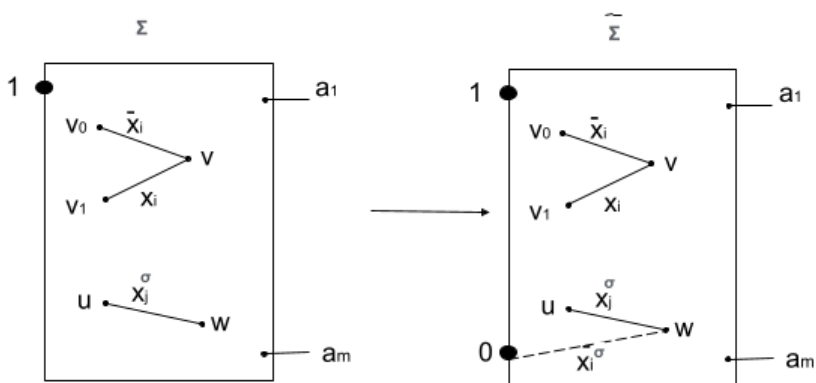


Рис. 12.5. Построение  $\hat{\Sigma}$  из  $\Sigma$

Схема  $\hat{\Sigma}$  является полной каскадной контактной схемой, содержит схему  $\Sigma$  в качестве подсхемы и функция проводимости от входа 1 к выходам схемы  $\Sigma$  точно такая же, как функция проводимости от входа с пометкой 1 к тем же самым выходам в схеме  $\hat{\Sigma}$ . ФАЛ проводимости от входа 1 к выходу  $a_j$  в схеме  $\hat{\Sigma}$  равна  $f_j$ . ФАЛ проводимости от входа 0 к выходу  $a_j$  в схеме  $\hat{\Sigma}$  равна  $\bar{f}_j$ .

$$\Sigma' = \hat{\Sigma}(x_1, \dots, x_n; 0; a_1, \dots, a_m)$$

Применяется операция приведения. В результате, получается схема  $\hat{\Sigma}'$ , которая реализует систему функций  $F' = (\bar{f}_1, \dots, \bar{f}_m)$ .

$$L(\hat{\Sigma}') \leq 2L(\Sigma) \rightarrow L(\Sigma') \leq 2L(\Sigma)$$

$\Sigma'$  — это инверсная относительно исходной схемы  $\Sigma$  ККС. ■

### Пример с конъюнктивным и дизъюнктивным контактными деревьями

Строится контактное дерево, которое реализует систему всех элементарных конъюнкций  $\vec{Q}(n)$ .

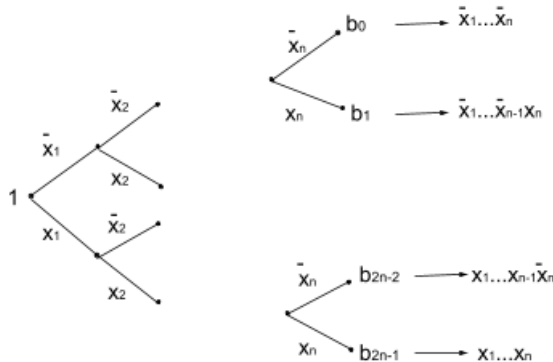


Рис. 12.6.  $(1, 2^n)$ — КД

$$L = 2^{n+1} - 2$$

Чтобы перейти в каскадную схему, необходимо ввести полюс 0 и применить присоединение двух противоположных контактов. Из каждой вершины тянется противоположный контакт в входную вершину с номером 0. В результате получится инверсная ККС — дизъюнктивное контактное дерево (ДКД).



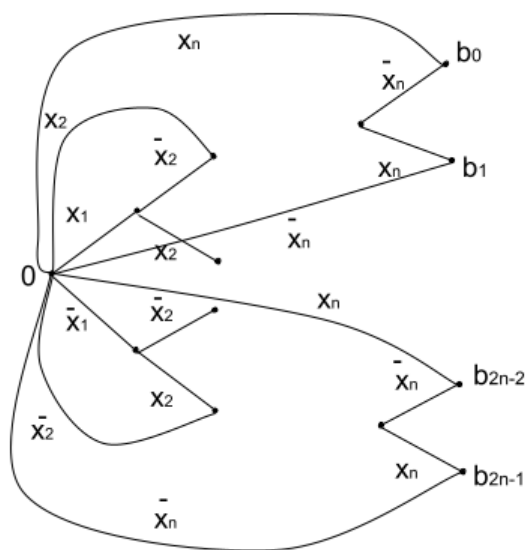


Рис. 12.7. Инверсная ККС – ДКД

$$L = 2^{n+2} - 2$$

### Каскадная СФЭ. Примеры

Каскадная СФЭ  $S$  структурно моделирует  $(1, m)$  – ККС  $\Sigma$ . Моделируются ККС только с одним входом. При построении вершина  $v$  была получена при присоединении двух противоположных контактов. Операция присоединения двух противоположных контактов в СФЭ моделируется добавлением трех функциональных элементов. Реализуется функция:

$$f = f_m \bar{x}_1 \vee f_1 x_i$$

При присоединении одного контакта реализуется функция:

$$g \cdot x_i^\sigma$$

Если контактной вершиной является входная вершина, то реализуется функция:

$$f_0 \cdot \bar{x}_i \vee x_i = f_0 \vee x_i$$

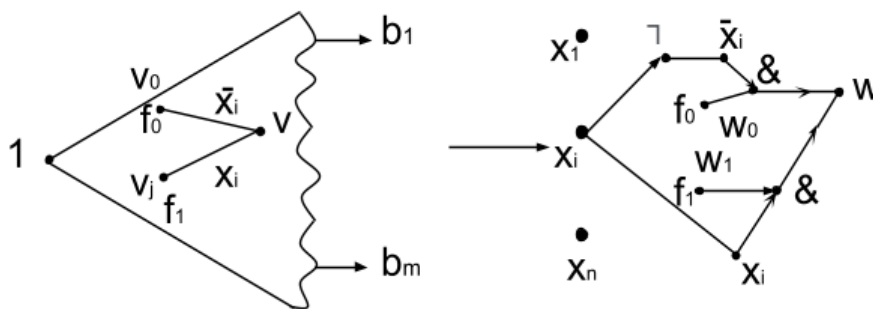


Рис. 12.8. Моделирование СФЭ при присоединении 2 противоположных контактов

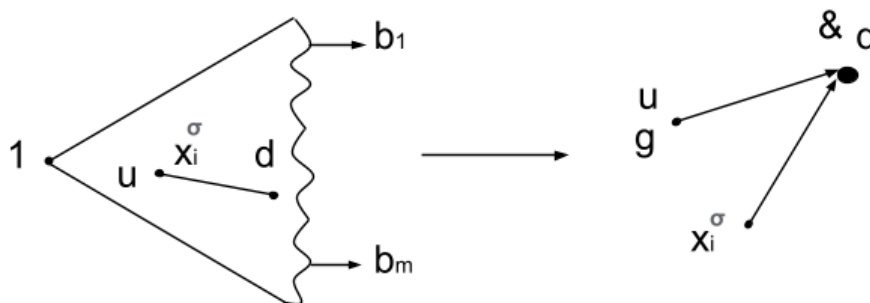


Рис. 12.9. Моделирование СФЭ при присоединении 1 противоположного контакта

При моделировании СФЭ вместо трех функциональных элементов используется один элемент дизъюнкции. Это специфика каскадных СФЭ. Элементы конъюнкции можно использовать несколько раз, если это соответствует функционированию схемы.

### Метод каскадов в общем виде. Пример

$$F = (f_1, \dots, f_m) \text{ in } P_2^m(n) : f_s \neq f_j \quad s \neq j$$

Каждая переменная является существенной хотя бы для одной функции. Метод каскадов позволяет построить  $(1, m)$  — ККС по этой системе функций, которая реализует эту систему функций  $F$ , и каскадную схему из функциональ-

ных элементов  $S_F$ .

Сначала происходит разложение нужных функций по переменным.

$$\forall i = 1, \dots, n \quad G_i = \{g(x_i, \dots, x_n) : \exists 1 \leq j \leq m \exists \sigma_1, \dots, \sigma_{i-1} : g = f_j(\sigma_1, \dots, \sigma_{i-1}, x_i)\}$$

Из множества  $G_i$  определяется множество  $\hat{G}_i$  и функции, которые зависят от  $x_i$ .

$$\hat{G}_i = \{g \in G_i\}$$

$$G_1 = \{f_1, \dots, f_m\}$$

$$\hat{G}_1 = \{f_j\}$$

$$\hat{G}_n \subseteq \{\bar{x}_n, x_n\}$$

Дальше строится схема. Каждому из множеств сопоставляется система точек.

$$g = \bar{x}_i \cdot g_0 \vee x_i \cdot g_1$$

$$g_0 = g(0, x_{i+1}, \dots, x_n) \in \hat{G}_{i'} \quad i' < i$$

$$g_1 = g(1, x_{i+1}, \dots, x_n) \in \hat{G}_{i''} \quad i'' < i$$

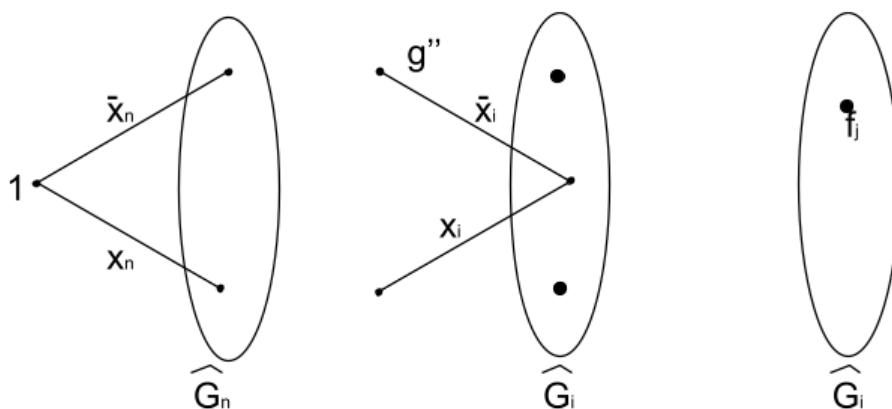


Рис. 12.10. ККС  $\Sigma_F$

**Пример 12.1.** Рассматривается система 2 функций от 4 переменных.

$$F = (f_1, \dots, f_m)$$

$$f_1 = \bar{x}_1 x_2 (x_3 \oplus x_4) \vee x_1 (x_2 \vee x_3 \bar{x}_4)$$

$$f_2 = x_1 (x_3 \oplus x_4) \vee \bar{x}_1 x_4$$

Строятся множества:

$$\hat{G}_1 = \{f_1, f_2\}$$

$$\hat{G}_2 = \{x_2(x_3 \oplus x_4), x_2 \vee x_3 \bar{x}_4\}$$

$$\hat{G}_3 = \{x_3 \oplus x_4, x_3 \cdot \bar{x}_4\}$$

$$\hat{G}_4 = \{\bar{x}_4, x_4\}$$

На основе этих разложений строится схема.

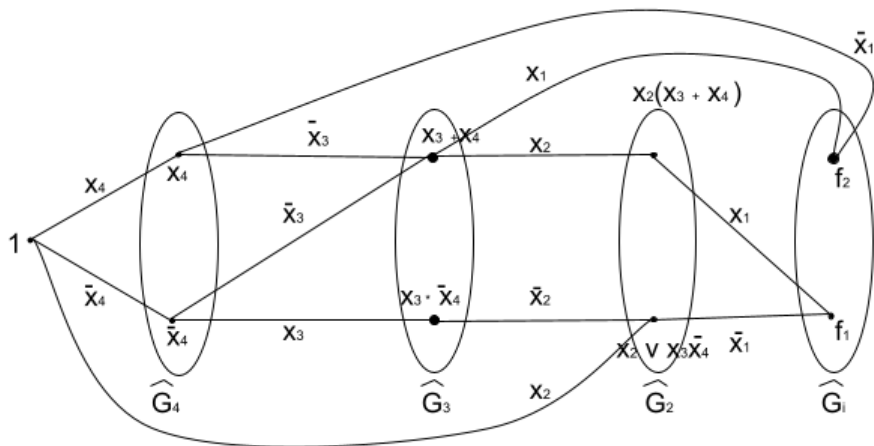


Рис. 12.11. КСФЭ  $S_F$

## Оценки сложности некоторых каскадных схем

**Утверждение 12.2.**

$$\forall n = 1, 2, \dots \quad L^K(I_n^\sigma) \leq 4n - 4$$

$$L^K(\vec{J}_n) \leq 4 \cdot 2^n - 6$$

Верхняя оценка для сложности каскадной схемы записывается следующим образом:

$$L^K(\vec{P}_2(n)) \leq 2 \cdot 2^{2^n}$$

## Метод Шеннона

**Утверждение 12.3.** Верхняя оценка функции Шеннона для контактной схемы:

$$L^K(n) \lesssim 4 \cdot \frac{2^n}{n}$$

Верхняя оценка функции Шеннона для СФЭ:

$$L^C(n) \lesssim 8 \cdot \frac{2^n}{n}$$

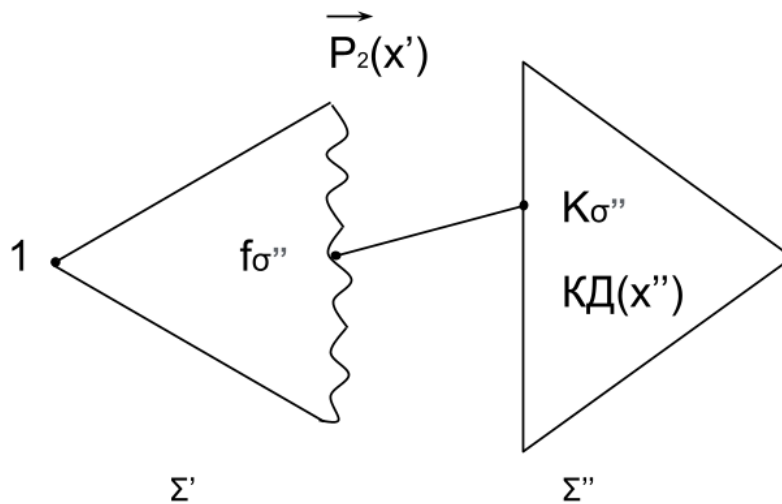


Рис. 12.12. Схема  $\Sigma_f$

**Доказательство.**

Вводятся промежуточные переменные.

$$\forall f(x_1, \dots, x_q, x_{q+1}, \dots, x_n) = \bigvee_{\sigma''=(\sigma_{q+1} \dots \sigma_n)} K_{\sigma''}(x'') \cdot f_{\sigma''}(x')$$

Сначала строится контактная схема  $\Sigma_f$ , которая реализует функцию  $f$ .

$$\Sigma_f = \Sigma''(\Sigma')$$

$\Sigma'' = (2^{n-q}, 1)$  – контактное дерево, а  $\Sigma'$  – универсальный контактный многополюсник от переменных группы  $x'$ .

Корректная операция суперпозиции записывается как:

$$\Sigma_f = \Sigma''(\Sigma')$$

Параметры должны быть следующими:

$$q = \lceil \log(n - 2 \log n) \rceil$$
$$L \leq 2 \cdot 2^{2q} + 2 \cdot 2^{n-q} \leq 4 \cdot \frac{2^n}{n} (1 + o(1))$$

■

## Лекция 13. Асимптотически наилучший метод синтеза СФЭ

### Нижние мощностные оценки функций Шеннона

Нижние оценки имеют линейный порядок роста, а верхние оценки имеют экспоненциальный порядок роста. Мощностной принцип Шеннона позволяет получить более высокие нижние оценки. Пусть есть  $U$  — класс схем,  $\psi$  — функционал сложности схем из  $U$ ,  $\psi(f)$  —  $\psi$  сложность функции  $f$ ,  $\psi(n)$  — функция Шеннона. Индивидуальные функции дают очень слабые нижние оценки порядка  $2^n$ . Вводится множество схем  $\Sigma$  из  $U$  ( $U(\tilde{\psi}, n)$ ), реализующих одну функцию из множества  $P_2(n)$  таких, что:

$$\psi(\Sigma) \leq \tilde{\psi}$$

Число попарно не эквивалентных схем в этом множество ( $||U(\tilde{\psi}, n)||$ ) — это число тех функций  $f$  от  $n$  переменных, которые можно реализовать схемами из этого класса. Рассматривается случай, когда  $\tilde{\psi}$  заменяется функцией Шеннона ( $\psi(n)$ ):

$$||U(\psi(n), n)|| = 2^{2^n}$$

Таким образом, получается первое мощностное соотношение Шеннона. Если выполняется следующее неравенство:

$$||U(\tilde{\psi}, n)|| \leq \delta \cdot 2^{2^n}, \quad 0 < \delta \leq 1$$

то тогда доля тех функций  $f$  из множества  $P_2(n)$ , для которых  $\psi(f) > \tilde{\psi}$ , не меньше, чем  $1 - \delta$ .

**Утверждение 13.1.** Для некоторых последовательностей  $\varepsilon_i = \varepsilon_i(n)$ , где  $i = 1, \dots, 4$  и  $n = 1, 2, \dots$ , таких, что  $\varepsilon_i \geq 0$  при  $n \geq n_0$  и  $\varepsilon_i(n)$  стремится к 0 при  $n$  стремящимся к бесконечности для почти всех ФАЛ  $f$ ,  $f \in P_2(n)$ , выполняются неравенства:

$$L^K(f) \geq (1 - \varepsilon_1(n)) \frac{2^n}{n}$$

$$L^C(f) \geq (1 + \varepsilon_2(n)) \frac{2^\sigma}{n}$$

$$L^\Phi(f) \geq (1 - \varepsilon_3(n)) \frac{2^n}{\log n}$$

$$D(f) \geq n - \log \log n - \varepsilon_4(n)$$

В следствии, получаются следующие неравенства:

$$D(n) \geq [n - \log \log n - o(1)]$$

$$L^C(n) \gtrsim \frac{2^n}{n}$$

$$L^\Phi(f) \gtrsim \frac{2^n}{\log n}$$

$$L^K(n) \gtrsim \frac{2^n}{n}$$

**Доказательство.**

Пусть:

$$U = U^K$$

$$\psi = L$$

$$\tilde{L} = \tilde{L}(n) = (1 - \varepsilon(n)) \frac{2^n}{n}$$

$$\|U^K(\tilde{L}, n)\| \leq (8n\tilde{L})^{\tilde{L}}$$

Таким образом, была найдена верхняя оценка. Чтобы доказать, что  $\delta \rightarrow 0$ , достаточно убедиться в том, что  $\log \delta \rightarrow -\infty$  с ростом  $n$ .

$$\delta = \frac{\|U^K(\tilde{L}, n)\|}{2^{2^n}} \rightarrow \log \delta \leq \tilde{L}(3 + \log n + n - \log n) - 2^n$$

Необходимо найти значение  $\varepsilon$ , которое обеспечит  $\delta \rightarrow 0$ .

$$\log \delta \leq (1 - \varepsilon(n)) \frac{2^n}{n} (3 + n) - 2^n \leq 2^n (1 - \varepsilon) \left(1 + \frac{3}{n}\right) - 2^n$$

$$\leq 2^n + 2^n \left(\frac{3}{n} - \varepsilon\right) - 2^n \leq 2^n \left(\frac{3}{n} - \varepsilon\right)$$

$$\varepsilon = \frac{4}{n}, \quad \log \delta \rightarrow -\infty, \quad n \rightarrow \infty$$



Рассматривается класс СФЭ. Пусть:

$$U = U^C$$

$$\psi = L$$

$$\tilde{L} = \tilde{L}(n) = (1 + \varepsilon(n)) \frac{2^n}{n} - n$$

$$\hat{L} = (1 + \varepsilon(n)) \frac{2^n}{n}$$

$$\|U^C(\tilde{L}, n)\| \leq (8(\tilde{L} + n))^{\tilde{L}+1} \leq (8\hat{L})^{\hat{L}}$$

Таким образом, получается неравенство для схем из функциональных элементов. Необходимо найти значение  $\varepsilon$ .

$$0 \leq \varepsilon \leq 1, \quad \delta = \frac{\|U^C(\tilde{L}, n)\|}{2^{2^n}} \rightarrow \log n \leq \hat{L}(3 + \log \hat{L}) - 2^n$$

$$\leq (1 + \varepsilon) \frac{2^n}{n} (3 + 1 + n - \log n) - 2^n \leq 2^n (1 + \varepsilon) \left(1 - \frac{\log n}{n} + \frac{1}{n}\right) - 2^n \leq 2^n \left(1 + \varepsilon - \frac{\log n - 4}{n}\right)$$

