

А.А. ПРИХОЖИЙ

**ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ
ЛОГИЧЕСКИЕ СИСТЕМЫ И АЛГОРИТМЫ**

Минск
БНТУ
2013

УДК 681.3

Прихожий, А.А. Частично определенные логические системы и алгоритмы / А.А. Прихожий. – Минск : БНТУ, 2013. – 343 с. – ISBN 978-985-550-179-5.

Исследованы фундаментальные основы построения компьютерных программ и цифровых устройств в не полностью специфицированных областях. Предложена теория, являющаяся базисом для спецификации, моделирования, преобразования, синтеза, распараллеливания, верификации и оптимизации частично определенных логических систем и алгоритмов. Издание ориентировано на подготовку научных кадров, аспирантов, магистрантов и инженеров по соответствующим учебным и научным специальностям.

Табл. 57. Ил. 172. Библиогр. 109 назв.

Рекомендовано к изданию научно-техническим советом
Белорусского национального технического университета
(протокол № 11 от 21.12.2012 г.)

Рецензенты:

зав. кафедрой «Радиотехнические устройства» БГУИР,
д-р техн. наук, профессор *Н.И. Листонад*;
зав. кафедрой «Робототехнические системы»,
д-р техн. наук, профессор *Г.Н. Здор*

ISBN 978-985-550-179-5

© Прихожий А.А., 2013
© Белорусский национальный
технический университет, 2013

СОДЕРЖАНИЕ

| | |
|---|----|
| ПРЕДИСЛОВИЕ | 9 |
| <i>PREFACE</i> | 11 |
| ВВЕДЕНИЕ | 12 |
| Глава 1. НЕ ПОЛНОСТЬЮ ОПРЕДЕЛЕННЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ | 14 |
| 1.1. Булевы функции | 14 |
| 1.2. Разложения булевых функций | 15 |
| 1.3. Формы представления булевых функций. | 16 |
| 1.4. Не полностью определенные булевы функции. | 19 |
| 1.5. Минимизация не полностью определенных булевых функций картами Карно. | 20 |
| 1.6. Двухуровневая минимизация не полностью определенных булевых функций методом Квайна-МакКласки. | 22 |
| 1.7. Историческая справка | 24 |
| Глава 2. ЧАСТИЧНЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ. ЧАСТИЧНАЯ АЛГЕБРА | 26 |
| 2.1. Частичные логические функции | 26 |
| 2.2. Одноместные частичные операции | 29 |
| 2.3. Двухместные частичные операции | 31 |
| 2.4. Формулы частичной алгебры | 39 |
| 2.5. Сильные законы частичной алгебры | 40 |
| 2.6. Слабые законы частичной алгебры | 43 |
| 2.7. Сильные формы представления частичных функций | 45 |

| | |
|--|----|
| 2.8. Слабые формы представления частичных функций..... | 51 |
| 2.9. Функционально полные базисы..... | 56 |
| 2.10. Историческая справка..... | 57 |

Глава 3. ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ. СМЕШАННАЯ АЛГЕБРА..... 58

| | |
|--|----|
| 3.1. Кодирование частичных переменных парами булевых переменных значения и области определенности..... | 58 |
| 3.2. Частично определенные функции..... | 59 |
| 3.3. Одноместные операции над частично определенными функциями..... | 62 |
| 3.4. Двухместные операции над частично определенными функциями..... | 65 |
| 3.5. Смешанная алгебра..... | 71 |
| 3.6. Обзор законов смешанной алгебры..... | 74 |
| 3.7. Законы преобразования функции значения..... | 75 |
| 3.8. Сильные законы частичной алгебры в смешанной алгебре..... | 82 |
| 3.9. Слабые законы частичной алгебры в смешанной алгебре..... | 84 |
| 3.10. Собственные законы смешанной алгебры..... | 87 |
| 3.11. Историческая справка..... | 88 |

Глава 4. РАЗЛОЖЕНИЯ ЧАСТИЧНО ОПРЕДЕЛЕННЫХ ФУНКЦИЙ..... 90

| | |
|--|-----|
| 4.1. Частичные разложения, обусловленные частичными операциями..... | 90 |
| 4.2. Разложения частично определенной функции по характеристической булевой функции..... | 96 |
| 4.3. Частичные разложения по двум частичным операциям..... | 103 |

| | |
|--|-----|
| 4.4. Частичные разложения по операциям исключающее или и эквивалентность | 108 |
| 4.5. Иерархические частичные разложения | 110 |
| 4.6. Историческая справка | 115 |

Глава 5. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧАСТИЧНО ОПРЕДЕЛЕННЫХ ФУНКЦИЙ

| | |
|---|-----|
| 5.1. Совершенная дизъюнктивно-конъюнктивная форма частично определенной функции | 116 |
| 5.2. Совершенная конъюнктивно-дизъюнктивная форма частично определенной функции | 121 |
| 5.3. Дизъюнктивно-конъюнктивная форма частично определенной функции | 123 |
| 5.4. Конъюнктивно-дизъюнктивная форма частично определенной функции | 125 |
| 5.5. Обобщение формы Руда-Маллера для частично определенной функции | 126 |
| 5.6. Историческая справка | 127 |

Глава 6. ИСПОЛЬЗОВАНИЕ СМЕШАННОЙ АЛГЕБРЫ ДЛЯ ОПТИМИЗАЦИИ ЛОГИЧЕСКИХ СИСТЕМ

| | |
|---|-----|
| 6.1. Задача оптимизации частично определенной логической системы | 129 |
| 6.2. Диапазоны булевых функций и операции над диапазонами | 131 |
| 6.3. Свойства операции минимизации | 139 |
| 6.4. Оптимальное разложение поведения системы по булевой операции | 144 |
| 6.5. Оптимальное разложение поведения системы по характеристической функции | 151 |

| | |
|--|-----|
| 6.6. Оптимальное разложение поведения системы по двум булевым операциям | 154 |
| 6.7. Оптимальное разложение поведения системы по операциям исключаящее или и эквивалентность | 159 |
| 6.8. Оптимизация иерархических разложений логических проектов | 164 |
| 6.9. Несущественные переменные и ортогональные остаточные функции | 171 |
| 6.10. Историческая справка | 174 |