В результате, получается верхняя оценка для  $\log \delta$ .

$$\varepsilon_2 = \frac{\log n - 5}{n}, \quad \delta \rightarrow 0$$

Рассматривается класс формул. Пусть:

$$U = U^\Phi$$

$$\psi = L$$

$$\tilde{L} = \tilde{L}(n) = (1 - \varepsilon(n)) \frac{2^n}{\log n} - 1$$

$$\|U^\Phi(\tilde{L}, n)\| \leq (8n)^{\tilde{L}}$$

$$\varepsilon_3(n) = \frac{4}{\log n} \rightarrow \delta = \frac{\|U^\Phi(\tilde{L}, n)\|}{2^{2^n}} \rightarrow 0, \quad n \rightarrow \infty$$

Рассматривается класс формул, но функционал сложности равен глубине:

$$U = U^\Phi$$

$$\psi = D$$

$$\tilde{D}(n) = n - \log(\log n + 4)$$

$$\|U^\Phi(\tilde{D}, n)\| < (8n)^{2^{\tilde{D}}}$$

$$\delta = \frac{\|U^\Phi(\tilde{D}, n)\|}{2^{2^n}} \rightarrow 0, \quad n \rightarrow \infty$$

■

**Задача синтеза схем для ФАЛ из специальных классов.**

### Мощностные оценки в таких задачах

Берутся не все функции, а подмножество функций  $Q \subseteq P_2$ , которому сопоставляется последовательность  $Q(1), Q(2), \dots, Q(n)$ .

$$Q(n) = Q \cap P_2(n)$$

Пусть  $U$  — класс схем. Вместо функции Шеннона рассматривается следующая функция:

$$\psi(Q(n)) = \max \psi(f), \quad f \in Q(n)$$

Функция Шеннона для  $\psi$  сложности в классе схем  $U$  функции из  $Q$ .

$$\psi(n) = \psi(P_2(n))$$

Получается следующее неравенство:

$$\|U(\psi(Q(n)), n)\| \leq |Q(n)|$$

Мощностной принцип остается справедливым.

$$\|U(\tilde{\psi}, n)\| \leq \delta \cdot 2^{2^n}, \quad 0 < \delta \leq 1$$

**Утверждение 13.2.** Если верно неравенство:

$$\frac{\log |Q(n)|}{\log \log |Q(n)|} \gg n$$

тогда можно написать обычную мощностную нижнюю оценку для СФЭ:

$$L^C(Q(n)) \gtrsim \frac{\log |Q(n)|}{\log \log |Q(n)|}$$

Пусть функция  $f$  симметрична по переменным  $x_1, x_2$ .

$$Q(n) = \{f(x_1, \dots, x_n)\}$$

Функция  $f$  не меняется при перестановке первых двух переменных.

$$|Q(n)| = 2^{\frac{3}{4} \times 2^n}$$

$x_1 x_2 x_3 \dots x_n$	$f$
000...0	
...	
001...1	
010...0	$a$
011...1	$a$
100...0	$a$
101...1	$a$
110...0	
111...1	

Рис. 13.1. Таблица значений

Получается мощная нижняя оценка для функции Шеннона, связанная с реализацией функций, симметричных по первым двум переменным в классе СФЭ.

$$L^C(Q(n)) \gtrsim \frac{3}{4} \frac{2^n}{n}$$

## Дизъюнктивно-универсальное множество. Метод Лупанова для синтеза СФЭ

Справедливы нижние оценки, так как нижние оценки дают истинную асимптотику роста соответствующих функций Шеннона. Верхние оценки Шеннона были улучшены математиком Лупановым Олегом Борисовичем. Метод Шеннона представляет собой:

$$x = (x_1, \dots, x_q, x_{q+1}, \dots, x_n)$$

Применяется разложения Шеннона:

$$f(x) = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x')$$

Это разложение представляется как схема в виде корректной суперпозиции, которая реализует функцию  $f$ .

$$\Sigma_f = \Sigma''(\Sigma')$$

В  $\Sigma''$  реализуется стандартная мультиплексорная функция. Вместо аргументов этой функции подставляются все возможные функции от группы переменных  $x'$ , которые реализованы схемой  $\Sigma'$ . Это универсальный многополюсник, реализующий на своих выходах все функции от переменных группы  $x'$ .

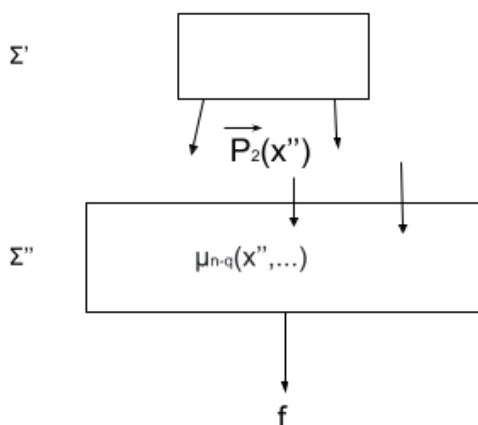


Рис. 13.2. Метод Шеннона

**Определение 13.1.** Пусть  $G \subseteq P_2(m)$  — дизъюнктивно-универсальное множество (ДУМ) порядка  $n$  и ранга  $p$  тогда и только тогда, когда  $\forall g \in P_2(m)$   
 $\exists g_1, \dots, g_p \in G : g = g_1 \vee \dots \vee g_p$ .

Пусть разбиение куба  $B^m$  записывается следующим образом:

$$\Delta = (\delta_1, \dots, \delta_p)$$

$$B^m = \delta_1 \cup \dots \cup \delta_p$$

$$\delta_1 \cap \delta_p = \emptyset$$

$$G = G^{(j)} \cup \dots \cup G^{(p)}$$

$$G^{(1)} = \{g(x_1, \dots, x_m) : g \equiv 0\}$$

Следовательно,  $G$  — стандартное ДУМ порядка  $n$  и ранга  $p$ , связанное с разбиением  $\Delta$ .

$$\forall g \in P_2(m), \quad \forall j = 1, \dots, p, \quad g_j = g \cdot X_{\delta_j} \in G^{(j)}$$

$X_{\delta_j} = \psi_j$  — характеристическая функция разбиения.

$$g = g_1 \vee \dots \vee g_p = \psi_1 g_1 \vee \dots \vee \psi_p g_p$$

Стандартный ДУМ высоты  $s$ :

$$s_1 = \dots = s_{p-1} = s \quad s_p \leq s$$

$$p = \frac{2^m}{s}$$

Мощностная характеристика записывается как:

$$|G^{(j)}| = 2^{|\delta_j|} = 2^{s_j} \leq 2^s$$

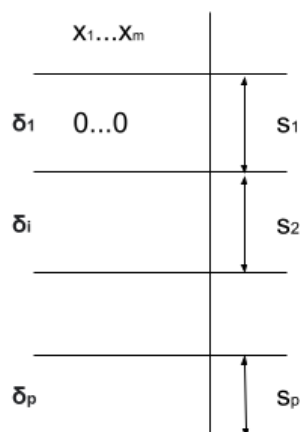


Рис. 13.3.  $G$  — стандартное ДУМ

## Асимптотически наилучшая верхняя оценка сложности для СФЭ

**Доказательство.**

Пусть  $G$  — стандартное ДУМ порядка  $q$  и высоты  $s$ .

$$p = \left\lceil \frac{2^q}{s} \right\rceil$$

$$|G| \leq p \cdot 2^s$$

Происходит разложение функции  $f$ .

$$f(x', x'') = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') f_{\sigma'}(x') = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') (g_{\sigma''}^{(1)} \vee \dots \vee g_{\sigma''}^{(p)})$$

По этому разложению можно построить улучшенную версию схемы Шеннона.

$$L(\Sigma_f) \leq p \times 2^s \times 4 \times 2^q$$

Основная сложность схемы записывается как:

$$L(\Sigma_f) \leq 4p \cdot 2^{s+q} + 2^{n-q}(p-1) + 4 \cdot 2^{n-q}$$

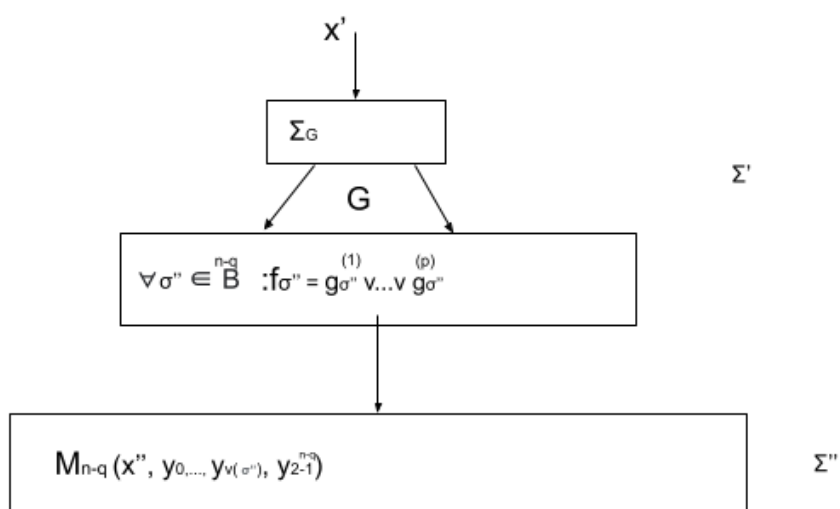


Рис. 13.4. Схема  $\Sigma_f$

$$q = \lceil \log n \rceil$$

$$s = \lceil n - 5 \log n \rceil$$

$$L(\Sigma_f) \leq \frac{2^n}{n} \left( 1 + \frac{5 \log n}{n} + o(1) \right)$$

■

## Лекция 14. Асимптотически наилучший метод синтеза формул

Метод Лупанова основан на разложении функции Шеннона  $f$  :

$$f(x_1, \dots, x_q, x_{q+1}, \dots, x_n) = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x') = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x') \left( \bigvee_{i=1}^p g_{\sigma''}^{(i)}(x') \right)$$

$G$  — ДУМ порядка  $q$  и ранга  $p$ . Это множество функций, которое позволяет получить любую функцию от первых  $q$  переменных как дизъюнкцию  $p$  функции, взятых из этого множества. Получилась следующая схема:

$$\Sigma_f = \Sigma''(\Sigma')$$

$$|G| = \lambda$$

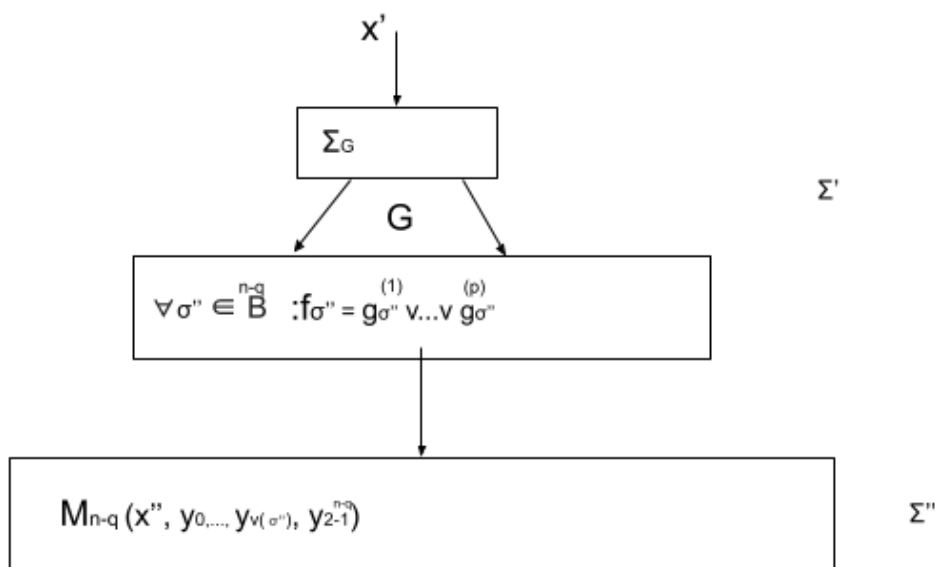


Рис. 14.1. Схема  $\Sigma_f$

Верхняя оценка асимптотически совпадает с мощностной нижней оценкой.

$$L(\Sigma_f) \lesssim \frac{2^n}{n}$$



## *m*-регулярное множество единичного куба, его свойства.

### Пример

В схеме выходы внутренних элементов могут ветвиться и использоваться несколько раз, а в формуле могут ветвиться только входы самой схемы. Необходимо разбить куб на множества, чтоб нужные функции  $g$  на каждом из этих множеств можно было промоделировать или заменить переменной или ее отрицанием. то позволяет перенести схему  $\Sigma_f$  в формульную модель реализации функции.

**Утверждение 14.1.** Для любых натуральных  $m$ ,  $\lambda$  и  $q = m + \lambda$  и для любой систем ФАЛ  $g = (g_1, \dots, g_\lambda)$  из  $P_2^\lambda(m)$  существует *m*-регулярное разбиение  $\Delta = (\delta_1, \dots, \delta_{2^q-m})$  куба  $B^q$  такое, что любая ФАЛ  $g_j$  на любой компоненте  $\delta_j$  совпадает либо с одной из БП  $x_{m+1}, \dots, x_q$ , либо с ее отрицанием.

Если в условиях утверждения:

$$\alpha = (\alpha_1, \dots, \alpha_\lambda) = v^{-1}(j-1)$$

то на  $\delta_j$  получается следующее:

$$g_i \equiv x_{i+m}^{\bar{\alpha}_j}$$

$$q = m + \lambda, \quad \delta \subseteq B^q$$

$B^q$  — *m*-регулярное множество тогда и только тогда, когда:

1)

$$|\delta| = 2^m$$

2) Любые два различных наборов из множества  $\delta$  имеют различные префиксы длины  $m$  тогда и только тогда, когда множеству  $\delta$  взаимно однозначно соответствует система функций  $g(g_1, \dots, g_\lambda) \in P_2(m)$ .

**Пример 14.1.** Значение параметров:

$$q = 6$$

$$146$$

$$m = 2$$

$$\lambda = 4$$

Вводятся переменные  $x_1, x_2, \dots, x_6$ .

$x_1x_2$	$x_3$	$x_4$	$x_5$	$x_6$
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

Рис. 14.2. Таблица значений

Начиная со второго до шестого, каждый столбец можно рассматривать как столбец значения функции от двух переменных.  $\delta$  — 2-регулярное множество в  $B^6$ . Набор из 4 функций — конъюнктивный дешифратор порядка 2  $\vec{Q}_2$ .

### Свойства

- 1)  $\forall h(x')$  совпадает на множестве  $\delta$  с некоторой функцией  $\hat{h}$  от переменных из множества  $P_2(m)$ .

$$\hat{h} = \overline{x_1 \oplus x_2}$$

- 2)  $\forall g_j$  из  $g$  совпадает на множестве  $\delta$  с БП  $x_{m+j}$ .

$$\hat{h} = \overline{x_1 \oplus x_2} \equiv x_3 \vee x_6$$

**Утверждение о существовании регулярного разбиения  
единичного куба. Продолжение примера**

**Доказательство.**

$x_1x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$h$
00	1	0	0	0	1
01	0	1	0	0	0
10	0	0	1	0	0
11	0	0	0	1	1

Рис. 14.3. Таблица значений для  $h$

$\forall i = 1, \dots, 2^\lambda$  сопоставляется набор:

$$\beta = (\beta_1, \dots, \beta_\lambda) \in B^\lambda : v(\beta) = (i - 1)$$

Необходимо определить как устроены компоненты.

$$\delta_i \leftrightarrow (g_1 \oplus \beta_1, \dots, g_\lambda \oplus \beta_\lambda)$$

$$\delta_1 \leftrightarrow (g_1, \dots, g_\lambda) = g$$

$$\beta = (1011) \quad i = 11$$

$x_1x_2$	$x_3$	$x_4$	$x_5$	$x_6$
00	0	0	1	1
01	1	1	1	1
10	1	0	0	1
11	1	0	1	0

Рис. 14.4. Таблица значений для  $\delta_{11}$

Следовательно, система подмножеств  $B^q$ , обладающая необходимыми свойствами моделирования функций  $g_j$  БП или их отрицаниями, записывается следующим образом:

$$\Delta = (\delta_1, \dots, \delta_{2^\lambda})$$

Необходимо доказать, что  $\Delta$  — это разбиение куба  $B^q$ . Разбиение — это объединение не пересекающихся множеств.

$$|\delta_i| = 2^m$$

$$\sum_{i=1}^{2^\lambda} |\delta_i| = 2^m \cdot 2^{q-m} = 2^q = |B^q|$$

$$\bigcup_{i=1}^{2^\lambda} \delta_i \supseteq B^q \rightarrow \Delta$$

$\Delta$  — искомое разбиение.

$$\forall \tau = (\hat{\tau}, \check{\tau}) \in B^q$$

$$\exists i \in [1, 2^\lambda] \quad \tau \in \delta_i$$

Сначала необходимо найти ту позицию внутри компоненты, на которой набор должен встретиться. Рассматриваются все наборы длины  $m$ . ■

## Асимптотически наилучшая верхняя оценка сложности для формул

**Утверждение 14.2.** Для любой ФАЛ  $f$ ,  $f \in P_2(n)$ , в  $U^\Phi$  существует реализующая ее формула  $F_f$ , для которой:

$$L(F_f) \leq \frac{2^n}{\log n} \left( 1 + \frac{2 \log \log n + O(1)}{\log n} \right)$$

$$D(F_f) \leq n - \log \log n + O(1)$$

Из этих оценок с учетом нижних оценок следствия из утверждения вытекает, что:

$$L^\Phi(n) \sim \frac{2^n}{\log n}$$

$$D(n) \leq n - \log \log n + O(1)$$

**Доказательство.**

Пусть  $m, s$  — натуральные параметры  $s \geq 2^m$ ,  $G$  — стандартное ДУМ порядка  $m$  и высоты  $s$ .

$$p = \left\lceil \frac{2m}{s} \right\rceil$$

$$\lambda = |G|p \cdot 2^s$$

$$\forall h(x') \quad \forall i \in [1, 2^\lambda] \quad h(x') \equiv \hat{h}(x_1, \dots, x_m) = g_1 \vee \dots \vee g_p \equiv x_{j_1}^{\tau_1} \vee \dots \vee x_{j_p}^{\tau_p} = J(x')$$

Получается элементарная дизъюнкция ранга  $p$ .

$$f(x_1, \dots, x_q, x_{q+1}, \dots, x_n) = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x') = \bigvee_{i=1}^{2^\lambda} X_{\delta_i}(x') \left( \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot J_{\sigma''_i}(x') \right)$$

Строится формула:

$$\tilde{F}_f = \bigvee_{i=1}^{2^\lambda} a_i(x) F_{n-q}(x'', J_{\delta_i}(x'), \dots, J_{\bar{1},i}(x'))$$

Происходит оптимизация ЭД:

$$J(x') = x_{j_1} \vee \dots \vee x_{j_i} \vee \bar{x}_{j_{i+1}} \vee \dots \vee \bar{x}_p = x_{j_1} \vee \dots \vee x_{j_i} \vee \overline{x_{j_1} \cdot \dots \cdot x_{j_p}} = \hat{J}$$

$\hat{F}_f$  получается из  $\tilde{F}_f$  оптимизацией ЭД  $J_{\sigma''_i}$  по числу отрицаний.

$$L_{\&, \vee}(F) \leq R(F_f) \leq 2^\lambda (2^m \cdot q + 2^{n-1+1} + p \cdot 2^{n-q})$$

$$L_{\perp}(F_f) \leq 2^{q-1} + 2^\lambda \cdot 2^{n-q} + 2^\lambda \cdot 2^{n-q}$$

Выбор параметров происходит следующим образом:

$$m = \lceil 3 \log \log n \rceil - 1$$

$$s = \lceil \log n - 2 \log \log n \rceil - 1$$

■

**Утверждение 14.3.** Для  $n = 1, 2, \dots$  выполняются неравенства:

$$L^\pi(\mu_n) \leq 2 \cdot 2^n + O\left(\frac{2^n}{n}\right)$$

$$L^C(\mu_n) \leq 2 \cdot 2^n + O\left(\frac{2^n}{n}\right)$$

$$D(\mu_n) \leq n + 6$$

причем существует такая реализующая ФАЛ  $\mu_n$  и неповторная по информации БП формула  $M_n$  с поднятыми отрицаниями, глубина которой удовлетворяет последнему из них, альтернирование не больше 3, а сложность не превосходит  $7 \cdot 2^n$ .

$F_f$  получается заменой  $F_{n-q}$  из утверждения для мультиплексорной функции на  $M_{n-q}$ . Следовательно, сложность этой формулы увеличивается.

### Асимптотически наилучшая верхняя оценка сложности для КС

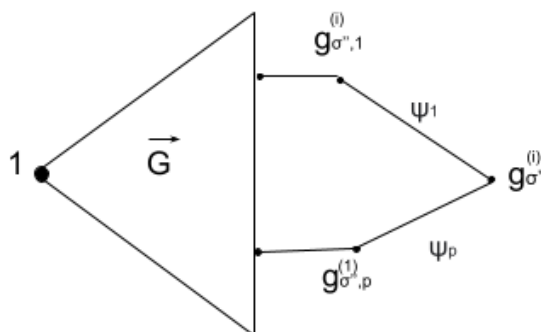


Рис. 14.5. ДУМ  $\vec{G}$

**Утверждение 14.4.** Для любой ФАЛ  $f$ ,  $f \in P_2(n)$ , существует реализующая ее КС  $\Sigma_f$  такая, что:

$$L(\Sigma_f) \leq \frac{2^n}{n} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$$

Из этой оценки с учетом нижней оценки следствия из утверждения вытекает, что:

$$L^K(n) \sim \frac{2^n}{n}$$

**Доказательство.**

$$f(x', x'') = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x') = \bigvee_{i=1}^{2^p} X_{\delta_i}(x') \cdot \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot \bigvee_{j=1}^p \psi_j g_{\sigma'', j}^{(i)}$$

$$\Psi = (\psi_1, \dots, \psi_p)$$

Только  $\Psi$  моделируется переменными.

Такое представление используется для синтеза контактных схем. ■

## Лекция 15. Асимптотически наилучший метод синтеза КС 1

**Утверждение 15.1.** Для любой ФАЛ  $f$ ,  $f \in P_2(n)$ , существует реализующая ее КС  $\Sigma_f$  такая, что:

$$L(\Sigma_f) \leq \frac{2^n}{n} \left( 1 + O\left(\frac{1}{\sqrt{n}}\right) \right)$$

Из этой оценки с учетом нижней оценки следствия из утверждения вытекает, что:

$$L^K(n) \sim \frac{2^n}{n}$$

**Доказательство.**

Рассматривается произвольная функция  $f$  и ее представления, разбив переменные этой функции на две группы:  $x'$  и  $x''$ :

$$f(x', x'') = \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x') = \bigvee_{i=1}^{2^p} X_{\delta_i}(x') \cdot \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot h_{i, \sigma''}(x')$$

При синтезе контактных схем будут моделироваться переменными не все функции этого ДУМ, а только небольшая их часть — характеристические множества. Только  $\psi$  моделируется переменными. Пусть  $m, s$  — натуральные параметры,  $s \leq 2^m$ ,  $G$  — стандартное ДУМ порядка  $m$  от БП  $(x_1, \dots, x_m)$  высоты  $s$ . Следовательно  $G$  — ДУМ ранга  $p$ ,  $p = \left\lfloor \frac{2^m}{s} \right\rfloor$ .

$$\lambda = |G| \leq p \cdot 2^s$$

Пусть  $q = m + p$  и  $\Delta = (\delta_1, \dots, \delta_{2^p})$  — разбиение куба  $B^q$  от  $x'$ , построенная для системы функций  $\psi = (\psi_1, \dots, \psi_p)$ .

Такое представление используется для синтеза контактных схем. ■



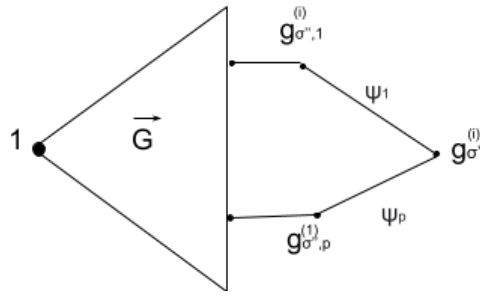


Рис. 15.1. ДУМ  $\vec{G}$

## Необходимые разложения для формулы

В этом случае, по этому разбиению все характеристические функции можно промоделировать переменными или их отрицаниями.

$$\forall h(x') \forall i = [1, 2^p] : h(x') \equiv_{\delta_i} \hat{h}(x_1, \dots, x_m) = \psi_1 g^{(1)} \vee \dots \vee \psi_p g^{(p)} \equiv x_{m+q}^{\tau_1} g^{(1)} \vee \dots \vee x_q^{\tau_p} g^{(p)} = \hat{h}_i$$

Таким образом можно построить разложение, на основе которого можно построить нужную формулу.

$$\begin{aligned} f(x', x'') &= \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot f_{\sigma''}(x') = \bigvee_{i=1}^{2^p} X_{\delta_i}(x') \cdot \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot \bar{h}_{i,\sigma''}(x') = \\ &= \bigvee_{i=1}^{2^p} X_{\delta_i}(x') \cdot \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot \left( x_{m+q}^{\tau_{i,\sigma''}^{(1)}} \cdot g_{i,\sigma''}^{(1)} \vee \dots \vee x_q^{\tau_{i,\sigma''}^{(p)}} \cdot g_{i,\sigma''}^{(p)} \right) \end{aligned}$$

## Построение КС по разложению и её сложность

Для каждого  $i$  строится схема  $\Sigma_i$ , которая состоит из 2 частей. Схема  $\Sigma_G$  реализует систему  $\vec{G}$  из функций  $G$ . Необходимо построить представление для любого  $i$  и  $\sigma''$ . Получится функция  $\hat{g}_{i,\sigma''}$ . Проводится контакт от выходов  $\Sigma_G$  к функции  $\hat{g}_{i,\sigma''}$ . Для всей дизъюнкции тратится  $p$  контактов. Эта схема обозначается как  $\Sigma'_i$ , которая реализует все функции при фиксированном  $i$ .

$$\Sigma' = \bigcup_{i=1}^{2^p} \Sigma'_i$$

Если работать внутри компоненты  $\delta_i$ , то функции и моделирующие их буквы ортогональны. Поэтому получается корректная суперпозиция на компоненте  $\delta_i$ .

Дальше необходимо умножить каждую такую функцию на ЭК от переменных группы  $x''$ , связанную с набором  $\sigma''$ . Для этого строится контактное дерево от переменных группы  $x''$ , где реализованы все конъюнкции. Вершина  $\hat{g}_{i,\sigma''}$  присоединяется к листу, где реализуется  $K_{\sigma''}(x'')$ . Вся эта функция должна быть умножена на характеристическую функцию  $i$  компоненты, которая зависит от переменной группы  $x'$ . Для этого строится следующее контактное дерево от переменных  $x'$ . Листья этого контактного дерева собираются в пучки и происходит проводное соединение с первым контактным деревом. Группы листьев не пересекаются, так как компоненты тоже не пересекаются. Эта часть схемы обозначается как  $\Sigma''$ .

В результате схема  $\Sigma_f$  получилась как результат корректной суперпозиции  $\Sigma''$  от  $\Sigma'$ .

$$\Sigma_f = \Sigma''(\Sigma')$$

Контактное дерево разделительно по листьям. Эта суперпозиция корректна только на компоненте  $\delta_i$ .

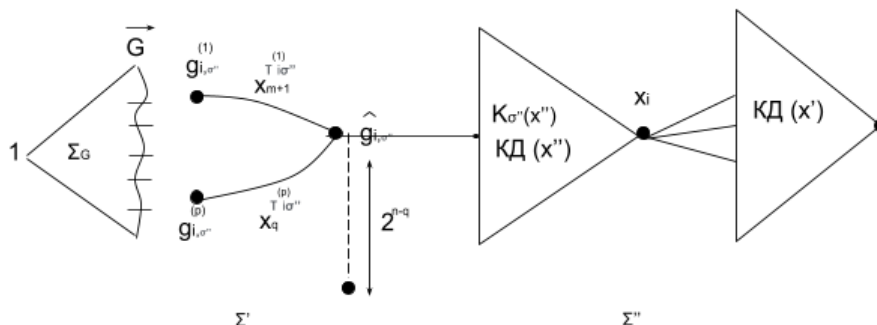


Рис. 15.2. Построение КС по разложению

Сложность этой схемы считается следующим образом:

$$L(\Sigma_G) \leq \lambda \cdot 2^{m+1}$$

$$L(\Sigma'_i) = L(\Sigma_G) + 2^{n-q} \cdot p$$

$$L(\Sigma') \leq 2^p L(\Sigma'_i) \leq \dots$$

$$L(\Sigma'') \leq 2^{q+1} + 2^p \cdot 2^{n-q+1}$$

Если просуммировать все сложности и подставить параметры, то можно получить общую оценку:

$$L(\Sigma_f) \leq (\lambda + 1)2^{q+1} + (p + 2) \cdot 2^{n-m}$$

Выбор параметров:

$$m = \left\lceil \frac{3}{2} \log n \right\rceil$$

$$s = \lfloor n - 2\sqrt{n} \rfloor$$

### Замечание о верхней оценке разбиения данной КС

Построенную КС  $\Sigma_f$  можно разбить на не более, чем:

$$\lambda p \cdot 2^p + 2^{n-m+1} + (\lambda + 1)2^{q+1} = O\left(\frac{2^n}{n\sqrt{n}}\right)$$

"звезд" каждая из которых состоит из контактов одного и того же типа.

Пусть  $\xi(\Sigma)$  — минимальное число связных однородных подсхем, на которые разбивается схема  $\Sigma$ . Однородная подсхема — это схема, состоящая только из контактов одного и того же типа.

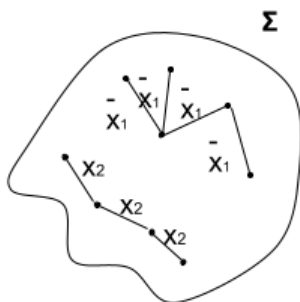


Рис. 15.3. "Звезда" контактов

По контактам они не пересекаются, а по вершинам подсхем могут пересекаться.

$$L(\Sigma_f) \sim \frac{2^n}{n}$$

Необходимо понять, можно ли построенную КС схему по разложениям разбить на однородные подсхемы. Чтобы получить хорошую оценку числа однородных звезд, надо центр звезд выбрать на выходах схемы  $\Sigma_G$ .

$$\xi(\Sigma) \leq L(\Sigma)$$

$$\xi(\Sigma_f) \leq \lambda \cdot p \cdot 2^p + \text{число всех остальных контактов}$$

В итоге получается:

$$\xi(\Sigma_f) \leq \lambda \cdot p \cdot 2^p + 2^{n-m+1} + (\lambda + 1)2^{q+1} = O\left(\frac{2^n}{n\sqrt{n}}\right)$$

Это означает, что в построенной КС по разложениям можно выделить такие однородные подсхемы, число которых не больше, чем константа.

## Задача синтеза схем для ФАЛ из специальных классов. Нижние оценки и их асимптотическая точность. Примеры

Специфика моделируется погружением функции в какой-то класс.

$$Q \subseteq P_2$$

Вводится последовательность:

$$Q_1, Q_2, \dots, Q_n = P_2(n) \cap Q$$

Вводится функция Шеннона в классе схем  $A$ :

$$L^A(Q(n)) = \max_{f \in Q(n)} L^A(f)$$

**Утверждение 15.2.** Для класса ФАЛ  $Q$  такого, что:

$$n = o\left(\frac{\log |Q(n)|}{\log \log |Q(n)|}\right)$$

$$\log n = o(\log \log |Q(n)|)$$

выполняются асимптотические неравенства:

$$L^C(Q(n)) \gtrsim \frac{\log |Q(n)|}{\log \log |Q(n)|}$$

$$L^K(Q(n)) \gtrsim \frac{\log |Q(n)|}{\log \log |Q(n)|}$$

$Q = \{f(x_1, \dots, x_n)\}$  симметричная по  $x_1$  и  $x_2$ .

$$|Q(n)| = 2^{\frac{3}{4}2^n}$$

$$L^C(Q(n)) \gtrsim \frac{3}{4} \frac{2^n}{n}$$

$$L^K(Q(n)) \gtrsim \frac{3}{4} \frac{2^n}{n}$$

Необходимо доказать, что эти оценки являются асимптотически точными. Для этого функция  $f$  разлагается по первым двум переменным:

$$\begin{aligned} f(x_1, \dots, x_n) &= \bar{x}_1 \bar{x}_2 \cdot f(0, 0, x_3, \dots, x_n) \vee \bar{x}_1 x_2 \cdot f(0, 1, x_3, \dots, x_n) \vee \\ &\vee x_1 \bar{x}_2 \cdot f(1, 0, x_3, \dots, x_n) \vee x_1 x_2 \cdot f(1, 1, x_3, \dots, x_n) = \\ &= \bar{x}_1 \bar{x}_2 g_0(x_3, \dots, x_n) \vee (\bar{x}_1 x_2 \vee x_1 \bar{x}_2) g_1(x_3, \dots, x_n) \vee x_1 x_2 g_2(x_3, \dots, x_n) \end{aligned}$$

В классе контактных схем можно построить схему  $\Sigma_f$ .

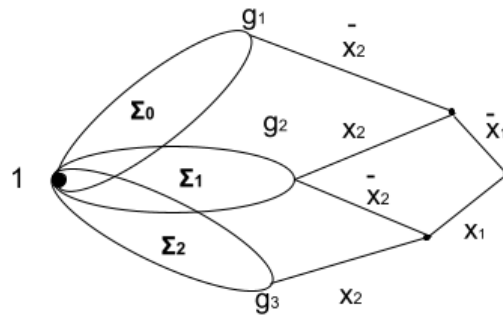


Рис. 15.4. Схема  $\Sigma_f$

$$L(\Sigma_f) \leq 6 + 3L^K(n-2)$$

С такой сложностью можно реализовать три любых функции а потом присоединить часть из 6 контактов.

$$L(\Sigma_f) \lesssim 6 + 3 \cdot \frac{2^{n-2}}{n-2} \lesssim \frac{3}{4} \frac{2^n}{n}$$

Была получена верхняя оценка, которая совпадает с нижней мощностной оценкой функции Шеннона.

$$Q = \{f(x_1, \dots, x_n) : f(1, x_2, \dots, x_n) \equiv 1\}$$

Если переменная  $x_2$  обратилась в 1, то и функция обратилась в 1. На нижней половине таблицы значений все функции равны 1. Произвольно функция задается только на первой половине столбца значений.

$$|Q(n)| = 2^{2^{n-1}}$$

Таким образом, нижняя оценка:

$$L^C(Q(n)) \gtrsim \frac{1}{2} \frac{2^n}{n}$$

Верхняя оценка получается следующим образом:

$$f(x_1, \dots, x_n) = x_1 \vee f(0, x_2, \dots) = x_1 \vee g(x_2, \dots, x_n)$$

$$L^C(Q(n)) \leq 1 + L^C(n-1) \lesssim \frac{1}{2} \frac{2^n}{n}$$

Верхняя оценка совпадает с нижней мощностной оценкой. И это доказывает, что:

$$L^C(Q(n)) \sim \frac{1}{2} \frac{2^n}{n}$$

### Синтез схем для дешифраторов, мультиплексоров и некоторых других ФАЛ, встречающихся в приложениях, оценки их сложности

**Утверждение 15.3.** Для  $n = 1, 2, 3, \dots$  выполняются неравенства:

$$L^K(\vec{Q}_n) \leq 2^n + O\left(\frac{2^n}{n}\right)$$

$$L^K(\vec{J}_n) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

$$Q_n = \{x_1^{\sigma_1} \dots x_n^{\sigma_n} : (\sigma_1, \dots, \sigma_n \in B^n)\}$$

$$J_n = \{x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n} : (\sigma_1, \dots, \sigma_n \in B^n)\}$$

$$|Q_n| = |J_n| = 2^n$$

$$\vec{Q}(n) = (\bar{x}_1, \dots, \bar{x}_n, \bar{x}_1, \dots, \bar{x}_{n-1}, x_n, \dots, x_1, \dots, x_n)$$

$$L^K(\vec{Q}(n)) = \min L(\Sigma)$$

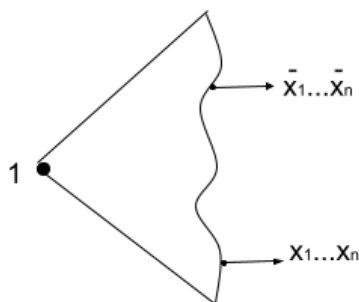


Рис. 15.5.  $\Sigma - (1, 2^n)$  КС

Пример такой съемы — это контактное дерево. Контактное дерево реализует систему из всех функций всех элементарных конъюнкций ранга  $m$  от  $n$  переменных. Но контактное дерево имеет сложность  $2^{n+1}$ . Это означает, что можно построить более экономную схему, чем контактное дерево.

В следствии получается, что оценки утверждений и следствия из утверждений дают асимптотические равенства:

$$L^K(\vec{Q}(n)) \sim 2^n$$

$$L^K(\vec{J}(n)) \sim 2^{n+1}$$

## Лекция 16. Самокорректирующаяся КС

Верхние оценки сложности для некоторых дешифраторов.

### Каскадная пи-схема

**Утверждение 16.1.** Для  $n = 1, 2, 3, \dots$  выполняются неравенства:

$$L^K(\vec{Q}_n) \leq 2^n + O\left(\frac{2^n}{n}\right)$$

$$L^K(\vec{J}_n) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

Это утверждение доказывает, что при синтезе всех элементарных конъюнкций ранга  $m$  от  $n$  переменных в классе контактных схем  $(1, 2^n)$  на одну конъюнкцию приходится 1 контакт. Всего конъюнкций  $2^n$ . Если реализовать систему всех элементарных дизъюнкций, то получаются 2 контакта на 1 реализуемую дизъюнкцию.

В следствии получается, что оценки утверждений и следствия из утверждений дают асимптотические равенства:

$$L^K(\vec{Q}(n)) \sim 2^n$$

$$L^K(\vec{J}(n)) \sim 2^{n+1}$$

**Доказательство.**

Набор переменных делится на две части:

$$x = (x_1, \dots, x_1, x_{q+1}, \dots, x_{n-q}) = (x', x'')$$

$$\sigma = (\sigma', \sigma'') \in B^n$$

Вводятся следующие параметры:

$$\lambda = 2^m$$

$$q = m + \lambda$$

$$g = (g_1, \dots, g_\lambda) = \vec{Q}_m(x_1, \dots, x_m)$$



Строится соответствующее разбиение  $B^q$ , построенная для системы  $g$ :

$$\Delta = (\delta_1, \dots, \delta_{2^q})$$

Каждая из этих функций на каждый из этих компонент либо переменной, либо ее отрицанием.

$$\forall \sigma' \in \delta_i : K_{\sigma'}(x') \equiv x_{j\sigma'}^{\tau_{\sigma'}}$$

Этот факт используется, чтобы реализовать систему всех конъюнкций ранга  $m$  от  $n$  переменных.

$$K_{\sigma}(x) = K_{\sigma'}(x') \cdot K_{\sigma''}(x'') = X_{\delta_i}(x') \cdot x_{j\sigma'}^{\tau_{\sigma'}} \cdot K_{\sigma''}(x'')$$

Это представление справедливо для любого набора  $\sigma'$  и  $\sigma''$ . На основе этого представления строится  $\Sigma_n^{(\&)}$ , которая будет реализовать каждую ЭК на базе такого представления.