**Глава 7. ПОСТРОЕНИЕ ДИАГРАММ РЕШЕНИЙ
ПО РАЗЛОЖЕНИЯМ ФУНКЦИЙ В БУЛЕВОЙ
И СМЕШАННОЙ АЛГЕБРАХ**..... 176

| | |
|---|-----|
| 7.1. Построение диаграмм решений по разложениям функций в булевой алгебре | 176 |
| 7.2. Построение диаграмм решений по разложениям функций в смешанной алгебре | 186 |
| 7.3. Двоичные if диаграммы решений | 192 |
| 7.4. Частные случаи двоичных if диаграмм решений | 197 |
| 7.5. Функциональные if диаграммы решений | 202 |
| 7.6. Диаграммы решений для частично определенных функций | 208 |
| 7.7. Историческая справка | 214 |

Глава 8. ОПЕРАЦИИ НАД ДИАГРАММАМИ РЕШЕНИЙ 216

| | |
|---|-----|
| 8.1. Булевы операции над двоичными диаграммами решений ... | 216 |
| 8.2. Оптимальное упорядочение переменных в сокращенной упорядоченной двоичной диаграмме решений | 218 |
| 8.3. Операция минимизации частично определенных двоичных диаграмм решений | 221 |

| | |
|---|-----|
| 8.4. Булевы операции над <i>if</i> -диаграммами решений | 226 |
| 8.5. Операция минимизации частично определенных <i>if</i> -диаграмм решений | 228 |
| 8.6. Преобразование двоичной диаграммы решений в <i>if</i> -диаграмму | 231 |
| 8.7. Сокращение глубины <i>if</i> -диаграммы | 234 |
| 8.8. Историческая справка | 236 |

Глава 9. ПРИМЕНЕНИЕ ДИАГРАММ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ

| | |
|---|-----|
| 9.1. Моделирование цифровых систем | 238 |
| 9.2. Распараллеливание вычислений в логических системах | 239 |
| 9.3. Моделирование и синтез сумматоров | 246 |
| 9.4. Моделирование и синтез умножителей | 256 |
| 9.5. Синтез частично определенных проектов | 258 |
| 9.6. Минимизация мощности, потребляемой цифровой системой | 262 |
| 9.7. Верификация логической системы | 271 |
| 9.8. Историческая справка | 272 |

Глава 10. ЛОГИЧЕСКИЙ ВЫВОД В ЧАСТИЧНОЙ ЛОГИКЕ И РЕШЕНИЕ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

| | |
|--|-----|
| 10.1. Методы представления знаний | 273 |
| 10.2. Частичная логика предикатов первого порядка | 276 |
| 10.3. Преобразование частичной формулы | 279 |
| 10.4. Представление не полностью определенных знаний | 284 |
| 10.5. Дедуктивный вывод в частичной логике | 285 |
| 10.6. Переход от частичной к полностью определенной логике | 290 |

| | |
|--|------------|
| 10.7. Дедуктивный вывод на модели частично определенных конечных автоматов | 293 |
| 10.8. Историческая справка | 297 |
| Глава 11. ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ АЛГОРИТМЫ | 298 |
| 11.1. Понятие частично определенного алгоритма | 298 |
| 11.2. Эквивалентные преобразования частично определенных алгоритмов | 302 |
| 11.3. Преобразование частично определенных ветвящихся алгоритмов | 306 |
| 11.4. Преобразование частично определенных циклических алгоритмов | 310 |
| 11.5. Сведение частично определенного алгоритма к системе частично определенных логических функций | 316 |
| 11.6. Синтез частично определенного алгоритма | 319 |
| 11.7. Оптимизация частично определенного алгоритма | 321 |
| 11.8. Историческая справка | 323 |
| ЛИТЕРАТУРА | 324 |
| ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ | 335 |
| ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ | 338 |

ПРЕДИСЛОВИЕ

Цифровые или дискретные системы, описанные программно или аппаратно, имеют поведение, определенное полностью для всех входных наборов и последовательностей или определенное частично лишь на некотором подмножестве наборов и последовательностей. Не полная определенность системы расширяет возможности и повышает эффективность проектирования, синтеза, распараллеливания, верификации и оптимизации по различным параметрам. В настоящей монографии разрабатывается теория частично определенных функций, систем и алгоритмов, показывается ее практическое значение для решения прикладных задач, связанных с созданием аппаратно-программных вычислительных средств.

Монография базируется на работах автора, выполнявшихся на протяжении более 20 лет, и отражает современное состояние исследуемой проблемы. В библиографии приведен обширный перечень работ автора по направлению исследований. Его работы опубликованы такими известными издательствами, как Springer, IEEE, Kluwer Academic Publishers и др. Исследования автора именно по данной тематике поддерживались академическим фондом DAAD, Германия. В списке использованных источников дан также глубокий анализ публикаций других авторов по исследуемой проблеме. По результатам каждой главы монографии приведена историческая справка.