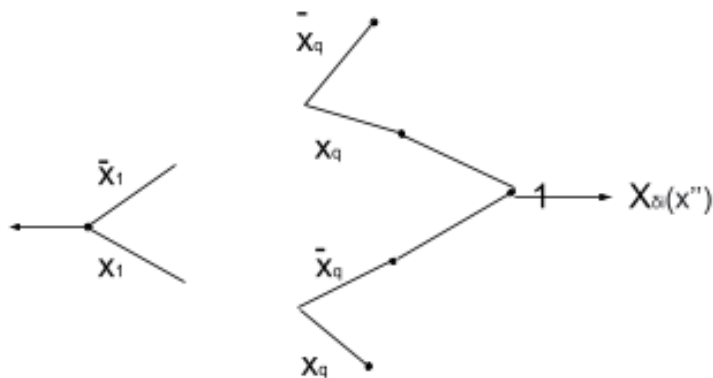


Рис. 16.1. Каскадная  $\pi$ -схема для совершенной ДНФ

Чтобы реализовать характеристическую функцию  $i$  компоненты, необходимо найти ЭК, которые входят в совершенную ДНФ  $a_i(x')$ . В совершенной ДНФ  $2^m$  ЭК. Эти ЭК склеиваются в один общий выход, где реализуется характеристическая функция  $i$  компоненты. Это будет каскадная пи-схема.

$$L \leq q \cdot 2^m$$

Сложность полученной схемы считается следующим образом:

$$L(\Sigma'_i) \leq q \cdot 2^m$$

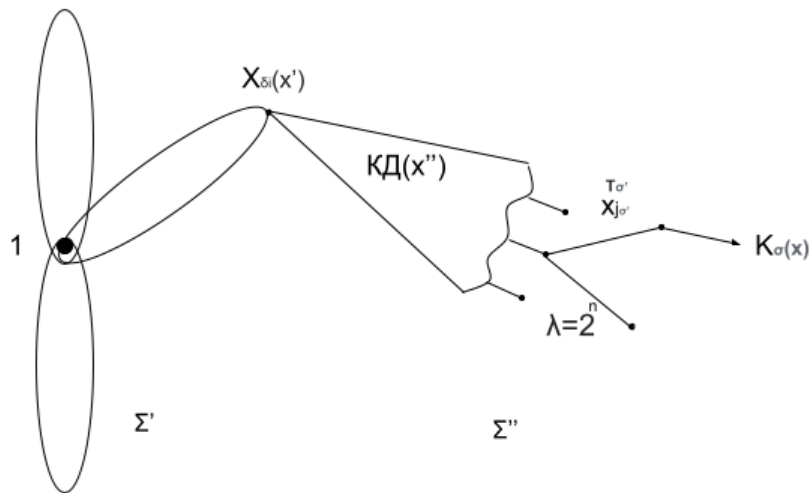


Рис. 16.2. Схема  $\Sigma_n^{(\&)}$

$$\Sigma' = \bigcup_{i=1}^{2^q} \Sigma'_i$$

$$L(\Sigma') \leq q \cdot 2^q$$

$$L(\Sigma''_i) \leq 2^{n-q+1} + 2^{n-q} \cdot 2^m \leq 2^{n-q+1} + 2^{n-\lambda}$$

$$L(\Sigma'') \leq 2^{n-m+1} + 2^n$$

Таким образом, общая сложность:

$$L(\Sigma_n^{(\&)}) \leq q \cdot 2^q + 2^{n-m+1} + 2^n$$

Выбор параметров:

$$m = \lceil \log n \rceil \rightarrow L(\Sigma_m^{(\&)}) \leq 2^n + O\left(\frac{2^n}{n}\right)$$

Схема  $\Sigma_n^{(\&)}$  является каскадной контактной схемой. Было утверждение, что с каскадной контактной схемой можно перейти в инверсную схему, реализующая отрицание функций.

$$\Sigma_n^{(\&)} \mapsto \Sigma_n^{(\vee)}$$

$$\vec{J}(n) = |\vec{Q}(n)|$$

$$L(\Sigma_n^{(\vee)}) \leq 2L(\Sigma_n^{(\&)}) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

Таким образом, была получена вторая оценка. ■

## Верхние оценки сложности для некоторых мультиплексоров

**Утверждение 16.2.** Для  $n = 1, 2, \dots$  выполняются неравенства:

$$L^\pi(\mu_n) \leq 2 \cdot 2^n + O\left(\frac{2^n}{n}\right)$$

$$L^C(\mu_n) \leq 2 \cdot 2^n + O\left(\frac{2^n}{n}\right)$$

$$D(\mu_n) \leq n + 6$$

причем существует такая реализующая ФАЛ  $\mu_n$  и бесповторная по информации БП формула  $M_n$  с поднятыми отрицаниями, глубина которой удовлетворяет последнему из них, альтернирование не больше 3, а сложность не превосходит  $7 \cdot 2^n$ .

**Доказательство.**

$$\mu(n) = (x_1, \dots, x_n; y_0, y_1, \dots, y_{2^n-1}) = \bigvee_{\sigma \in B^n} K_\sigma(x) \cdot y_{V(\sigma)}$$

Чтобы построить схему, необходимо каждый выход дешифратора соединить с контактом с новым выходом схемы. Получится  $\pi$ -схема  $\Sigma_n^{(\mu)}$ .

$$L(\Sigma_n^{(\mu)}) \leq 2^n + L(\Sigma_n^{(\&)}) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

Следовательно, этим доказывается, что:

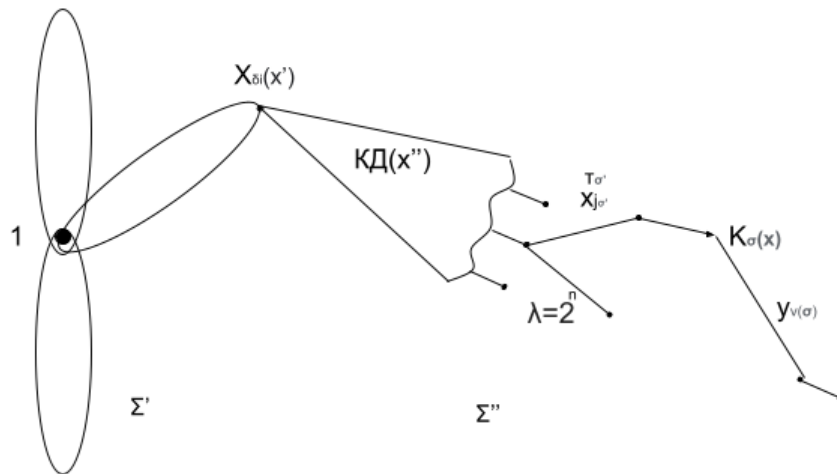


Рис. 16.3.  $\pi$  - схема  $\Sigma_n^{(\mu)}$

$$L^\pi(\mu_n) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

■

**Доказательство.**

От этой  $\pi$  - схемы можно перейти к моделирующей ее формуле.  $\tilde{F}_n$  — формула с поднятыми отрицаниями, которая моделирует  $\pi$  - схему  $\Sigma_n^{(\mu)}$ .

$$L_{\&,\vee}(\tilde{F}_n) \leq L(\Sigma_n^{(\mu)})$$

Из формулы  $\tilde{F}_n$  необходимо перейти в СФЭ  $S_n$  в результате склеивания отрицаний, связанных с переменной  $x_i$ .

$$L_1(S_n) \leq n$$

$$L(S_n) \leq 2^{n+1} + O\left(\frac{2^n}{n}\right)$$

■

Необходимо построить реализующую формулу  $M_n$  с поднятыми отрицаниями, у которой глубина не больше, чем  $n + 6$ , альтернирование не больше 3, а

сложность не превосходит  $7 \cdot 2^n$ . Записывается представление для мультиплекторной функции:

$$\mu(x, y) = \bigvee_{i=1}^{2^\lambda} X_{\delta_i}(x') \cdot \left( \bigvee_{\sigma'' \in B^{n-q}} K_{\sigma''}(x'') \cdot \bigvee_{\sigma' \in \delta_i} x_{j_{\sigma'}}^{\tau_{\sigma'}} \cdot y_{\nu(\sigma', \sigma'')} \right)$$

Если построить формулу по этому представлению, то получится большое альтернирование. Поэтому происходит переход в другую формулу, где внешняя дизъюнкция заменяется внешней конъюнкцией:

$$\tilde{F}_n(x, y) = \bigvee_{i=1}^{2^\lambda} X_{\delta_i}(x') \cdot \&_{\sigma'' \in B^{n-q}} \left( J_{\sigma''}(x'') \vee \bigvee_{\sigma' \in \delta_i} x_{j_{\sigma'}}^{\tau_{\sigma'}} \cdot y_{\nu(\sigma', \sigma'')} \right)$$

$$Alt(\tilde{F}_n) = 3$$

$$n \geq 6 \quad m = \left\lceil \log \left( \frac{n}{3} \right) \right\rceil$$

$$L(\tilde{F}_n) \leq 7 \cdot 2^n - 1$$

Формула  $M_n$  получается из  $\tilde{F}_n$  оптимизацией по глубине. Она имеет такую же сложность, но глубина имеет следующий вид:

$$D(M_n) \leq \lceil \log(7 \cdot 2^n) \rceil + 3 \leq n + 6$$

## Самокорректирующаяся КС. Обрыв и замыкание контакта

**Утверждение 16.3.** Для монотонной симметрической ФАЛ  $s_n^{2,n}$  с порогом 2 при  $n \geq 2$  выполняются неравенства:

$$2n - 3 \leq L^C \left( s_n^{[2,n]} \right) \leq 2n + O(\sqrt{n})$$

Вследствие этого получается:

$$L^C \left( s_n^{[2,n]} \right) \sim 2n$$

Есть методы, которые позволяют, не слишком сильно увеличивая сложность схемы, гарантировать ее работоспособность при наличии нескольких неисправностей. Такие схемы называются самокорректирующимися схемами. Пусть  $\Sigma \in U^K$ .  $\Sigma$  — это контактная схема, корректирующая  $p$  обрывов и  $q$  замыканий, или  $\Sigma$  — это  $(p, q)$  самокорректирующаяся контактная схема.  $\Sigma$  является самокорректирующейся схемой тогда и только тогда, когда любая контактная схема  $\Sigma'$ , полученная из схемы  $\Sigma$  обрывом любых, но не более, чем  $p$  контактов и замыканием любых, но не более, чем  $q$  контактов, эквивалентна схеме  $\Sigma$ .

Если контакт обрывается, то проводимость исчезает, а вершины остаются. При замыкании он становится тождественно проводящим проводником.

### Тривиальное решение задачи повышения надёжности схемы

Происходит продублирование контакта параллельно в  $p + 1$  раз. Если обрывать не более, чем  $p$  контактов, то функция проводимости между входом и выходом останется прежней. Эта схема является  $(p, 0)$  самокорректирующейся схемой.

Чтобы сделать схему  $(0, q)$ , контакт продублируется последовательно  $q + 1$ . Если замкнуть любые  $q$ , то функция проводимости останется прежней.

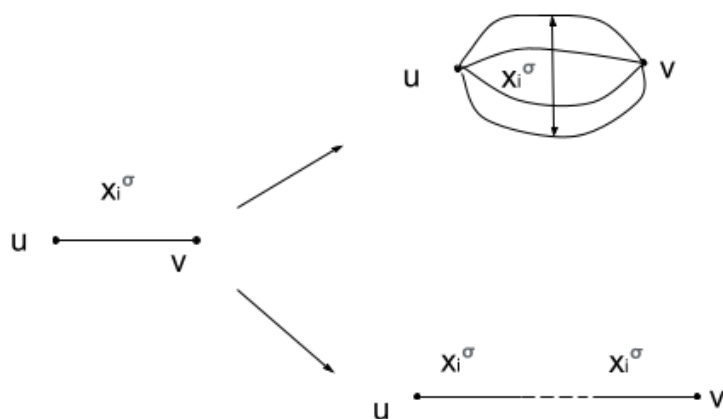


Рис. 16.4. Схема  $(p, 0)$  и  $(0, q)$

$\forall \rightarrow \Sigma^{(p,q)}$  заменой каждого контакта схемой  $(p, q)$ .

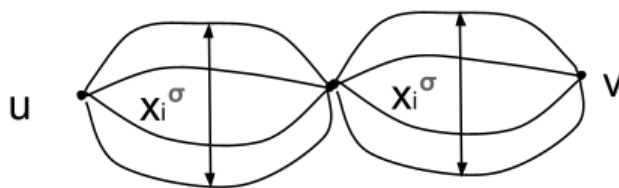


Рис. 16.5. Схема  $(p, q)$

**Утверждение 16.4.** Для любых  $p \geq 0, q \geq 0$  и любой КС  $\Sigma$  существует эквивалентная ей КС  $\Sigma', \Sigma' \in U_{(p,q)}^K$ , для которой:

$$L(\Sigma') \leq (p+1)(q+1)L(\Sigma)$$

### Однородная связная подсхема. Пример. Коррекция с помощью разбиения

Построенную КС  $\Sigma_f$  можно разбить на не более, чем на:

$$\lambda p \cdot 2^p + 2^{n-m+1} + (\lambda + 1)2^{q+1} = O\left(\frac{2^n}{n\sqrt{n}}\right)$$

"звезд каждая из которых состоит из контактов одного и того же типа.

**Утверждение 16.5.** Для любой КС  $\Sigma$  существуют эквивалентные ей  $(1,0)$  и  $(0,1)$  - самокорректирующиеся КС  $\Sigma'$  и  $\Sigma''$  соответственно такие, что:

$$L(\Sigma') \leq L(\Sigma) + \xi(\Sigma)$$

$$L(\Sigma'') \leq L(\Sigma) + \xi(\Sigma)$$

$\Sigma, \Sigma' \subseteq \Sigma$  — однородная связная подсхема схемы  $\Sigma$ . Однородная означает, что состоит из контактов одного типа. Связная означает, что эти контакты образуют связный граф. Рассматривается функция голосования:

$$x_1x_2 \vee x_1x_3 \vee x_2x_3$$

$\xi(\Sigma)$  — минимальное число однородных подсхем, на которые разбивается схема  $\Sigma$ . Эта функция может быть разбита на 4 подсхемы. Эти подсхемы не пересекаются по контактам, но пересекаются по полюсам.

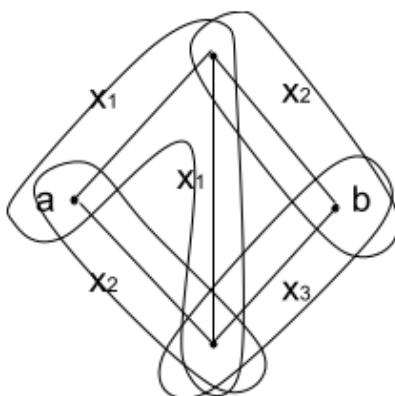


Рис. 16.6. Разбиение функции голосования

На основе такого разбиения можно сделать самокоррекцию более эффективной. Но для этого сначала надо построить самокорректирующиеся однородные подсхемы.

$$\Sigma = \Sigma^{(1)} \cup \Sigma^{(2)} \cup \dots \cup \Sigma^{(r)} \rightarrow \hat{\Sigma} \quad r = \xi(\Sigma)$$

$\hat{\Sigma}$  эквивалентна  $\Sigma$  и корректирует 1 обрыв или 1 замыкание заменой:

$$\hat{\Sigma} = \hat{\Sigma}^{(1)} \cup \dots \cup \hat{\Sigma}^{(r)}$$

Соответственно, увеличится сложность этой схемы.

$$L(\hat{\Sigma}) \leq L(\Sigma) + r$$

Эта схема корректирует либо 1 обрыв, либо 1 замыкание.

## Асимптотика функции Шеннона для самокорректирующихся КС

**Утверждение 16.6.** Для  $n = 1, 2, \dots$  имеют место следующие асимптотические равенства:

$$L_{(1,0)}^K(n) \sim L_{(0,1)}^K(n) \sim \frac{2^n}{n}$$

$U_{(p,q)}^K$  — это класс КС, корректирующих  $p$  обрывов и  $q$  замыканий.

$$L_{(p,q)}^K(f) = \min L(\Sigma)$$

$$L_{(p,q)}^K(n) = \max L_{(p,q)}^K(f)$$



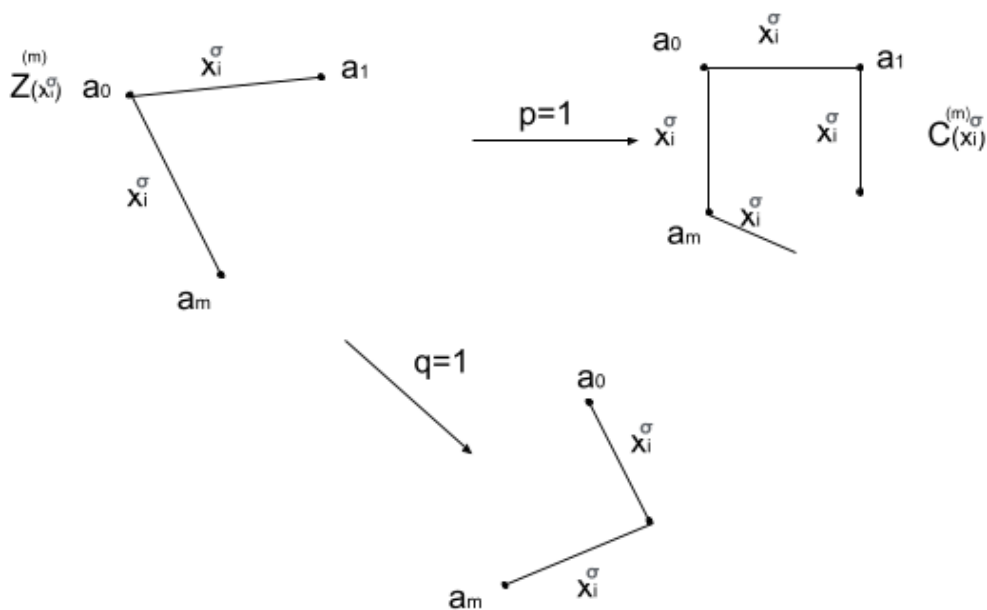


Рис. 16.7. Коррекция обрыва или замыкания

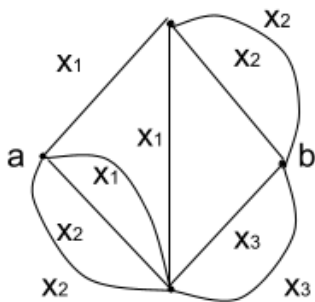


Рис. 16.8. Коррекция 1 обрыва

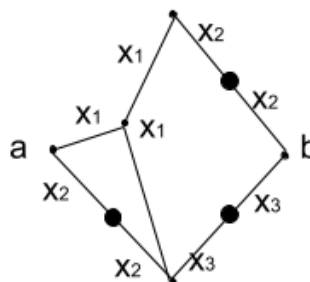


Рис. 16.9. Коррекция 1 замыкания

### Случай схемы Кардо

**Утверждение 16.7.** Для  $n = 1, 2, \dots$  имеют место равенства:

$$L_{(1,0)}^K(l_n) = L_{(1,0)}^K(\bar{l}_n) = 4n$$

$f = x_1 \oplus \dots \oplus x_n$  — схема Кардо.

Из этой схемы можно сделать схему  $\hat{\Sigma}_f --- (1,0)$ , корректирующую 1 обрыв, проведя 4 контакта.

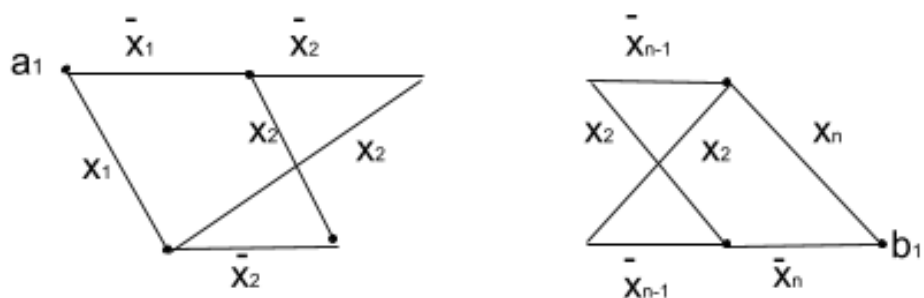


Рис. 16.10. Схема Кардо

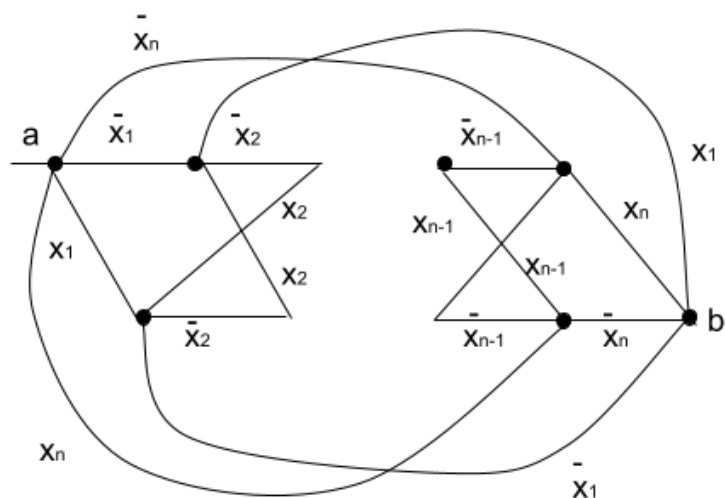


Рис. 16.11. Схема Рабиновича

$$L(\hat{\Sigma}_f) = 4n$$

## Лекция 17. Задача контроля схем

### Постановка задачи контроля схем. Область определённости

Есть ненадежная схема. Необходимо эту модель ненадежных схем сформулировать. Пусть есть схема  $\Sigma$ , которая реализует функцию  $f = f_1(x_1, \dots, x_n)$  с областью определенности:  $A = \{\alpha_1, \dots, \alpha_p\} \subseteq B^n$

Пусть есть источник неисправностей  $I$ .

$$\Sigma \xrightarrow{I} \{\Sigma_1, \Sigma_2, \dots, \Sigma_s\}$$

$\Sigma_1$  — исправное состояние схемы, а  $\Sigma_2, \Sigma_s$  — неисправные состояния схемы.  $(\Sigma, n)$  — ненадежная схема. В неисправном состоянии реализуется функция  $f_2, \dots, f_s$ . Необходимо найти неисправность.

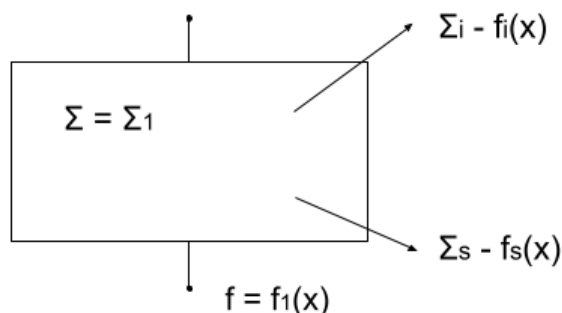


Рис. 17.1. Исправное и неисправное состояние схемы

### Таблица контроля. Цель контроля: проверка, диагностика

Необходимо знать таблицу значений этих функций. Должна быть дана матрица  $M \in B^{(p,s)}$ , где  $p$  — это мощность области определения. По строкам идут наборы, которые включены в область определения, а по столбцам идут функции. Эта матрица называется таблицей контроля.

	$f_1$		$f_i$		$f_s$
$a_1$					
$a_j$			$f_{ij}(x)a$		
$a_p$					

Рис. 17.2. Таблица контроля

С помощью таблицы значений можно найти исправные и неисправные схемы. Кроме таблицы контроля также надо сформулировать цель контроля. Целью контроля может быть проверка правильности (отличить исправную схему от любой неисправной схемы). Вторая цель — это диагностика схемы (в каком неисправном состоянии находится схема). При диагностике сравниваются выходные значения. Чтобы отличить два неисправных состояния, необходимо знать, что любые два столбца в матрице различные. Предполагается, что матрица  $M$  отделима по столбцам. Если два столбца одинаковы, то их надо склеить в один и считать, что этот склеенный столбец соответствует неисправности объединенной.

При цели контроля необходимо указать перечень тех пар столбцов, которые надо отличить.

$$N = \{(r, q) : r < q \in [1, s]\}$$

Цель контроля — это проверка правильности.

$$N = \{(1, 2), (1, 3), \dots, (1, s)\}$$

Это перечень тех пар столбцов, которые надо отличить друг от друга, если целью контроля является проверка правильности работы схемы.

При диагностике необходимо включать все пары.

$$\{(1, 2), \dots, (1, s), (2, 3), \dots, (2, s), \dots, (s-1, s)\}$$

## Тест для таблицы и тест для схемы. Тупиковый, минимальный, проверяющий и диагностический тесты

Решением этой задачи является такое множество строк, которое позволило бы отличить любую пару неисправности, входящую в цель контроля. Таким образом, строится тест:

$$T \subseteq [1, p] : \forall (r, q) \in N \exists i \in T : M \langle i, r \rangle \neq M \langle i, q \rangle$$

Это условие верно тогда и только тогда, когда  $T$  — это тест для пары  $(M, N)$ . Когда говорится про тест для схемы, то вместо номеров строк выступают соответствующие им наборы. Есть понятие тупикового теста и минимального теста.  $T$  — это тупиковый тест тогда и только тогда, когда не существует подмножество  $T' \subseteq T$ , которое тоже является тестом.  $T$  — это тест минимальной длины тогда и только тогда, когда для любого  $T'$  теста мощность множества  $T'$  имеет следующий вид:

$$|T'| \geq |T|$$

Среди тупиковых тестов есть также и минимальные тесты. Проверяющий тест — это тест, являющийся для проверки правильности работы схемы. Диагностический тест — это тест, целью которого является диагностика.

### Пример задачи. Способы построения тестов

**Задача:** построить все тупиковые тесты для этой пары по данной таблице контроля и по цели контроля.

**Пример 17.1.** Пусть есть исправная функция:

$$f = f_1(x_1, x_2, x_3) = (x_1 \oplus x_2)\bar{x}_3$$

*Первая неисправная функция инвертированием переменных:*

$$f_2 = x_1 \oplus \bar{x}_2 \cdot \bar{x}_3 \quad (x_2 \leftrightarrow \bar{x}_2)$$

*Вторая неисправная функция перепутыванием входов:*

$$f_3 = x_3 \oplus x_2\bar{x}_1 \quad (x_1 \leftrightarrow x_3)$$

Третья неисправная функция забиванием входов константами:

$$f_4 = x_3 \quad (x_1 := 1, \quad x_2 := 1)$$

Составляется таблица контроля для этого случая.

$x_1$ $x_2$ $x_3$	$f_1$	$f_2$	$f_3$	$f_4$
000	0	1	0	0
001	0	0	1	1
010	1	0	1	0
011	0	0	0	1
100	1	0	0	0
101	1	1	1	1
110	0	1	0	0
111	1	1	1	1

Рис. 17.3. Таблица контроля

Сформулируется матрица  $M$ .

$$p = 6, \quad s = 4 \quad M = \begin{array}{|l} 0100 \\ 0011 \\ 1010 \\ 0001 \\ 1000 \\ 0100 \end{array}$$

101 и 111 не могут дать различия между столбцами, поэтому их необходимо убрать. Первая и последняя строки совпадают, поэтому одну из них временно можно убрать. Тогда  $p = 5$ . Далее необходимо найти диагностический тест для этой матрицы. Если в диагностический тест будут входить вторая и третья строчки, то можно отличить любые два столбца. Пример минимального проверяющего теста состоит из 5 строки.

Вводится функция покрытия.

**Определение 17.1.** Множество строк  $T$ ,  $T \in [1, s]$  — тест для  $(M, N)$  тогда и только тогда, когда любой пары  $(r, q) \in N$  существует  $i \in T_i$ , для которого:

$$M \langle i, r \rangle \neq M \langle i, q \rangle$$

Строятся все тупиковые тесты, среди которых необходимо найти минимальные тесты. Для каждого  $i \in [1, p]$  сопоставляется БП  $y_i$  ( $y_i = 1$  тогда и только тогда, когда  $i$  строка включается в  $T$ ). В этом случае, можно определить функцию теста:

$$F(y_1, \dots, y_p) \leftrightarrow \forall \beta = (\beta_1, \dots, \beta_p)$$

$$F(\beta) = 1 \leftrightarrow I(\beta) = \{i : \beta_i = 1\}$$

Задача поиска всех тупиковых тестов сводится к задаче построения всех тупиковых покрытий.

$$|N| = S$$

Пусть  $(r, q) \in N$  — пара с номером  $t$  из множества  $S$ .

$$t = 1, \dots, S$$

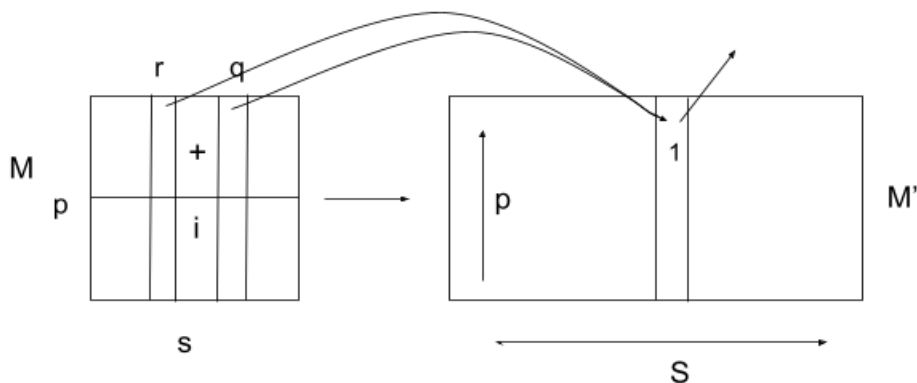


Рис. 17.4. Сопоставление матрице  $M$

$$M' \langle i, t \rangle = 1$$

$T$  является тестом для матрицы  $M$  тогда и только тогда, когда это множество строк является покрытием для матрицы  $M'$ .

## Утверждение о задании ФАЛ теста с помощью КНФ и его следствие. Продолжение примера

**Утверждение 17.1.** Для любой отделимой по столбцам матрицы  $M \in B^{p,s}$  и целью контроля  $N$  функция теста:

$$F(y_1, \dots, y_s) = \bigwedge_{(r,q) \in N} \left( \bigvee_{i \in [1,p]} y_i \right)$$

От этой функции покрытия можно получить сокращенную ДНФ. Простая импликанта функции покрытия записывается следующим образом:

$$\bigvee y_{i_1}, \dots, y_{i_m} \bigvee$$

Эта импликанта взаимно однозначно соответствует тупиковому тесту  $T = (y_1, \dots, y_m)$ .

Строится матрица  $M'$ , которая соответствует попарным суммам столбцов.

$$M = \begin{array}{|c|} \hline 0100 \\ \hline 0011 \\ \hline 1010 \\ \hline 0001 \\ \hline 1000 \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 100110 \\ \hline 011110 \\ \hline 101101 \\ \hline 001011 \\ \hline 111000 \\ \hline \end{array} = M'$$

Цель контроля — это диагностика схемы. Для матрицы  $M'$  необходимо написать функцию покрытия.

$$F(y_1, \dots, y_5) = (y_1 \vee y_3 \vee y_5) \cdot (y_2 \vee y_5) \cdot (y_2 \vee y_3 \vee y_4 \vee y_5) \cdot (y_1 \vee y_2 \vee y_3) \cdot (y_1 \vee y_2 \vee y_4) \cdot (y_3 \vee y_4)$$

Таким образом, получилась КНФ функции покрытия. Далее можно построить все тупиковые диагностические тесты для исходной задачи.

$$F(y_1, \dots, y_5) = (y_5 \vee y_2 y_1 \vee y_2 y_3) \dots = y_2 y_3 \vee y_1 y_2 y_4 \vee y_1 y_3 y_5 \vee y_1 y_4 y_5 \vee y_2 y_4 y_5 \vee y_3 y_4 y_5$$

Это означает, что вторая и третья строчки — тупиковый диагностический тест для матрицы  $M$ . Вместо этой пары необходимо написать соответствующие наборы значения переменных.

$$\{2, 3\} \rightarrow \{(001), (010)\}$$



Эти два набора — это тупиковый диагностический тест для исходной схемы  $\Sigma$ . Если поменять  $y_5$  на  $y_6$ , то тестов будет больше. Если провести все замены, то получится множество всех тупиковых тестов для схемы.

## Оценки длин тестов

**Утверждение 17.2.** *Любой тупиковый диагностический тест  $T$  для отдельной по столбцам матрицы  $M \in B^{p,s}$  имеет:*

$$[\log s] < T < (s - 10)$$

Когда все столбцы разные, то выполняется неравенство:

$$2^t \geq s \rightarrow t \geq [\log s]$$

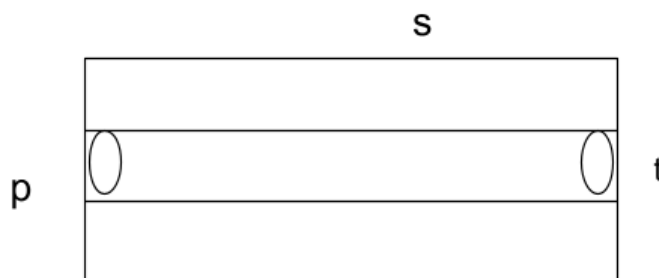


Рис. 17.5. Матрица  $M'$

$\forall p \geq q \geq 1$  вводится отношение эквивалентности  $\theta_q$  на множестве столбцов матрицы  $M$ :

$$i \sim_q j$$

тогда и только тогда, когда эти столбцы совпадают в первых  $q$  строках. Из этого определения ясно, что следующая эквивалентность  $q + 1$  сильнее, чем эквивалентность с номером  $q$ .  $|\theta_q|$  — число класса эквивалентности по  $\tilde{q}$ . Число класса эквивалентности будет постепенно расти, если увеличивать строку.

$$|\theta_0| = 1 < |\theta_1| < \dots < |\theta_r| = |\theta_{r+1}| = s$$

Дальше происходит стабилизация.

$$r \leq s - 1$$

Таким образом, была найдена верхняя оценка длины диагностического тупикового теста.

**Утверждение 17.3.** Пусть есть функция  $\varphi(s)$ ,  $t(s)$ ,  $p(s)$  и при этом верно:

$$t(s) \geq 2 \log s + \varphi(s)$$

$$\varphi(s) \geq 0 \quad \varphi(s) \xrightarrow{s \rightarrow \infty} \infty$$

Доля тех матриц  $M$ ,  $M \in B^{p,s}$ , для которых первые  $t$  строк образуют диагностический тест, который стремится к 1 при  $s \rightarrow \infty$ .

$$\frac{2^t (2^{t-1}) (2^t - s + 1) \cdot 2^{(p-t) \cdot s}}{2^{ps}} \rightarrow 1 \quad s \rightarrow \infty$$



Рис. 17.6. Диагностический тест

## Лекция 18. Эквивалентные преобразования КС

### КС с неразделёнными полюсами

Разделенные полюса — это когда в схеме есть группа входов и группа выходов. Пусть  $(p, q)$  — это КС  $\Sigma(x_1, \dots, x_n; a_1, \dots, a_p; b_1, \dots, b_q)$ , которая реализует матрицу:

$$F = \left\| f_{ij} \right\|_{\substack{1 \leq j \leq q \\ 1 \leq i \leq p}}$$

где  $f_{ij}$  — ФАЛ проводимости от  $a_i$  к  $b_j$ .

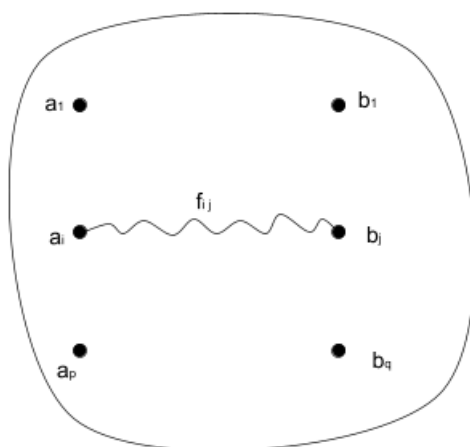


Рис. 18.1.  $(p, q)$  — КС

В этой модели не выполняется принцип эквивалентной замены. Если число входов равно числу выходов:

$$p = q, \quad a_1 = b_1, \quad a_p = b_q$$

то  $\Sigma(x_1, \dots, x_n; a_1, \dots, a_p)$  — КС с  $p$  неразделенными полюсами. Тогда реализуется квадратная матрица от функции проводимости  $f_{ij}$  от  $a_i$  к  $a_j$ .

$$F_\Sigma = F = \left\| f_{ij} \right\|_{1 \leq i, j \leq p}$$

Необходимо найти специфику матрицы, которая реализуется в схеме с неразделенными полюсами.

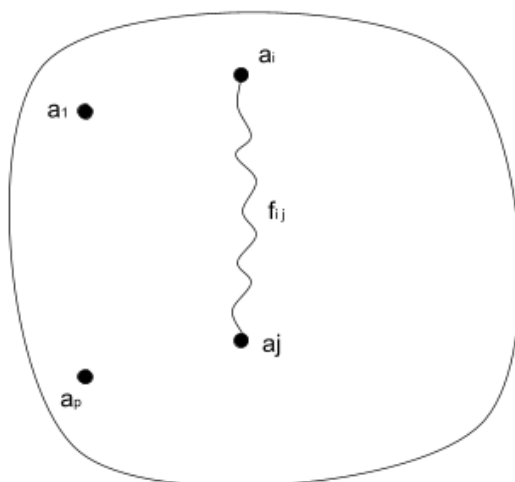


Рис. 18.2. КС с неразделенными полюсами

1)  $F$  — рефлексивная матрица.

$$f_{i,i} \equiv 1$$

2)  $F$  — транзитивная матрица.

$$\forall i, j, l: f_{ij} \geq f_{il} \cdot f_{lj}$$

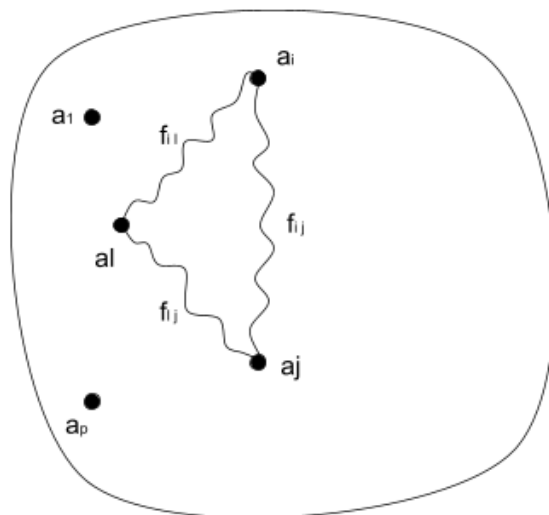


Рис. 18.3. Транзитивность матрицы

3) При неориентированных КС  $F$  — симметричная матрица.

$$f_{ij} = f_{ji}$$

## Тождество КС. Примеры

$$t : \Sigma' \sim \Sigma''$$

Это тождество верно тогда и только тогда, когда матрица  $F_{\Sigma'}$  совпадает с матрицей  $F_{\Sigma''}$ . Предполагается, что у этих схем одинаковый набор полюсов.

Если в схеме нет полюсов, то считается что схема реализует пустую матрицу независимо от того сколько там контактов. Тривиальная пустая схема  $t_1$  — это одна вершина, которая не является полюсом. Эта схема эквивалентна пустой схеме. Тождество коммутативности при конъюнкции для  $t_2$  записывается следующим образом:

$$t_{\&}^K : x_1 x_2 = x_2 x_1$$



Рис. 18.4. Тождество  $t_2$

Матрица, реализуемая тождеством  $t_2$ :

$$F_{\Sigma'} = F_{\Sigma''} = \begin{bmatrix} 1 & x_1 x_2 \\ x_1 x_2 & 1 \end{bmatrix}$$

При тождестве  $t_3$  если есть 2 противоположных контакта, соединенные к полюсам, а внутренняя вершина этой цепи не является полюсной вершиной, то эту цепь можно убрать и получить пару изолированных полюсов.

$$F_{\Sigma'} = F_{\Sigma''} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\bar{x}_1 \cdot x_1 = 0$$

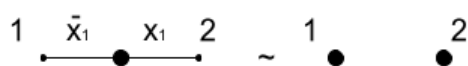


Рис. 18.5. Тожество  $t_3$

## Подстановка (модификация) тождества КС. Виды модификаций. Примеры

$\hat{t}$  — это подстановка (модификация) тождества  $t$  тогда и только тогда, когда  $\hat{t}$  получается из тождества  $t$ :

- переименованием управляющих БП с возможным отождествлением и инвертированием в каждой из частей  $t$ .
- переименованием с возможным отождествлением и снятием полюсов.

Рассматривается тождество  $t_4$ .

$$t_4 : x_2 = x_2 \bar{x}_1 \vee x_2 x_1$$

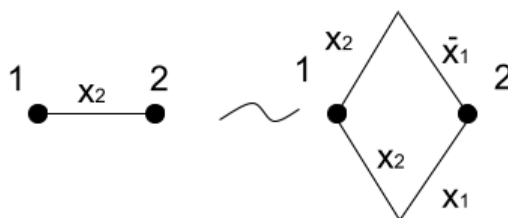


Рис. 18.6. Тожество  $t_4$

Происходит модификация тождества  $t_4$  в  $\hat{t}_4$ .