За поддержку данных научных исследований на протяжении многих лет автор выражает искреннюю благодарность директору центра ECSI Жану Мермье (Франция), директору центра СМР Бернарду Куртуа (Франция), директору лаборатории по разработке мультимедийных архитектур Марко Маттавелли (Швейцария), профессору Вольфгангу Небелю и профессору Бернду Беккеру (Германия), профессору Анне-Марие Трулеманс (Бельгия), профессору В.С. Рогозе (Украина) и профессору В.Н. Ярмолику, а также академическому фонду DAAD (Германия). Искренняя благодарность за поддержку работы коллегам по кафедре программного обеспечения и автоматизированных систем, факультету информационных технологий и робототехники, научному сектору БНТУ.

Автор выражает также благодарность рецензентам – директору Информационно-аналитического центра Министерства образования Республики Беларусь, заведующему кафедрой радиотехнических устройств Белорусского государственного университета информатики и радиоэлектроники, доктору технических наук, профессору Листопаду Н.И. и заведующему кафедрой «Робототехнические системы» Белорусского национального технического университета, доктору технических наук, профессору Здору Г.Н. за ценные замечания. Автор будет признателен всем, кто найдет возможность прислать свои замечания, пожелания и указания на замеченные в книге недостатки по электронному адресу prihozhy@yahoo.com.

Анатолий Прихожий
Минск, декабрь, 2012 года

PREFACE

Digital or discrete systems that are implemented as software or hardware have a behavior specified completely for all input sets and sequences or specified incompletely for only a subset of the input sets and sequences. Incompleteness of system specification extends capabilities and increases efficiency of design, synthesis, parallelization, verification and optimization of hardware-software implementations. In the book, a theory of incompletely specified logical functions, systems and algorithms is developed. The theory is applied to solving various application tasks that are aimed at creation of hardware-software computing facilities.

The book is mostly based on **author's** publications written during more than 20 years period and presents state of the art in scientific direction under consideration. The **author's** works are published by such well known publishers as Springer, IEEE, Kluwer Academic Publishers and others. The scientific research was supported by the DAAD foundation, Germany. Key works by other authors on incompletely specified functions, circuits, **systems and don't cares** utilization are listed, analyzed and covered.

My gratitude is large to director of ECSI center Jean Mermet (France), director of CMP center Bernard Courtois (France), director of multimedia architectures research group Marco Mattavelli (Switzerland), professor Wolfgang Nebel and professor Bernd Becker (Germany), professor Anne-Marie Trullemans (Belgium), professor V.S. Rogoza (Ukraine), professor V.N. Yarmolik (Belarus) and the DAAD foundation (Germany) for their significant support of the work and collaboration during last twenty years.

The author acknowledges the colleagues of computer and system software department, faculty of information technologies and robotics and research sector of Belarusian National Technical University who have helped in preparing the manuscript and meeting with approval to publish this book.

Anatoly Prihozhy
Minsk, December, 2012

ВВЕДЕНИЕ

Бурное развитие компьютерной техники, компьютерных и телекоммуникационных сетей, мобильной связи, цифровых систем управления, встроенных управляющих систем, мультимедийных технологий, аудио-видео аппаратуры и сопутствующего программного обеспечения, медицинской техники и других научно-технических направлений вызывает необходимость создания адекватных формальных методов моделирования и проектирования цифровых и дискретных систем.

В ранних работах по проектированию логических систем и их программированию неопределенность рассматривалась как ситуация, приводящая к конфликту, сбою и останову. С.К. Клини впервые предложил рассматривать неопределенность как значение, используемое в процессе вычислений (см. [31 (1952)]). Названные его именем функции Клини адаптируют теорию булевых функций к использованию неопределенного значения и описанию не полностью специфицированных систем.

На протяжении десятков лет исследования по описанию и применению неопределенности развивались главным образом параллельно с развитием технологий проектирования логических схем. Предложен ряд алгоритмов синтеза, оптимизации, генерации тестов и др., использующих неопределенность значений сигналов в процессе своей работы. Вместе с тем эти работы по использованию неопределенности не получили глубокой систематизации и должного уровня развития. Доказательством является уже то, что основные известные разложения булевых функций, такие как разложение Шеннона, разложения Давио и др., применимы только к полностью определенным булевым или псевдобулевым, двухзначным или многозначным функциям и никак не применимы к частично определенным функциям. Сказанное относится также к законам булевой алгебры, формам представления булевых функций, преобразованию булевых выражений.

В 1991 году в Институте технической кибернетики Национальной академии наук Беларуси автором данной книги были опубликованы три брошюры «Логические алгоритмы», «Синтез логических алгоритмов» и «Оптимизация логических алгоритмов». В них

разработан формальный аппарат оперирования неопределенностью в процессе синтеза, преобразования и оптимизации частично определенных алгоритмов. Эти три работы явились отправной точкой для ряда последующих публикаций, включая публикации в зарубежных изданиях. В настоящей монографии этот аппарат получил дальнейшее еще более глубокое развитие.

Данная монография ставит целью создание формальных методов эффективного манипулирования неопределенностью, частично определенными функциями и не полностью специфицированными логическими проектами и алгоритмами в процессе решения задач моделирования, синтеза, распараллеливания, верификации и оптимизации полностью и частично определенных логических программно-аппаратных систем. Предложенный аппарат базируется на теории, связывающей частичную логику с булевой логикой. Замечательным свойством разложений функций, предложенных в рамках теории, является увеличение степени неопределенности продуктов разложения, что является источником расширения возможностей в сравнении с исходной спецификацией по поиску оптимальных структур и параметров цифровых систем и компьютерных программ в процессе решения прикладных оптимизационных задач. Результатом предложенной теории является также новый тип диаграмм решений, использование которых увеличивает быстродействие проектируемых систем и программ или уменьшает техническую сложность и стоимость их реализации.