Рассматривается трехполюсное тождество  $t_5$ . Функция проводимости между любыми двумя различными полюсами равна  $x_1$ .

$$F = \begin{bmatrix} 1 & x_1 & x_1 \\ x_1 & 1 & x_1 \\ x_1 & x_1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1 \\ x_1 & 1 & x_1 \\ x_1 & x_1 & 1 \end{bmatrix}$$

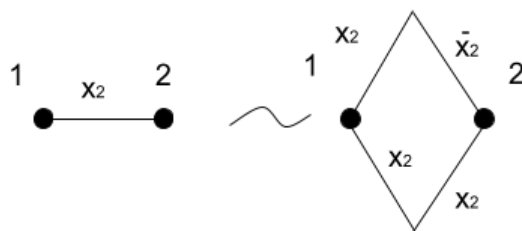


Рис. 18.7. Модификация  $\hat{t}_4$

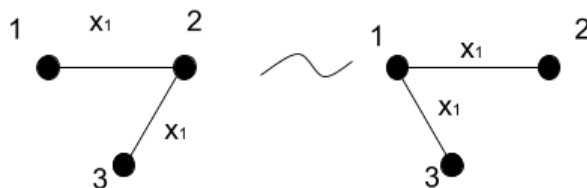


Рис. 18.8. Тожество  $t_5$

При модификации происходит отождествление полюсов 1 и 3 и снятие пометок.



Рис. 18.9. Модификация  $\hat{t}_5$

## Понятие подсхемы, выполняющее принцип эквивалентной замены

$\Sigma' \subseteq \Sigma$  тогда и только тогда, когда:

$$V(\Sigma') \subseteq V(\Sigma)$$

$$E(\Sigma') \subseteq E(\Sigma)$$

Необходимо определить полюса схемы.  $v \in V(\Sigma')$  — это полюс подсхемы  $\Sigma'$ , если:

- $v$  — полюс исходной схемы  $\Sigma$
- $v$  является концевой вершиной контакта, который принадлежит  $V(\Sigma) \setminus V(\Sigma')$ .

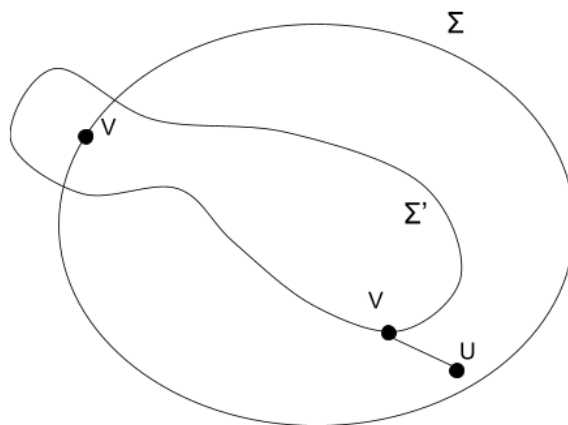


Рис. 18.10. Определение вершины подсхемы

Выполняется принцип эквивалентной замены.

$$\Sigma \xrightarrow{\hat{i}} \check{\Sigma} \rightarrow \Sigma \sim \check{\Sigma}$$

$$\hat{i} : \hat{\Sigma}' \sim \hat{\Sigma}''$$

## Примеры эквивалентных преобразований КС

Рассматривается отождествление переменных у конъюнкции. К левой части схемы применяется модификация  $\hat{t}_4$  (применяется перемена переменных  $(x_2 \rightarrow x_1)$ ). В правой части схемы применяется тождество  $t_3$ .

Рассматривается однополюсное тождество  $t_6^m$ , которое реализует матрицу размеров 1 на 1, состоящая тождественно из 1. На полюсе 1 висит цикл, который можно сбросить и получить изолированный полюс 1.

В модификации  $\hat{t}_5$  можно применить тождество  $t_6^m$  и сбросить висячий цикл  $x_1$ . Таким образом, 2 параллельных контакты преобразовываются в один контакт. Отсюда выводится тождество  $t_7$ .



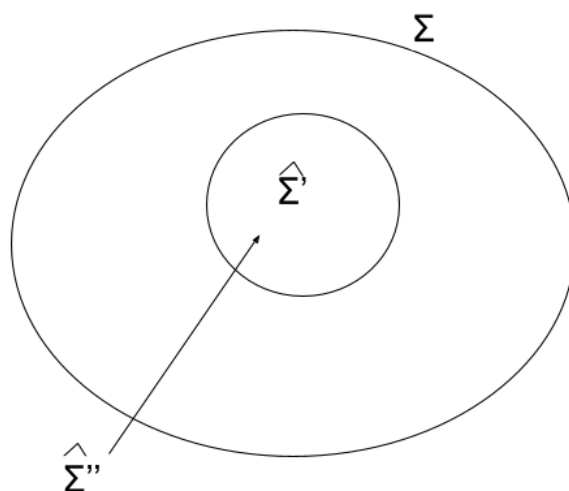


Рис. 18.11.  $\Sigma \rightarrow \check{\Sigma}$

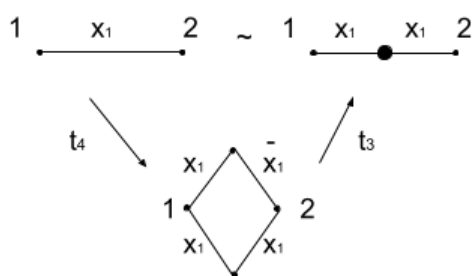


Рис. 18.12. Эквивалентное преобразование

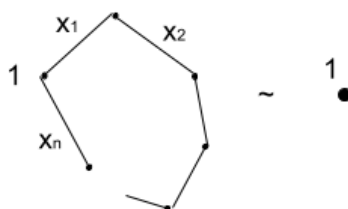


Рис. 18.13. Тожество  $t_6^m$

## Основные тождества. Вывод вспомогательных тождеств

Основные тождества  $\tau_n$  – это все тождества от  $t_1$  до  $t_6^m$ ,  $m = 1, 2, \dots, n$ .  $t_7$  – это первое вспомогательное тождество.

Из основных тождеств можно вывести вспомогательные тождества.  $t_8$  – это трехполюсное тождество. Необходимо проверить эквивалентность двух схем.

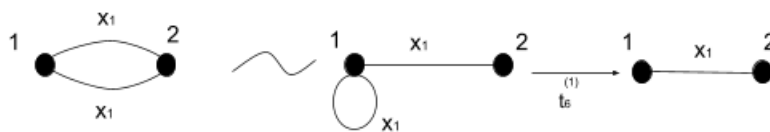


Рис. 18.14. Преобразование модификации  $\hat{t}_5$

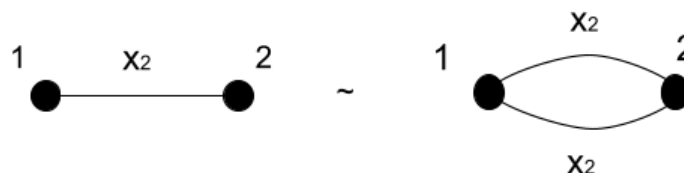


Рис. 18.15. Тождество  $t_7$

Происходит нетривиальный вывод вспомогательного тождества. Берется под-схема, которая состоит из одного контакта  $x_2$ . Объявляются полюса подсхемы и применяется тождество  $t_4$ . Далее 2 раза применяется тождество  $t_5$  (сбра-сывание однотипных контактов). Выделяется подсхема с полюсами 2 и 3, где применяется тождество  $t_3$  для удаление цепи между полюсами.

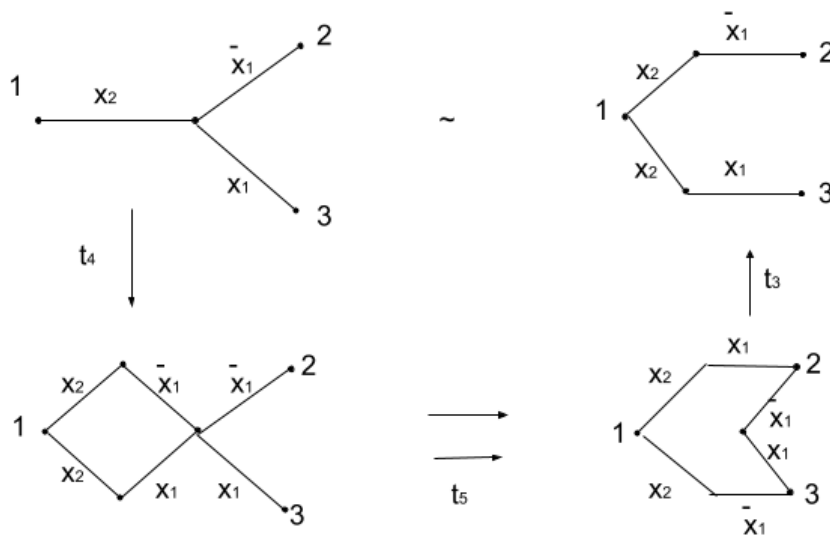


Рис. 18.16. Вывод тождества  $t_8$

Выводится тождество  $t_9$  с однополюсной подсхемой и висячим контактом. Это эквивалентно одному висячему полюсу с той же пометкой. Сначала из

одного контакта  $x_1$  необходимо сделать 2 параллельных контакты  $x_1$ , применяя тождество  $t_7$ , но для этого необходимо объявить висячую вершину полюсом этой подсхемы. Далее, применяя тождество  $t_6^{(2)}$ , можно сбросить цикл и получить изолированную полюсную вершину с пометкой 1.

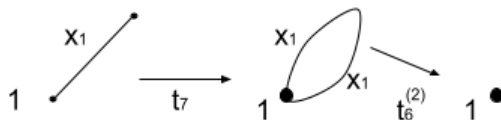


Рис. 18.17. Вывод тождества  $t_9$

Выводится тождество  $t_{10}$  — замыкание по транзитивности. Сначала применяется тождество  $t_7$  и контакт дублируется. По тождеству  $t_5$  нижний контакт  $x_1$  перебрасывается между полюсами 1 и 3.

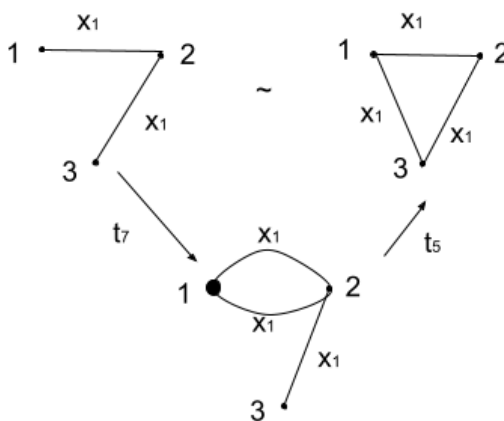


Рис. 18.18. Вывод тождества  $t_{10}$

Выводится тождество  $t_{11}$  — Лемма о звезде. Выделяется подсхема и применяется тождество  $t_7$ . Далее применяется тождество  $t_5$ .

**Утверждение 18.1.** Из системы основных тождеств  $\tau_n$  можно вывести все вспомогательные тождества.

## Полная система тождеств. Обобщённые тождества

Чтобы доказать полноту тождества, необходимо ввести обобщенные тождества. Их обобщение заключается в том, что вместо отдельных контактов рас-

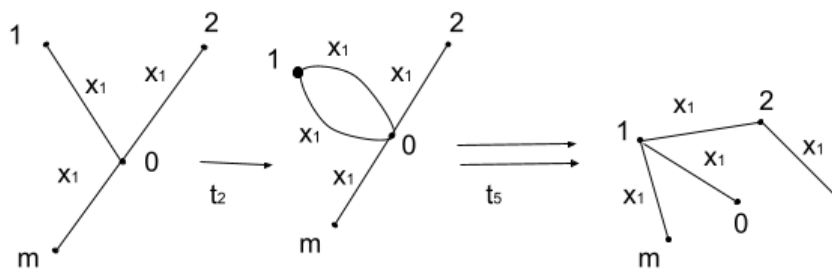


Рис. 18.19. Вывод тождества  $t_{11}$

сматриваются целые цепочки контактов. Все обобщенные тождества представляют собой основные тождества. Первым примером обобщенного тождества является каноническая цепь  $I_j$  порядка  $n$  и номера  $j$ .

$$j = v(\sigma_1, \dots, \sigma_n) + 1$$

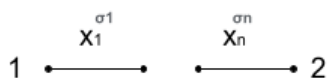


Рис. 18.20. Цепь  $I_j$

Также рассматривают укороченные варианты цепочек  $I'$  и  $I''$ .

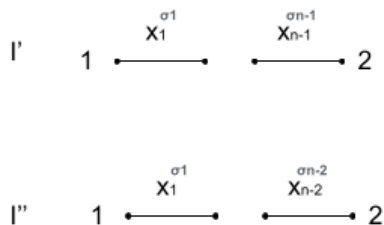


Рис. 18.21. Укороченные цепи

Обобщенные тождества порядка  $n$ :

$$\tau_{(n)} = \{t_1^{(n)} = t_1, t_2^{(n)}, t_2^{(n)} \dots t_{11}^{(n)}\}$$

$\tilde{I}$  — перестановка контакта в цепи  $I$ .

$$I = I^{2^n}$$

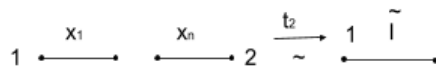


Рис. 18.22. Обобщенное тождество  $t_2^{(n)}$

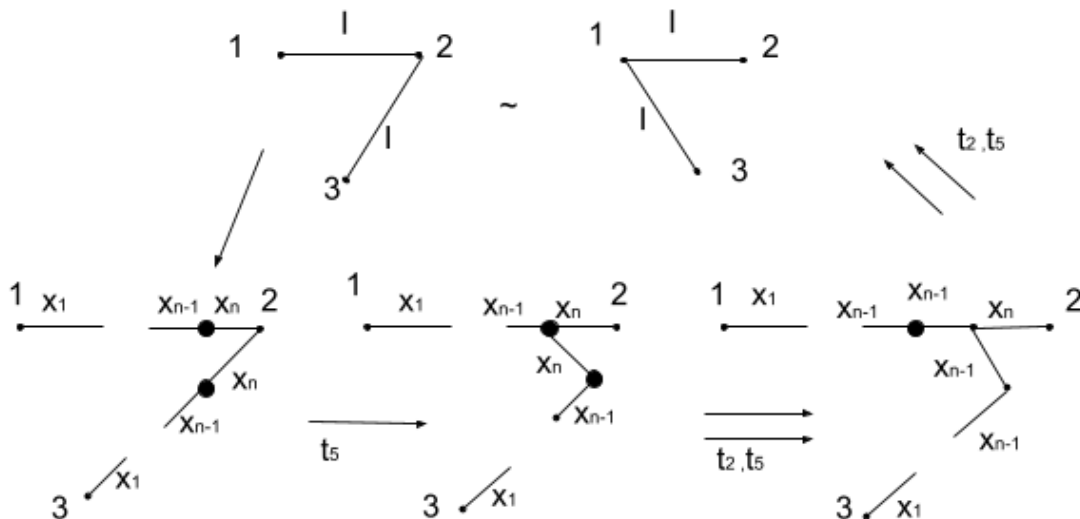


Рис. 18.23. Вывод тождества  $t_5^{(n)}$

Из тождеств  $t_2^{(n)}$ ,  $t_5^{(n)}$ ,  $t_6^{(n)}$  выводятся тождества  $t_7^{(n)}$ ,  $t_9^{(n)}$ ,  $t_{11}^{(n)}$  аналогично выводу вспомогательных тождеств  $t_7$ ,  $t_9$ ,  $t_{11}$ .

Выводится тождество  $t_8^{(n)}$ .

$$t_8 = t_8^{(2)}$$

Происходит индукция по  $n$  и применяется тождество  $t_8$ . Далее применяется тождество  $t_8^{(n-1)}$ .

Выводится тождество  $t_3^{(n)}$ . Рассматриваются две соседние цепочки и по тождеству  $t_8^{(n)}$  склеиваются начальные пары каждого отрезка. Далее применяются тождества  $t_8^{(n-1)}$  и  $t_2$  и получаются изолированные полюса.

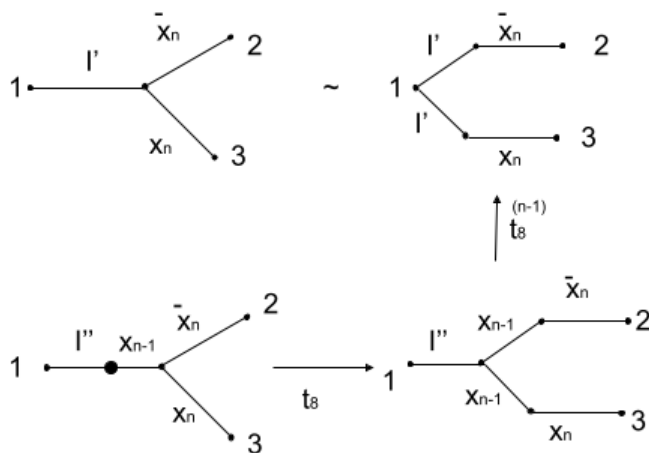


Рис. 18.24. Вывод тождества  $t_8^{(n)}$

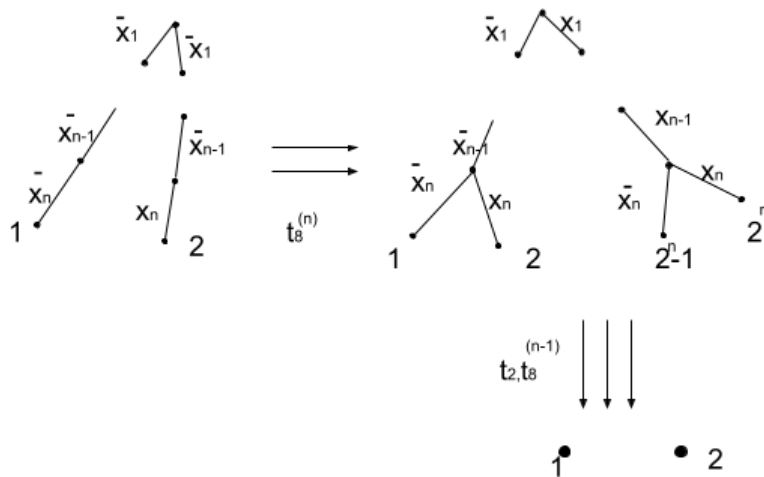


Рис. 18.25. Вывод тождества  $t_3^{(n)}$

## Лекция 19. Полнота системы основных тождеств

### Вывод обобщённых тождеств

$\tau_n$  — система основных тождеств. Есть 5 обычных тождеств и тождество  $t_6^{(n)}$ , у которого длина цикла сбрасывается. Необходимо доказать, что  $\tau_\infty$  является полной системой тождеств. Из системы основных тождеств можно вывести вспомогательные тождества. Вместо отдельных контактов могут появляться цепочки или основные цепочки.

Тождество  $t_1^{(n)} = t_1$  позволяет избавляться от висячей не полюсной вершины. Тождество  $t_2^{(n)}$  заключается в том, что основную цепочку можно заменить любой цепочкой, которая состоит из тех же контактов, но в этой цепочке контакты можно переставлять. Цепочки могут быть укороченными.

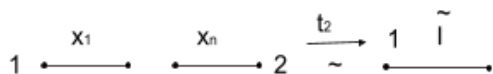


Рис. 19.1. Тождество  $t_2^{(n)}$

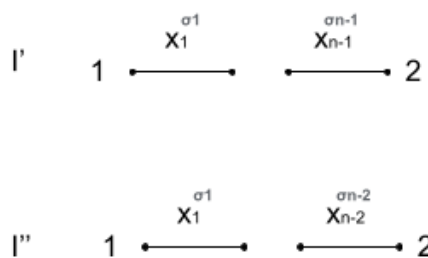


Рис. 19.2. Укороченные цепи

В тождестве  $t_5^{(n)}$  имеется дело с перебрасыванием контакта.

$\tau^{(n)}$  — система обобщенных тождеств порядка  $n$ .

**Утверждение 19.1.** Из системы основных тождеств можно вывести систему обобщенных тождеств.

### Каноническая форма КС. Переход к канонической форме через ЭП

Система основных тождеств полна, но она счетная. Рассматривается контактная схема  $\Sigma = \Sigma(x_1, \dots, x_n; a_1, \dots, a_p)$  с неразделенными полюсами.  $\hat{\Sigma}$  — кано-

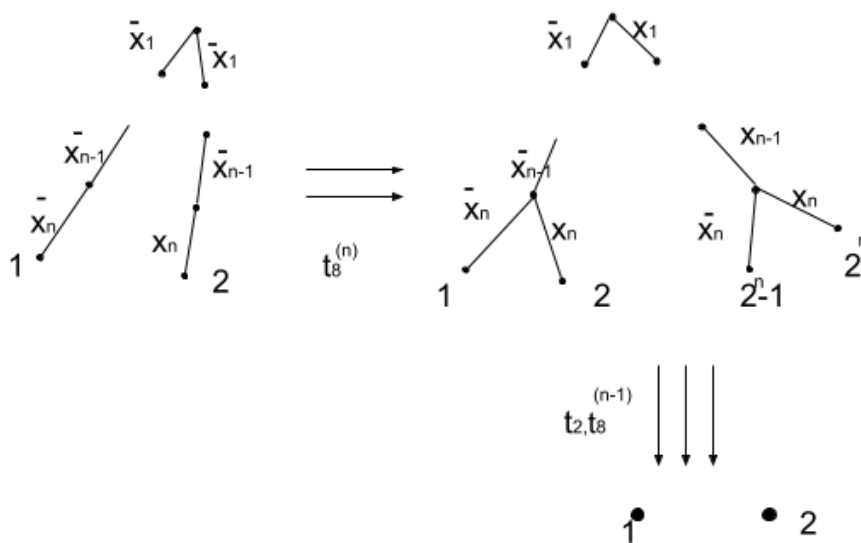


Рис. 19.3. Тожество  $t_3^{(n)}$

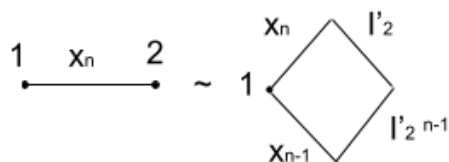


Рис. 19.4. Тожество  $t_4^{(n)}$

ническая форма схемы  $\Sigma$  тогда и только тогда, когда:

$$\hat{\Sigma} = \bigcup_{1 < i < j < p} \hat{\Sigma}_{ij}$$

$\hat{\Sigma}_{ij}$  —  $\pi$ -схема, построенная по совершенной ДНФ функции  $f_{ij}$  и натянутая на эти 2 полюса:  $a_i$  и  $a_j$ .

**Утверждение 19.2.** Для любой схемы  $\Sigma = \Sigma(x_1, \dots, x_n; a_1, \dots, a_p)$  существует эквивалентное преобразование и с помощью систем основных тождеств можно перейти от схемы  $\Sigma$  к ее каноническому виду  $\hat{\Sigma}$ .

Доказательство утверждения заключается в том, что исходная схема приводится к схеме, обладающей такими свойствами, как:



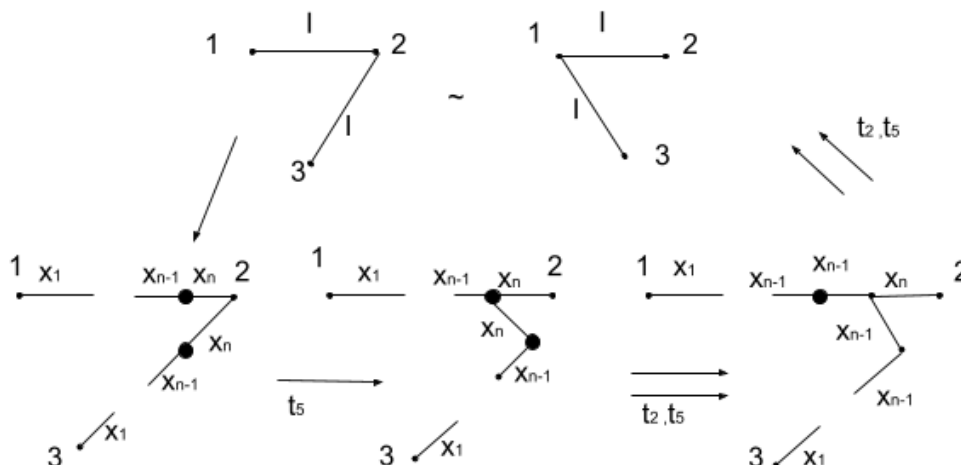


Рис. 19.5. Тожество  $t_5^{(n)}$

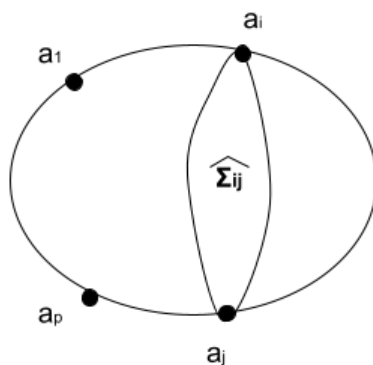


Рис. 19.6. Схема  $\hat{\Sigma}$

- 1) любой контакт принадлежит некоторой канонической цепи.
- 2) любая внутренняя вершина схемы является внутренней вершиной некоторой канонической цепи.
- 3) в схеме нет висячих циклов.
- 4) в схеме нет транзитных проводимостей. Это означает, что если схема каноническая, то каждый контакт лежит на некоторой цепи. Если есть транзитная проводимость, то обязательно есть прямая цепь, соединяющая полюса  $a_i$  и  $a_j$ .

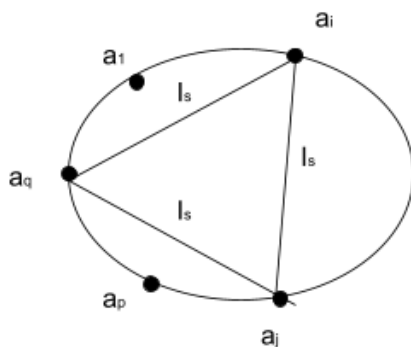


Рис. 19.7. Транзитная цепь

**Доказательство.**

Строится эквивалентное преобразование схемы  $\Sigma = \Sigma(x_1, \dots, x_n; a_1, \dots, a_p)$ .

$$\Sigma \xrightarrow[t_4^{(n)}]{} \Sigma^{(1)} \xrightarrow[\tau']{} \Sigma^{(2)} \xrightarrow[t_6^{(n)}]{} \Sigma^{(3)} \xrightarrow[t_{10}^{(n)}]{} \Sigma^{(4)}$$

$\Sigma^{(4)}$  обладает всеми нужными 4 свойствами и является канонической.

$$\tau' = \{t_{11}^{(n)}, t_9^{(n)}, t_3^{(n)}, t_1\}$$

На первом этапе необходимо получить схему, в которой каждый контакт лежит на канонической цепи. Для этого достаточно к каждому контакту из схемы  $\Sigma$  применять тождество  $t_4^{(n)}$ . На втором этапе идет индукция по числу тех вершин, которые вторым свойством не обладают. Пусть есть вершина  $v$ , в которой сходятся канонические цепи типа  $I_{i_1}, I_{i_2}, I_{i_q}$ . Используя тождество  $t_{10}^{(n)}$  останется только одна цепь из каждой группы. Дальше применяется тождество  $t_9^{(n)}$  и на вершину  $v$  привешиваются недостающие цепочки и появляются вершины. Изолировать эти вершины можно тождеством  $t_3^{(n)}$ , а убрать эти вершины можно тождеством  $t_1$ . В результате всех этих преобразований можно избавиться от вершины  $v$ .

Чтобы избавиться от всяких циклов, можно использовать тождество  $t_6^{(n)}$ , а чтобы избавиться от транзитных проводимостей, есть тождество  $t_{10}^{(n)}$ . ■

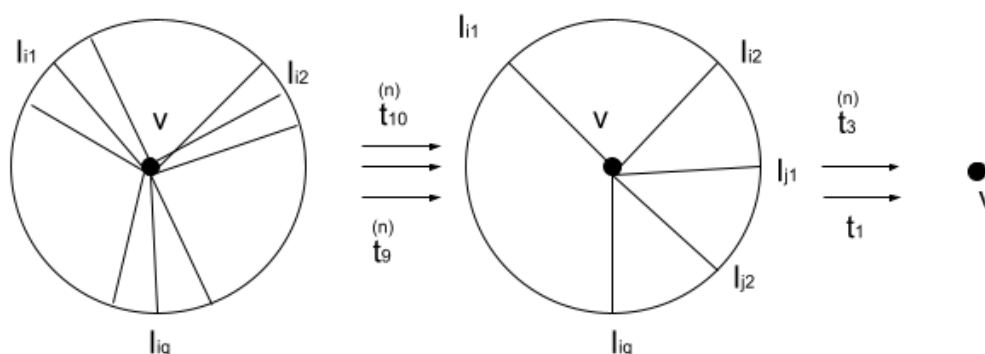


Рис. 19.8. Шаг 2. Удаление вершины  $v$

### ЭП эквивалентных КС

**Утверждение 19.3.** Система тождеств  $\tau_n$  — это конечная полная система тождеств для эквивалентных преобразований КС от переменных  $x_1, \dots, x_n$ .

**Доказательство.**

$\Sigma' \sim \Sigma''$  от переменных  $x_1, \dots, x_n$ . Эти обе схемы приводятся в канонический вид, которые потом можно преобразовывать в исходные схемы. ■

## Лекция 20. Цикломатическое число КС

### Суммарное цикломатическое число КС. Пример

Пусть  $G$  — это граф,  $V(G)$  — множество вершин,  $E(G)$  — множество ребер графа и  $c(G)$  — множество связанных компонент графа. Вводится цикломатическое число графа  $G$ .

$$\theta(G) = |E(G)| - |V(G)| + c(G)$$

Цикломатическое число — это максимальное число линейно независимых простых циклов, которые можно вложить в граф  $G$ .

$$C = C_1 + C_2$$



Рис. 20.1. Сложение циклов

Выполняется перенос этой характеристики на схему.

$$\Sigma = (x_1, \dots, x_n) \rightarrow \theta(\Sigma) = \sum_{\alpha \in B^n} \theta(\Sigma|_{\alpha})$$

$\Sigma|_{\alpha}$  — это граф  $G$ , получившийся из  $\Sigma$  удалением всех контактов вида  $x_1^{\bar{\alpha}_1}, \dots, x_n^{\bar{\alpha}_n}$  и заменой остальных контактов ребрами. Таким образом, получается суммарное цикломатическое число КС  $\Sigma$ .

**Пример 20.1.** Рассматриваются схемы  $\Sigma'$  и  $\Sigma''$ , которые эквивалентны.

$$\Sigma' = x_1(x_2 \vee x_3)$$

$$\Sigma'' = x_1x_2 \vee x_1x_3$$

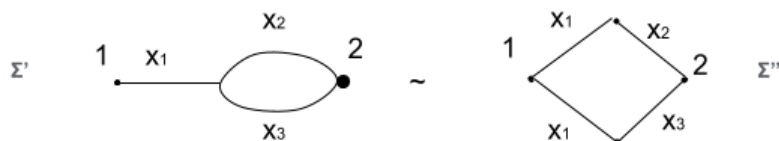


Рис. 20.2. Схемы  $\Sigma'$  и  $\Sigma''$

Эти 2 формулы эквивалентны и получаются друг из друга тождеством дистрибутивности конъюнкции по отношению к дизъюнкции. Считаются суммарные цикломатические числа для этих схем:

$$\theta(\Sigma') = 2$$

$$\theta(\Sigma' |_{\alpha}) = 0 \quad \alpha \neq (011), (111)$$

$$\theta(\Sigma' |_{(011)}) = \theta(\Sigma |_{(111)}) = 1$$

$$\theta(\Sigma'') = 1$$

$$\theta(\Sigma'' |_{\alpha}) = 0 \quad \alpha \neq (111)$$

$$\theta(\Sigma'' - |_{(111)}) = 1$$

## Утверждение об изменении суммарного цикломатического числа при ЭП. Пример

**Утверждение 20.1.** Если:

$$\Sigma'(x_1, \dots, x_n) \xrightarrow[t_1, t_5]{\Sigma''} \Sigma''(x_1, \dots, x_n)$$

то их суммарные цикломатические числа совпадают.

$$\theta(\Sigma') = \theta(\Sigma'')$$

Если:

$$\Sigma'(x_1, \dots, x_n) \xrightarrow[t_k]{\Sigma''} \Sigma''(x_1, \dots, x_n) \quad k < n$$

то разность их суммарных цикломатических чисел делится на  $2^{n-k}$ .

В следствии получается следующее:

$$\Sigma' \xrightarrow[t_1, t_5, t_6^{(1)}, t_6^{(2)}]{} \Sigma''$$

Существует эквивалентное преобразование их схемы  $\Sigma'$  в схему  $\Sigma''$  с помощью тождеств  $t_1 - t_5, t_6^{(3)}$ .

Выделяется подсхема в схеме  $\Sigma'$  и объявляются полюса этой подсхемы. Дважды применяется тождество  $t_4$  и вводится фиктивная переменная  $x_1$ . Дальше находятся однотипные контакты. Трижды применяется тождество  $t_5$ , чтобы перебросить контакты  $x_1$ . Дальше применяется тождество  $t_3$  для удаление цепи. Дальше происходит удаление висячего цикла, используя тождество  $t_6^{(3)}$ .

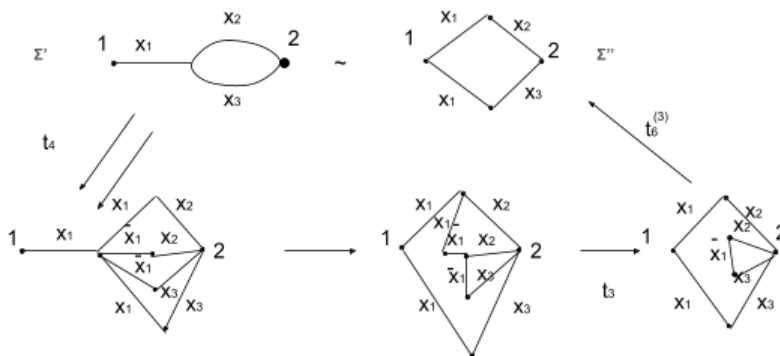


Рис. 20.3. Эквивалентное преобразование  $\Sigma'$

**Доказательство.**

Пусть есть схема  $\Sigma$ , берется ее подсхема и применяется тождество  $t_3$ . Получается схема  $\hat{\Sigma}$ , где цепочка подсхемы исчезла.

$$\forall \alpha \in B^n \quad \theta(\Sigma|\alpha) = \theta(\hat{\Sigma}|\alpha)$$

$$\theta(\Sigma) = \theta(\hat{\Sigma})$$

$x_{j_1}^{c_1} \dots x_{j_l}^{c_l}$  — буквы различных переменных, встречающихся на цикле. Это означает, что цикл будет проводить на  $2^{n-l}$  наборах, где  $l \leq m$ .

$$\theta(\hat{\Sigma}) - \theta(\Sigma) = 2^{n-l}$$

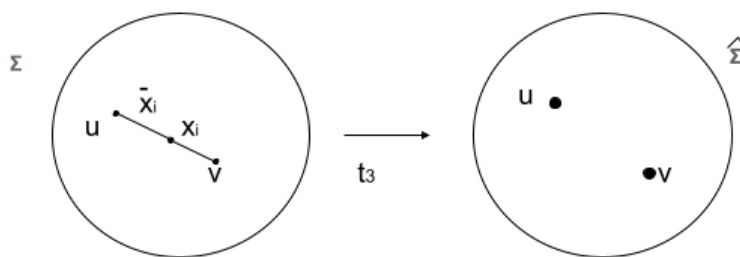


Рис. 20.4. Эквивалентное преобразование  $\Sigma$

Разность суммарных цикломатических чисел делиться на  $2^{n-m}$ . ■

### Доказательство отсутствия конечной полной системы тождеств в классе КС

**Утверждение 20.2.** Пусть в классе  $U^K$  не существует конечной полной системы тождеств.

**Доказательство.**

Первым доказывается то, что из  $\tau_\infty$  нельзя выделить конечную полную подсистему. Предполагается, что  $\tau_N$  — это конечная полная система тождеств. Необходимо доказать, что это предположение неверно. Если это верно, то:

$$\tau_N \mapsto t_6^{(N+1)}$$

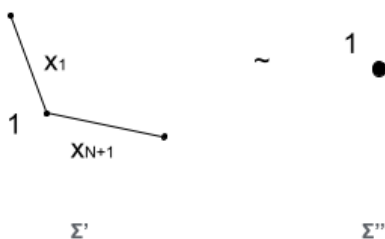


Рис. 20.5. Конечная полная система тождеств

$$\Sigma' = \Sigma'(x_1, \dots, x_{N+1})$$

$$\Sigma'' = \Sigma''(x_1, \dots, x_{N+1})$$

$$\theta(\Sigma') = 1$$

$$\theta(\Sigma'') = 0$$

Если это предположение верно, то разность цикломатического числа должна делиться на  $2^{(N+1)-N}$ . Но разность не делится, и поэтому полученное противоречие доказывает, что нет конечной полной подсистемы в системе основных тождеств.

Дальше необходимо доказать, что  $\tau$  — конечная полная система тождеств для эквивалентных преобразований  $U^K$ . Пусть  $N$  — максимальное число переменных, встречающихся в тождествах системы  $\tau$ . Следовательно из  $\tau_N$  выводятся все тождества системы  $\tau$ . Это доказательство противоречит тому, что  $\tau_N$  не является конечной полной системой тождеств. Таким образом, не существует конечной полной системы тождеств для эквивалентных преобразований КС. ■

## Примеры задач на синтез схем ФАЛ из специальных классов

**Пример 20.2.** *Задается класс  $Q$ .*

$$Q \subseteq P_2 \rightarrow Q(1), \dots, Q(n)$$

$$Q(n) = \{f \in P_2(n) : f \in Q\} = P_2(n) \cap Q$$

*Для этого класса вводится функция Шеннона:*

$$L^A(Q(n)) = \max_{f \in Q(n)} L^A(f) \quad A \in \{K, C\}$$

*Нижние оценки позволяют утверждать следующее:*

$$L^C(Q(n)) \gtrsim \frac{\log |Q(n)|}{\log \log |Q(n)|}$$



$$\frac{\log |Q(n)|}{\log \log |Q(n)|} \gg |Q(n)|$$

$$L^K(Q(n)) \gtrsim \frac{\log |Q(n)|}{\log \log |Q(n)|}$$

Необходимо найти асимптотику этой функции Шеннона. Далее необходимо убедиться в том, что полученная на первом этапе верхняя оценка является асимптотически точной.