Вне всякого сомнения, с течением времени значение моделей и средств оперирования неопределенностью для развития компьютеров, компьютерных систем и сетей, информационных технологий, а также средств и технологий программирования будет только возрастать. Автор считает логически закономерным и предсказуемым появление в ближайшем будущем компьютеров и языков программирования для спецификации и реализации частично определенных вычислений. В широком понимании, на решение именно этой научно-технической проблемы направлена предлагаемая вниманию читателя монография.

Глава 1. НЕ ПОЛНОСТЬЮ ОПРЕДЕЛЕННЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ

1.1. Булевы функции

Булевы переменные принимают значения из множества $B = \{0, 1\}$ и обозначаются латинскими буквами x_1, \dots, x_n . Множество из n булевых переменных обозначается $X = \{x_1, \dots, x_n\}$.

Булевой функцией $f(x)$, зависящей от векторной переменной $x = (x_1, \dots, x_n)$, называется отображение $f: B^n \rightarrow B$ множества B^n в множество B . Множество B^n есть декартово произведение множества B на себя n раз. Всего существует 2^{2^n} булевых функций. Примером булевой функции является $f(x) = x_i$. Для набора $a \in B^n$ она реализует проекцию a на компонент i : $a = (a_1, \dots, a_n) \rightarrow a_i$. Отрицанием (инверсией) булевой переменной x_i называется функция $f(x) = \neg x_i$, реализующая проекцию $a = (a_1, \dots, a_n) \rightarrow \neg a_i$. Конъюнкция булевых переменных x_1, \dots, x_n — это функция $f(x) = x_1 \wedge \dots \wedge x_n$, принимающая значение 1, если все переменные имеют значение 1, и принимающая значение 0 в противном случае. Конъюнкция может записываться также в виде последовательности литералов $x_1 \dots x_n$. Функция дизъюнкция $f(x) = x_1 \vee \dots \vee x_n$ принимает значение 0, если все переменные имеют значение 0, и принимает значение 1 в противном случае.

Переменная x_i или ее отрицание $\neg x_i$ называется литералом. Литерал x_i называется положительным, литерал $\neg x_i$ — отрицательным. Конъюнкция литералов называется кубом. Константа 1 есть по определению пустой куб.

Куб из n литералов называется минтермом над множеством X , если все литералы построены на различных переменных. Для векторного значения $a = (a_1, \dots, a_n)$ минтерм $m_a = x_1^{a_1} \dots x_n^{a_n}$, где $x_i^{a_i} = x_i$, если $a_i = 1$, и $x_i^{a_i} = \neg x_i$, если $a_i = 0$.

Литералы и кубы представляют специфические булевы функции. Они могут быть использованы для описания любой m -компонентной векторной булевой функции $f: B^n \rightarrow B^m$, где m — число скалярных булевых функций, входящих в векторную функцию. Множество всех векторных m -компонентных булевых функций обозначается $B_{n,m}$, множество всех скалярных функций — $B_{n,1}$ или B_n .

1.2. Разложения булевых функций

Пусть $f \in B_n$ – некоторая булева функция. *Кофактором* функции f для $x_i = c$ при $c \in \{0, 1\}$ называется функция $f_{x_i=c}$, определяемая следующим образом:

$$f_{x_i=c}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, c, x_{i+1}, \dots, x_n).$$

Остаточная функция $f_{x_i=1}$ называется *положительным кофактором*, остаточная функция $f_{x_i=0}$ называется *отрицательным кофактором*. Часто используется *булева разность* кофакторов, записываемая в виде

$$f_{x_i=1 \oplus 0} = f_{x_i=1} \oplus f_{x_i=0}.$$

Все три кофактора $f_{x_i=1}$, $f_{x_i=0}$ и $f_{x_i=1 \oplus 0}$ принадлежат множеству B_{n-1} булевых функций, поскольку они зависят от не более $n-1$ аргументов. Кофакторы являются теми кирпичиками, из которых строятся разложения булевых функций. *Разложение Шеннона* по переменной x_i сводит булеву функцию f к более простым остаточным функциям кофакторам $f_{x_i=1}$ и $f_{x_i=0}$ посредством булевых операций отрицание, конъюнкция, дизъюнкция:

$$f(x) = x_i \wedge f_{x_i=1} \vee \neg x_i \wedge f_{x_i=0}. \quad (1.1)$$

Положительное разложение Давио по переменной x_i сводит функцию f к кофакторам $f_{x_i=0}$ и $f_{x_i=1 \oplus 0}$ используя булевы операции конъюнкция и исключающее или:

$$f(x) = f_{x_i=0} \oplus x_i \wedge f_{x_i=1 \oplus 0}. \quad (1.2)$$

Отрицательное разложение Давио по переменной x_i сводит функцию f к кофакторам $f_{x_i=1}$ и $f_{x_i=1 \oplus 0}$, используя булевы операции отрицание, конъюнкция и исключающее или:

$$f(x) = f_{x_i=1} \oplus \neg x_i \wedge f_{x_i=1 \oplus 0}. \quad (1.3)$$

В силу взаимной исключаемости операндов, операция \vee в разложении Шеннона (1.1) может быть заменена операцией \oplus . В результате получаем еще одну версию разложения Шеннона:

$$f(x) = x_i \wedge f_{x_i=1} \oplus \neg x_i \wedge f_{x_i=0}. \quad (1.4)$$

1.3. Формы представления булевых функций

Наиболее естественной формой представления булевой функции $f(x)$ является *таблица истинности*. Табл.1.1 дает пример булевой функции, представленной таблицей истинности. Строки таблицы соответствуют различным наборам значений аргументов, число которых равно 2^n . Первые n столбцов таблицы соответствуют аргументам. Столбец $n+1$ соответствует функции.