### Пример 20.3.

$$Q(n) = \{f(x_1, \dots, x_n) : f(0, 0, x_3, \dots, x_n) = 0\}$$

Решение задачи начинается с подсчета числа функций.

$$|Q(n)| = 2^{\frac{3}{4}2^n}$$

Применяется мощностная нижняя оценка.

$$L^K(Q(n)) \gtrsim \frac{\frac{3}{4}2^n}{n + \log \frac{3}{4}} \sim \frac{3}{4} \frac{2^n}{n}$$

Необходимо найти верхнюю асимптотическую оценку. Происходит разложение произвольной функции:

$$f(x_1, \dots, x_n) = \bar{x}_1 \bar{x}_2 f(0, 0, x_3, \dots, x_n) \vee \bar{x}_1 x_2 g_1(x'') \vee x_1 \bar{x}_2 g_2(x'') \vee x_1 x_2 g_3(x'')$$

Таким образом, верхняя оценка:

$$L(\Sigma_f) \leq 5 + 3L^K(n-3)$$

Необходимо убедиться в том, что верхняя оценка совпадает с нижней оценкой.

$$L^K(n-2) \sim \frac{2^{n-2}}{n-2} \sim \frac{2^{n-2}}{n}$$

$$L^K(Q(n)) \leq 3 \cdot \frac{2^{n-2}}{n} = \frac{3}{4} \frac{2^n}{n}$$

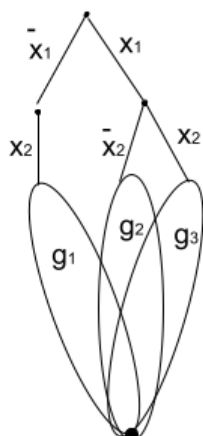


Рис. 20.6. Разложение произвольной функции

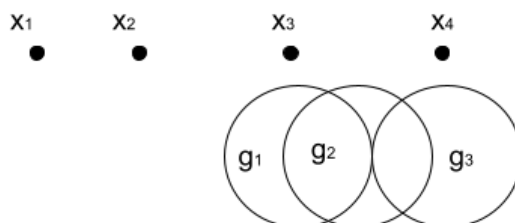


Рис. 20.7. Разложение произвольной функции

**Пример 20.4.** Пусть есть линейная функция от переменных  $x_1$  и  $x_2$ :

$$Q(n) = \{f(x_1, \dots, x_n) : \forall (\sigma_3, \dots, \sigma_n) f(x_1, x_2, \sigma_3, \dots, \sigma_n)\}$$

Находится число функций:

$$|Q(n)| = 8^{2^{n-2}}$$

Записывается мощностная нижняя оценка:

$$L^C(Q(n)) \gtrsim \frac{3}{4} \frac{2^n}{n}$$

Разложение функции записывается следующим образом:

$$f = x_1 g_1(x_3, \dots, x_n) \oplus x_2 g_2(x_3, \dots, x_n) \oplus g_3(x_3, \dots, x_n)$$

$$L(f) \leq 10 + 3L^K(n-2)$$

*Получается верхняя оценка, которая совпадает с нижней оценкой:*

$$L^K(Q(n)) \leq \frac{3 \cdot 2^n}{4n}$$

## Лекция 21. Модификации классов схем

### Вычисляющая программа. Команды ввода, вычисляющие команды, вывод

Пусть есть схема  $\Sigma = \Sigma(x_1, \dots, x_n; z_1, \dots, z_m)$  — СФЭ над базисом  $B = \{\varepsilon_i\}_{i=1}^b$ , где  $\varepsilon_i$  — элемент, реализующий базисный функциональный элемент с номером  $i$   $\varphi_i(x_1, \dots, x_{k_i})$ . Рассматривается схема из функциональных элементов и вычисляющая программа.  $\Sigma$  — вычисляющая программа, если на  $V = V(\Sigma)$  задана монотонная нумерация. Каждой вершине схемы придается номер числами от 1 до  $p$ , где  $p$  — это число вершин.

$$p = |V(\Sigma)|$$

Монотонная нумерация означает, что для любых 2 вершин:

$$\forall u, v : (u, v) \in E(\Sigma)$$

Следовательно, если это выражение верно, то:

$$(v) > (u)$$

Вершина, которая является начальной вершиной дуги всегда имеет меньший номер, чем вершина, которая является концевой вершиной.

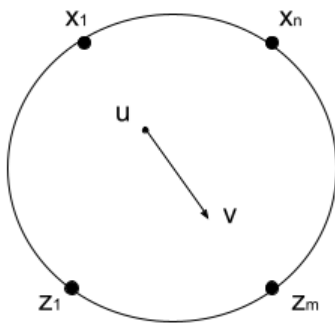


Рис. 21.1. Схема  $\Sigma$

Если была задана такая нумерация, то схему  $\Sigma$  можно считать вычислительной программой, которая работает на одном процессоре. Этот процессор

за единицу времени и за один такт своей работы может вычислять одну из базисных функций. Входные вершины получают номера от 1 до  $n$ .

$$p = n + \text{число функциональных элементов}$$

$$p = n + L(\Sigma)$$

$v_i$  — вершина с номером  $i$ , где  $i \in [1, p]$ . Этой вершине сопоставляется внутренняя БП  $u_i$ . Все команды в вычислительной программе делятся на 3 группы:

- 1) Команды ввода. Они присваивают внутренним переменным значение входных переменных:

$$u_1 := x_1, \dots, u_n := x_n$$

- 2) Вычисляющие команды.

$$u_i := \varphi^{(i)}(u_1^{(i)}, \dots, u_{k^{(i)}}^{(i)})$$

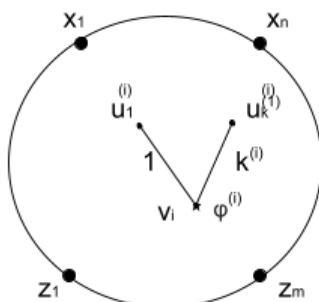


Рис. 21.2. Вычисляющие команды

Каждой вычисляющей команде соответствует вершина элемент. Эта команда выполняется на такте с номером  $i$ .

- 3) Команда вывода данных.

$$z_1 := u_{j_1}, \dots, z_m := u_{j_m}$$

## Ширина программы. Пример

Необходимо найти количество ячеек памяти надо запасти, чтобы правильно организовать работу вычисляющей программы. Переменные  $u_1, \dots, u_n$  хранятся постоянно. Дальше появляются дополнительные ячейки с номерами  $n + 1, \dots, n + r$ . В этих ячейках памяти необходимо размещать переменные  $u$ , которые вычисляются на втором этапе работы программы. Необязательно, чтобы этих ячеек было столько же, сколько вершин функциональных элементов. Одну ячейку можно использовать несколько раз.

Вычисляющая программа  $\Sigma$  работает в момент времени  $1, 2, \dots, p + m$ . На каждую команду тратится 1 единица времени. Пусть для каждого  $i = 1, \dots, p$  переменная  $u_i$  вычисляется в момент времени  $i$ , попадает в ячейку памяти и хранится в ней до момента времени  $i + a_i$ .  $a_i$  — это максимум номеров концевых вершин дуг, исходящих из вершины  $v_i$ , имеющий номер  $i$ . Следовательно,  $r$  — это максимальное по всем  $i$  число пересекающихся отрезков вида  $[i, a_i]$ . Это — ширина программы  $\Sigma$  —  $H(\Sigma)$ .

**Пример 21.1.** Пусть есть схема с 3 входами:  $x_1, x_2, x_3$ . У этой схемы 7 функциональных элементов. Имеется 7 внутренних вершин и 1 из них выходная вершина  $z_1$ .

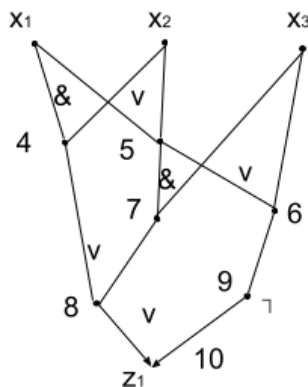


Рис. 21.3. Схема  $\Sigma$

Необходимо найти количество ячеек памяти, чтобы организовать работу этой программы. Вычисляются ширина и сложность этой программы:

$$H(\Sigma) = 3$$

$$L = 7$$

Если переделать эту схему, число элементов останется прежним, а ширина уменьшится до 2.

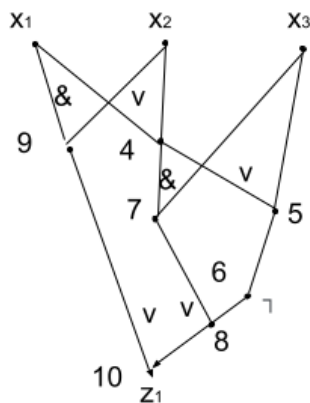


Рис. 21.4. Схема  $\Sigma'$

$$H(\Sigma') = 2$$

Любую ФАЛ можно вычислить некоторой СФЭ с шириной 2.

### Двоичная решающая диаграмма (BDD)

В этой программе нет никаких команд условного или безусловного перехода. На самом деле эти команды являются важным элементом любой вычисляющей программы. Есть другие модели, в которых появляются эти команды. Это — двоичная решающая диаграмма (BDD). Это примерно то же самое, что и каскадная контактная схема с 1 входом и 2 выходами, построенная из ориентированных контактов. Из каждой внутренней вершины  $v$  этой схемы исходят 2 дуги.

В вершине  $b_1$  реализуется  $f(x_1, \dots, x_n)$ , а в вершине  $b_0$  реализуется ее отрицание  $\bar{f}(x_1, \dots, x_n)$ .

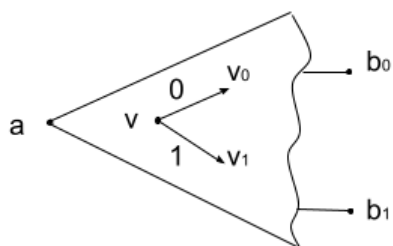


Рис. 21.5. Двоичная решающая диаграмма

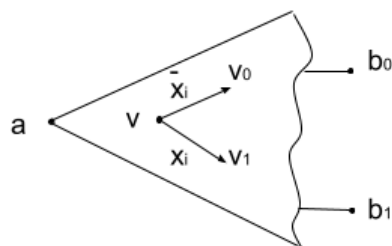


Рис. 21.6. ККC

### Схема на КМОП-транзисторах. Пример

Транзисторный уровень — это, когда из контактов (транзисторов) собираются небольшие блоки типа функциональных элементов. Размыкающий контакт проводит тогда и только тогда, когда переменная  $x_i = 0$ . В КМОП технологии он соответствует транзистору с индуцированным каналом — *P*-МОП транзистор. Этот транзистор работает так же, как и размыкающий контакт, когда на затворе имеется низкий потенциал. Единственное отличие в том, что транзистор "хорошо" проводит высокий потенциал (1).



Рис. 21.7. *P*-МОП транзистор

Замыкающему контакту соответствует *N*-МОП транзистор, который "хорошо" проводит низкий потенциал (0).

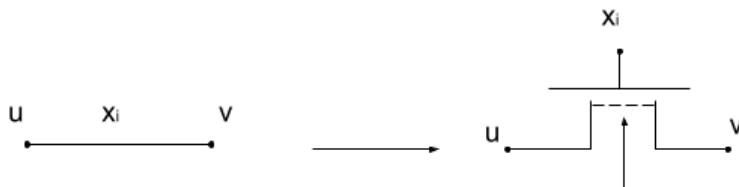


Рис. 21.8. *N*-МОП транзистор



Если функция равна 1, то от источника высокого потенциала к этому выходу должны идти проводящая цепь, состоящая из открытых транзисторов  $P$  типа, и не должно быть цепей, соединяющих выход с источником низкого потенциала.

Предполагается, что необходимо реализовать функцию  $F = (f_1, \dots, f_m) \in P_2(n)^m$ . Строятся две контактные схемы:  $\Sigma'$  и  $\Sigma''$ . На выходе  $\Sigma'$  реализуется система функций  $f_1, \dots, f_m$ . Функция проводимости  $\Sigma' - f_j$ . Функция проводимости  $\Sigma'' - \bar{f}_j$ .

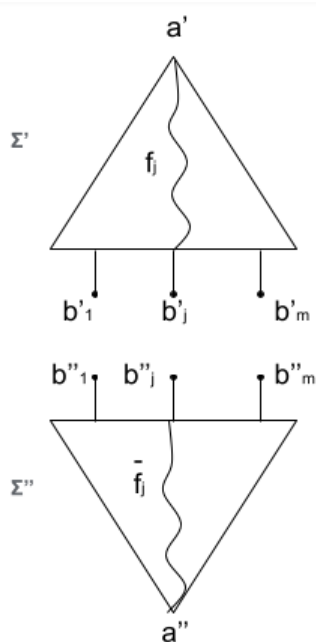


Рис. 21.9. Схемы:  $\Sigma'$  и  $\Sigma''$

Чтобы получить КМОП схему, на схеме  $\Sigma'$  на входе ставится высокий потенциал 1 и контакт  $x_i^\sigma$  заменяется  $P$  транзистором, у которого на затворе  $x_i^{\bar{\sigma}}$ . На входе схемы  $\Sigma''$  ставится низкий потенциал 0 и контакт  $x_i^\sigma$  заменяется  $N$  транзистором  $x_i^{\bar{\sigma}}$ . Входы этих схем соединяются.

## Автомат. Скалярные и векторные канонические уравнения. Схемная реализация автомата. Элементы задержки

Конечный автомат — это устройство с входным каналом и выходным каналом. По входному каналу отправляются входные значения.  $A = \{a_1, \dots, a_p\}$  —

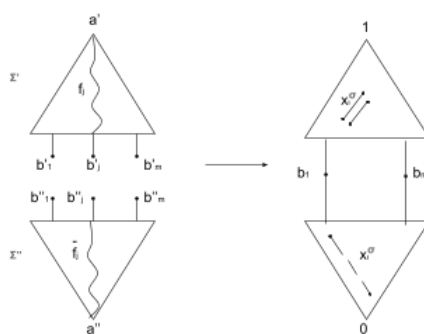


Рис. 21.10. КМОП схема

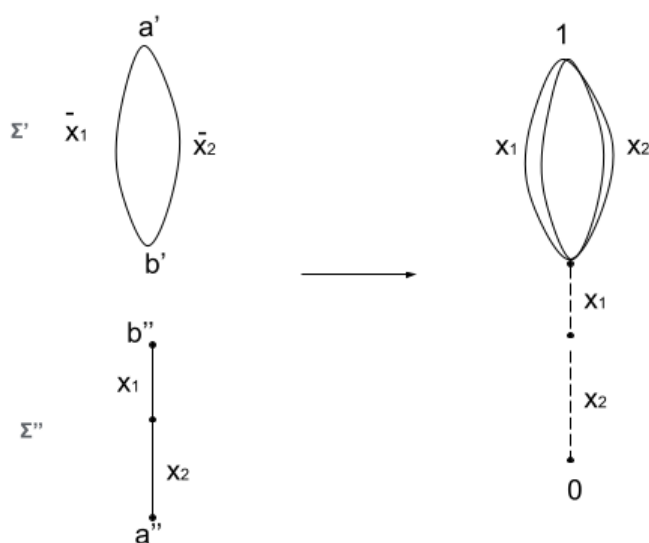


Рис. 21.11. Синтез КМОП схемы

входной алфавит.  $Q = \{Q_0, \dots, Q_{r-1}\}$  — алфавит состояний.  $W = \{w_1, \dots, w_s\}$  — выходной алфавит.

$$a = \langle A, Q, W, f, g, Q_0 \rangle$$

$$f: A \times Q \rightarrow W$$

$$g: A \times Q \rightarrow Q$$

Есть дискретная школа времени  $t = 1, 2, \dots$ . Так же есть скалярные канонические уравнения, которые задаются функциями  $f$  и  $g$ .

$$q(t+1) = g(x(t), q(t))$$

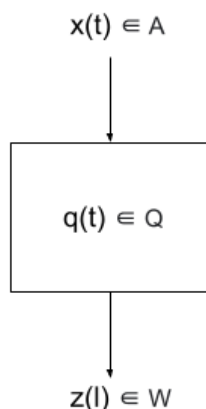


Рис. 21.12. Конечный автомат

$$z(t) = f(x(t), q(t))$$

$$q(1) = Q_0$$

Входной символ, состояние и выходной символ — это вектора из 0 и 1 определенной длины.

$$A = B^n$$

$$Q = B^l$$

$$W = B^m$$

После скалярных канонических уравнений записываются векторные канонические уравнения.

$$g = (g_1, \dots, g_l) \in P_2^l(n+l)$$

$$f = (f_1, \dots, f_m) \in P_2^m(n+l)$$

$$q = (q_1, \dots, q_l)$$

$$x = (x_1, \dots, x_n)$$

$$z = (z_1, \dots, z_m)$$

$$q_1(t+1) = g_1(x_1(t), \dots, x_n(t), q_1(t), \dots, q_l(t))$$

$$q_l(t+1) = g_l(x_1(t), \dots, x_n(t), q_1(t), \dots, q_l(t))$$

$$z_1(t) = f_1(x_1(t), \dots, x_n(t), q_1(t), \dots, q_l(t))$$

$$z_m(t) = f_m(x_1(t), \dots, x_n(t), q_1(t), \dots, q_l(t))$$

$$q_1(1) = q_2(1) \dots = q_l(1) = 0$$

Таким образом, получаются нулевые начальные условия

$z$  — элемент единичной задержки. С помощью элемента задержки можно получить нулевые начальные условия. Через элемент задержки появляется обратная связь.

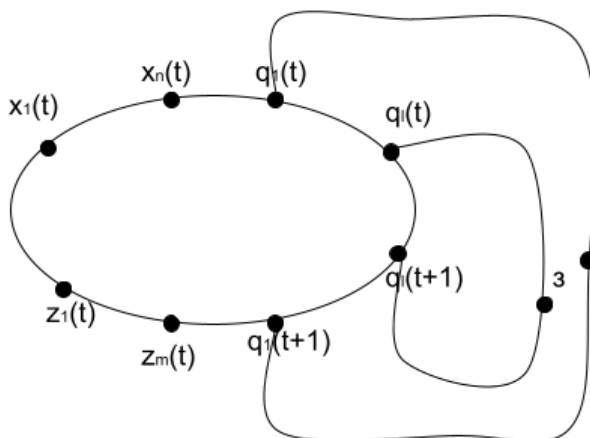


Рис. 21.13. Обратная связь

### Реализация элементов задержки через схемы с "мгновенными" обратными связями

Элемент задержки синтезируется из конъюнкции, дизъюнкции и отрицания.

$$q_1(t+1) = x_1(t) \cdot x_2(t) \vee \bar{x}_2(t) \cdot q_1(t)$$

$$z_1(t) = q_1(t)$$

$$q_1(1) = 0$$

Есть ячейка памяти, у которой  $x_1$  — информационный вход,  $x_1$  — разрешение записи. Чтобы обеспечить задержку в схему добавляются 2 отрицания.

Любое нечетное отрицательное число в цикле создает генератор импульса и задает шкалу времени. В этой шкале времени строится единичная задержка.

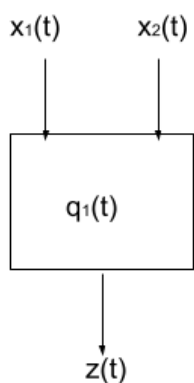


Рис. 21.14. Схема автомата

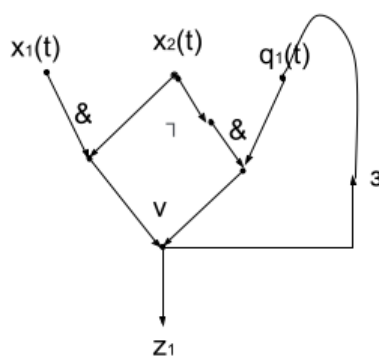


Рис. 21.15. Элемент задержки

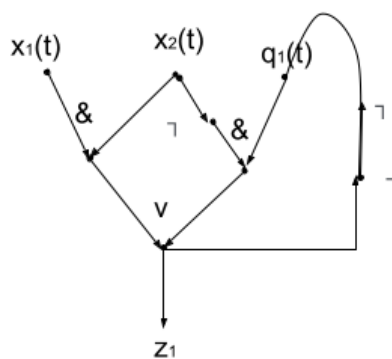


Рис. 21.16. Асинхронная схема

$c = u$  — тактовый генератор. 2 ячейки памяти соединяются последовательно. К второй ячейке идет прямой импульс, а к первой ячейке через инвертор.

$$z(T + 1) \stackrel{?}{=} x(T)$$

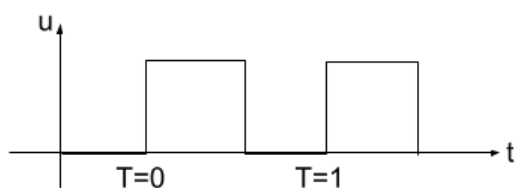


Рис. 21.17. Диаграмма работы устройства

Все единичные задержки управляются одним тактовым генератором и импульсом  $c$ .

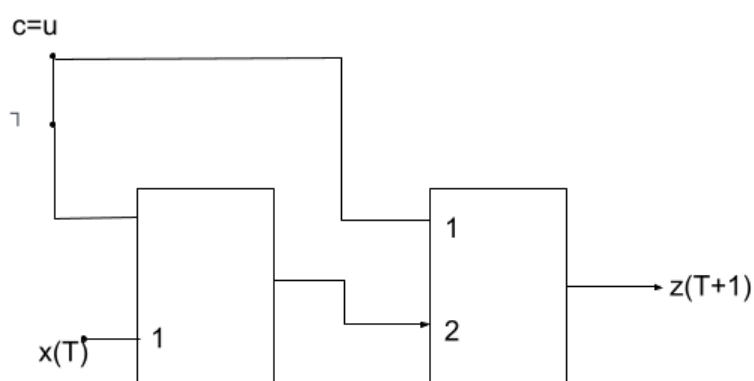


Рис. 21.18. Ячейки памяти



ФАКУЛЬТЕТ  
ВЫЧИСЛИТЕЛЬНОЙ  
МАТЕМАТИКИ И  
КИБЕРНЕТИКИ  
МГУ ИМЕНИ  
М.В. ЛОМОНОСОВА

*teach-in*  
ЛЕКЦИИ УЧЕНЫХ МГУ