Разложение Шеннона (1.1) позволяет *разбить* одну таблицу на две, представляющие кофакторы по переменной x_i . Так табл.1.1 разбивается по переменной x_1 на две части, помещенные в табл.1.2. Верхняя часть описывает кофактор $f_{x_1=0}$, нижняя часть – кофактор $f_{x_1=1}$.

Если для каждой строки таблицы, в которой булева функция f принимает значение 1, построить минтерм и все минтермы соединить операцией дизъюнкция, то получим *совершенную дизъюнктивную нормальную форму* (СДНФ) булевой функции f .

$$\text{СДНФ}(f) = \bigvee_{\substack{a \in B^n, \\ f(a)=1}} \bigwedge_{i=1, \dots, n} x_i^{a_i}. \quad (1.5)$$

Так в соответствии с (1.5) по табл.1.1 получаем

$$\text{СДНФ}(f) = \neg x_1 \wedge x_2 \wedge \neg x_3 \vee x_1 \wedge \neg x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge x_3.$$

Таблица 1.1

Пример булевой функции

| x_1 | x_2 | x_3 | Функция f |
|-------|-------|-------|-------------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Таблица 1.2

Разбиение одной таблицы на две

| | x_2 | x_3 | Кофакторы |
|-------------|-------|-------|-----------|
| $f_{x_1=0}$ | 0 | 0 | 0 |
| | 0 | 1 | 0 |
| | 1 | 0 | 1 |
| | 1 | 1 | 0 |
| $f_{x_1=1}$ | 0 | 0 | 0 |
| | 0 | 1 | 1 |
| | 1 | 0 | 0 |
| | 1 | 1 | 1 |

Если для каждой строки таблицы, в которой булева функция f принимает значение 0, построить дизъюнкт и все дизъюнкты соединить операцией конъюнкция, то получим *совершенную конъюнктивную нормальную форму* (СКНФ) булевой функции f .

$$СКНФ(f) = \bigwedge_{\substack{a \in B^n, \\ f(a)=0}} \bigvee_{i=1, \dots, n} x_i^{-a_i}. \quad (1.6)$$

Так в соответствии с (1.6) по табл.1.1 получаем

$$СКНФ(f) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3).$$

В формуле (1.5) операция \vee может быть заменена операцией \oplus . В результате получаем *совершенную нормальную форму по исключающему или* (СИНФ). Так по табл.1.1 получаем

$$СИНФ(f) = \neg x_1 \wedge x_2 \wedge \neg x_3 \oplus x_1 \wedge \neg x_2 \wedge x_3 \oplus x_1 \wedge x_2 \wedge x_3.$$

СИНФ не единственная форма, построенная на операции исключающее или. *Форма Риды-Маллера положительной полярности* (пФРМ) или полином Жегалкина строится путем многократного применения положительного разложения Давио по всем переменным x_1, \dots, x_n . Так для функции, представленной табл.1.1,

$$пФРМ(f) = x_2 \oplus x_1 \wedge x_2 \oplus x_1 \wedge x_3 \oplus x_2 \wedge x_3 \oplus x_1 \wedge x_2 \wedge x_3.$$

Форма Риды-Маллера отрицательной полярности (оФРМ) строится путем многократного применения отрицательного разло-

жения Давио по всем переменным x_1, \dots, x_n . Так для функции, представленной табл.1.1,

$$\text{оФРМ}(f) = 1 \oplus \neg x_1 \oplus \neg x_3 \oplus \neg x_1 \neg x_2 \wedge \neg x_3.$$

Форма Рида-Маллера фиксированной полярности (фФРМ) строится путем многократного применения положительного и отрицательного разложений Давио с тем ограничением, что каждая переменная имеет одну фиксированную полярность. Так для функции, представленной табл.1.1,

$$\text{фФРМ}(f) = \neg x_1 \oplus x_3 \oplus \neg x_1 \wedge \neg x_2 \oplus \neg x_1 \wedge \neg x_2 \wedge x_3,$$

при условии, что переменным x_1, x_2 назначена отрицательная полярность, переменной x_3 – положительная полярность в разложении Давио.

Форма Кронекер (ФКРО) строится путем многократного применения положительного и отрицательного разложений Давио, а также разложения Шеннона (1.4) с тем ограничением, что каждая переменная использует один фиксированный тип разложения. Так для функции, представленной табл.1.1,

$$\text{ФКРО}(f) = x_1 \oplus x_1 \wedge \neg x_3 \oplus \neg x_1 \wedge x_2 \wedge \neg x_3,$$

при условии, что переменная x_1 использует разложение Шеннона, переменная x_2 - положительное разложение Давио, переменная x_3 – отрицательное разложение Давио.

Форма псевдо Кронекер (ФПСКРО) строится путем многократного применения в произвольном сочетании и без привязки к переменным положительного и отрицательного разложений Давио, а также разложения Шеннона.

В последние годы широкое распространение получили графовые формы представления булевых функций. Наиболее успешной является *сокращенная упорядоченная диаграмма двоичных решений* (СУДДР). Она строится на базе разложения Шеннона и является каноническим представлением функции. Пример СУДДР для нашей демонстрационной функции дан на рис.1.1. Граф является ориентированным ациклическим с одним корнем, двумя листьями 0 и 1, четырьмя нетерминальными вершинами, помеченными тремя переменными x_1, x_2, x_3 . Граф не содержит одинаковых подграфов.

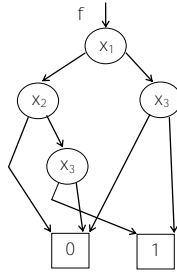


Рисунок 1.1 – Пример СУДП

1.4. Не полностью определенные булевы функции

Большой интерес представляют *не полностью определенные булевы функции*, рассматриваемые как отображение $\varphi: C \rightarrow B$, где множество $C \subset B^n$. Не полностью определенная булева функция $\varphi(x)$ может рассматриваться также как отображение $\varphi: B^n \rightarrow M$, где $M = \{0, 1, dc\}$ – множество *частичных значений*. Значение dc интерпретируется как *don't care* – безразличное или неопределенное значение. Всего существует 3^{2^n} не полностью определенных функций, причем булевы функции являются частным случаем не полностью определенных функций. Не полностью определенные булевы функции могут быть представлены посредством полностью определенных булевых функций из $B_{n,1}$.

Множество входных наборов $a = (a_1, \dots, a_n) \in B^n$, на которых $\varphi(a) = 0$, называется *множеством исключения*, обозначаемым $OFF(\varphi)$. Аналогичным образом определяется *множество включения* $ON(\varphi)$. Соответствующие характеристические функции этих множеств обозначаются f_{OFF} и f_{ON} . Они являются полностью определенными, поскольку для всех a выполняется $f_{OFF}(a) = 1$ тогда и только тогда, когда выполняется $\varphi(a) = 0$. Аналогично, для всех a выполняется $f_{ON}(a) = 1$ тогда и только тогда, когда выполняется $\varphi(a) = 1$.

В случае, когда φ не полностью специфицирована, множество $B^{\wedge} C$ представляет *множество безразличия* (неопределенности). Оно обозначается $DC(\varphi)$ и определяет функцию f_{DC} . Безразличное (неопределенное) значение обозначается dc , при этом $\varphi(a) = dc$, если $a \in DC(\varphi)$. Для задания не полностью определенной булевой функ-

ции φ достаточно использовать любые две из трех функций f_{OFF} , f_{ON} , f_{DC} . Функция $f_C = f_{OFF} \vee f_{ON}$ называется *функцией области определенности*. Она принимает значение 1 на тех входных наборах a , на которых $\varphi(a)=0$ или $\varphi(a)=1$. На наборах из $DC(\varphi)$ функция f_C принимает значение 0.

Замечательным свойством не полностью определенной функции φ является возможность ее *доопределения* до полностью определенной булевой функции f без нарушения поведения системы, которое эта функция описывает. Доопределение f еще называют реализацией функции φ . Если на наборе a имеет место $\varphi(a)=dc$, то доопределение дает $\varphi(a)=0$ либо $\varphi(a)=1$. В результате множества $OFF(\varphi) \cup DC^*(\varphi)$ и $ON(\varphi) \cup DC^*(\varphi)$ расширяются по сравнению с множествами $OFF(\varphi)$ и $ON(\varphi)$ за счет множества $DC(\varphi)$. На наборе $a \in DC^*(\varphi)$ значение $f(a)=0$, на наборе $a \in DC^*(\varphi)$ значение $f(a)=1$. Легко видеть, существует $2^{|DC(\varphi)|}$ доопределений функции φ до функции f . Поскольку параметры реализации функции φ варьируются при переходе от одной f к другой, множество доопределений является пространством поиска при решении задач оптимизации.

Пример не полностью определенной функции φ дан в табл.1.3. Один из вариантов доопределений f функции φ описан последним столбцом этой же таблицы.

1.5. Минимизация не полностью определенных булевых функций картами Карно

Карта Карно есть таблица истинности булевой функции, перестроенная таким образом, чтобы *соседние наборы* значений аргументов представлялись соседними клетками. Наборы называются соседними, если они отличаются по значению только одной переменной. Карта Карно – плоская развертка n -мерного булева куба.

Полный куб включает 2^n точек. Одной клетке карты Карно соответствует одна точка куба. Двум соседним клеткам соответствует 1-мерный куб, четырем соседним клеткам – 2-мерный куб и т.д. Полный n -мерный куб описывается константой 1. Куб мерности k описывается конъюнкцией $n-k$ литералов. Нахождение кратчайшего представления не полностью определенной булевой функции

Таблица 1.3

Пример не полностью определенной булевой функции

| x_1 | x_2 | x_3 | Функция f | Доопределение f |
|-------|-------|-------|-------------|-------------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | dc | 1 |
| 1 | 0 | 0 | dc | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

| | | x_2 | | | |
|-------|---|-------|----|---|----|
| | | 0 | 1 | 1 | dc |
| x_1 | 0 | 0 | 1 | 1 | dc |
| | 1 | 1 | dc | 0 | 0 |
| | | x_3 | | | |

Рисунок 1.2 – Карта Карно не полностью определенной функции f

f сводится к поиску минимального числа кубов на карте Карно, покрывающих все клетки со значением 1 и не покрывающих клетки со значением 0, при этом клетки со значением dc могут как покрываться так и не покрываться кубами. *Кратчайшая дизъюнктивная нормальная форма* КрДНФ строится в виде дизъюнкции конъюнкций, соответствующих найденным кубам.

Не полностью определенная функция f , описанная табл.1.3, представляется картой Карно, показанной на рис.1.2. Двум строкам таблицы соответствует переменная x_1 , причем для первой строки, отмеченной линией, $x_1=1$, для второй строки $x_1=0$. Четырем столбцам таблицы соответствуют переменные x_2, x_3 . Как отмечено линией сверху, для первых двух столбцов $x_2=1$, для последних двух – $x_2=0$. Как отмечено линией снизу, для средних столбцов $x_3=1$, для крайних – $x_3=0$. Заливкой показаны клетки со значениями 1 и dc , покрываемые двумя кубами. Первый куб включает средние клетки первой строки. Второй куб включает левые клетки второй строки. В результате, $\text{КрДНФ}(f) = x_1 \wedge x_3 \vee \neg x_1 \wedge x_2$.

1.6. Двухуровневая минимизация не полностью определенных булевых функций методом Квайна-МакКласки

Первоначально метод Квайна-МакКласки был разработан для полностью определенных булевых функций. Затем метод был расширен на не полностью определенные функции. Исходная минимизируемая не полностью определенная булева функция f представляется таблицей истинности. Метод включает три этапа:

- доопределение функции f до булевой функции f путем замены всех значений dc на значение 1;
- построение *сокращенной дизъюнктивной нормальной формы* (СкДНФ) по исходной совершенной дизъюнктивной нормальной форме СДНФ, описываемой доопределенной таблицей истинности;
- построение *матрицы Квайна* и нахождение минимального покрытия конституент 1 функции f простыми импликантами СкДНФ.

Продемонстрируем работу метода на примере функции f из табл.1.3. Множество конституент 1 функции f , полученной при доопределении функции f на первом этапе, разбивается на три *группы* по числу элементов 1 в конституенте, как показано в табл.1.4 для шага 0. К наборам из соседних групп применяется операция *склеивания*. Используя наборы, отличающиеся только по одной переменной, операция склеивания строит новый набор, в котором на месте этой переменной ставится символ *, а другие элементы набора наследуют соответствующие элементы исходных наборов. Так набор 010 из группы 1 отличается только по переменной x_3 от набора 011 из соседней группы 2. Операция склеивает наборы и получает новый набор 01*, который называется *импликантой* и помещается в группу 1 шага 1. Новый набор *поглощает* исходные наборы. Склеивание пяти наборов из трех групп шага 0 дает четыре импликанты, помещаемые в две группы шага 1. Все пять исходных наборов участвовали в склеивании, поэтому все они являются поглощенными.

На шаге 1 два набора из группы 1 не склеиваются с двумя наборами из группы 2. Отсутствие новых наборов делает все четыре набора не поглощенными. Не поглощенные наборы называются *простыми импликантами*, мы отмечаем их подчеркиванием.

Таблица 1.4

Построение СкДНФ

| Шаг | Группа | Наборы, импликанты |
|-----|--------|-------------------------|
| 0 | 1 | 010, 100, |
| | 2 | 011, 101 |
| | 3 | 111 |
| 1 | 1 | <u>01*</u> , <u>10*</u> |
| | 2 | <u>*11</u> , <u>1*1</u> |

Отсутствие наборов на шаге 2 завершает процесс склеивания и поглощения. Сокращенная ДНФ формируется из 4 простых импликант. Каждой импликанте a ставится в соответствие элементарная конъюнкция, включающая положительный литерал x_i , если $a_i=1$, включающая, отрицательный литерал $\neg x_i$, если $a_i=0$, и не включающая переменную x_i вовсе, если $a_i=*$. По таблице 1.4 получаем:

$$\text{СкДНФ}(f) = \neg x_1 \wedge x_2 \vee x_1 \wedge \neg x_2 \vee x_2 \wedge x_3 \vee x_1 \wedge x_3.$$

СкДНФ является *избыточной*, поэтому переходим к третьему этапу минимизации. Здесь выполняется построение *матрицы Квайна* Q , строки которой соответствуют простым импликантам, столбцы соответствуют конstituентам 1 не полностью определенной функции f (но не функции \tilde{f}). Элемент q_{ij} матрицы помечается символом \times , если конstituента j участвовала в склеивании при получении простой импликанты i . Матрица Квайна для нашего примера представлена табл.1.5. Подмножество строк-импликант таких, что для каждого столбца-конstituенты найдется в подмножестве строка, в которой стоит символ \times на пересечении со столбцом-конституентом, называется *покрытием* S матрицы Q . Покрытие S называется *безыбыточным*, если удаление из него хотя бы одной строки делает S не покрытием.

Таблица 1.5

Матрица Квайна Q

| № | Импликанты | Конституенты 1 | | |
|---|------------|----------------|----------|----------|
| | | 010 | 101 | 111 |
| 1 | 01* | \times | | |
| 2 | 10* | | \times | |
| 3 | *11 | | | \times |
| 4 | 1*1 | | \times | \times |

Ядро покрытия включает импликанты, для каждой из которых найдется конституента, покрываемая только этой строкой. В табл.1.5 заливкой показана импликанта, входящая в ядро. С увеличением размера матрицы \mathcal{Q} число различных покрытий \mathcal{C} растет экспоненциально.

По одному безызбыточному покрытию строится одна тупиковая ДНФ. Среди всех тупиковых ДНФ, та из них, которая построена из минимального числа простых импликант, называется кратчайшей (КрДНФ). ДНФ называется минимальной (МДНФ), если она построена из простых импликант, суммарное число литералов в которых минимально. Для матрицы \mathcal{Q} , представленной табл.1.5, существуют два безызбыточных покрытия: $\{01^*, 10^*, *11\}$, $\{01^*, 1^*1\}$. Второе покрытие является и кратчайшим и минимальным:

$$\text{КрДНФ}(f) = \text{МДНФ}(f) = \neg x_1 \wedge x_2 \vee x_1 \wedge x_3.$$

Все изложенное может быть применено к построению кратчайших и минимальных конъюнктивных нормальных форм не полностью определенных булевых функций.

1.7. *Историческая справка*

Булева алгебра предложена Джорджем Булем [9 (1853)] в его работах по формализации законов мышления. Им введены истинностные значения, логические переменные и операции, исследованы законы математической логики. Разложения полностью определенных булевых функций исследованы в работах К. Шеннона [79 (1949)], Р. Ашенхерста [2 (1957)], Д. Рота и Р. Карпа [78 (1962)], М. Давио, Д. Дешампа и А. Тэйси [22 (1978)] и других ученых [5, 23, 24, 35]. В работах И.И. Жегалкина [90 (1927)], И Риди и Д. Маллера [42, 74 (1954)] предложено полиномиальное представление булевых функций на базе операции исключающее или.

Идею оперирования неопределенностью в метаматематике и логических системах впервые предложил С.К. Клини [38 (1952)]. Карты Карно были изобретены Э.В. Вейчем и усовершенствованы М. Карно [34 (1953)]. Метод двухуровневой минимизации булевых функций разработан Квайном [72 (1966)] и МакКласки [43 (1965)]. Эвристические методы минимизации исследованы в работах

[7, 11, 31, 34, 88, 92]. Исследования по минимизации не полностью определенных булевых функций в классе дизъюнктивных нормальных форм выполнены в работах [31, 67, 88]. В работах [22, 24, 89, 109] предложены алгоритмы минимизации не полностью определенных булевых функций в полиномиальном представлении.

Уже в ранних работах по переключательным функциям и проектированию логических схем было замечено, что одни части схемы влияют на другие части тем, что другие части становятся не полностью определенными. Понятие наблюдаемости неопределенности исследовалось в ряде работ [4, 14, 17-19, 23, 29, 32, 33, 45, 46, 79, 86, 87, 109] с целью проектирования, тестирования и оптимизации логических схем. В работе [6] неопределенность в спецификации проекта использована в высокоуровневом синтезе цифровых систем.

Глава 2. ЧАСТИЧНЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ. ЧАСТИЧНАЯ АЛГЕБРА

2.1. Частичные логические функции

Построение не полностью определенной функции $\phi(x)$ базируется на использовании двух множеств: множества $B=\{0, 1\}$ и множества $M=\{0, 1, dc\}$. Значение dc интерпретируется как *don't care* – безразличное или неопределенное значение. Множество B используется для построения области определения, множество M – для построения области значений функции $\phi(x)$. Если $\phi(x)$ и $\varphi(x)$ – две не полностью определенные скалярные функции, зависящие от векторного аргумента $x=(x_1, \dots, x_n)$, то построение суперпозиции $\phi(x_1, \dots, x_{i-1}, \varphi(x), x_{i+1}, \dots, x_n)$ не представляется возможным из-за различий в области определения и области значений. Как следствие, одни не полностью определенные функции не могут быть выражены в базисе других не полностью определенных функций.

В *частичной алгебре* множество M одновременно используется для построения области определения и области значений функции. Скалярная *частичная переменная* y_i принимает значение из M . *Векторная частичная переменная* $y=(y_1, \dots, y_n)$ является набором скалярных частичных переменных. Скалярная *частичная функция* $\Phi(y)$ или *функция Клини* есть отображение $\Phi: M^n \rightarrow M$. Всего существует 3^{3^n} частичных булевых функций, причем не полностью определенные функции являются частным случаем частичных. Если $\Phi(y)$ и $\Psi(y)$ – две частичные скалярные функции, зависящие от частичной векторной переменной $y=(y_1, \dots, y_n)$, то суперпозиция $\Phi(y_1, \dots, y_{i-1}, \Psi(y), y_{i+1}, \dots, y_n)$ корректно описывает третью частичную функцию. В результате получаем механизм описания произвольных частичных функций в базисе ограниченного числа простых одноместных и двухместных частичных операций.

Пусть $a=(a_1, \dots, a_n)$ – набор значений частичных переменных y_1, \dots, y_n , где $a_i \in M$. Набор a называется *конституентой* 0, 1 или dc частичной функции $\Phi(a)$, если $\Phi(a)=0$, $\Phi(a)=1$ или $\Phi(a)=dc$ соответственно. Наборы a и b называются *ортогональными*, если найдется такое $i \in N$, что имеет место один из двух случаев: $a_i=1$, $b_i=0$ или

$a_i=0, b_i=1$. Частичная функция $\Phi(y)$ является *корректно определенной*, если для любых двух наборов a и b , на которых $\Phi(a)=1, \Phi(b)=0$ либо $\Phi(a)=0, \Phi(b)=1$, выполняется условие их ортогональности. В противном случае частичная функция определена некорректно. Не корректно определенной является функция $\Psi(y)$ из табл.2.1. Наборы $(0,0)$ и $(0,dc)$ не являются ортогональными, в то же время значения функции $\Psi(0,0)=0$, а $\Psi(0,dc)=1$.

Наборы a и b называются *склеиваемыми* в функции $\Phi(y)$, если $\Phi(a)=\Phi(b)$ и наборы удовлетворяют двум требованиям:

- найдется k такое, что имеет место один из двух случаев $a_k=1, b_k=0$ или $a_k=0, b_k=1$;
- $a_i=b_i$ при $i=1, \dots, n, i \neq k$.

Результатом склеивания является набор $c=(a_1, \dots, a_{k-1}, dc, a_{k+1}, \dots, a_n)$. Наборы a и b являются *склеенными*, если $\Phi(c)=\Phi(a)$. Если $\Phi(c)=dc$, то *склеивание* наборов выполняется заменой значения $\Phi(c)=dc$ на значение $\Phi(c)=\Phi(a)$. Частичная функция $\Psi(y)$ получается из частичной функции $\Phi(y)$ склеиванием всех склеиваемых наборов a и b с заменой значения $\Phi(c)=dc$ на значение $\Psi(c)=\Phi(a)$.

Набор a называется *развертываемым* для корректной частичной функции $\Phi(y)$, если $\Phi(a)=1$ или $\Phi(a)=0$ и для всех наборов b , получаемых из набора a заменой всех значений $a_i=dc$ на значение 0 или 1 в произвольных комбинациях, найдется такой, что функция имеет значение $\Phi(b)=dc$. Развернутая частичная функция $\Psi(y)$ получается из частичной функции $\Phi(y)$ *развертыванием* всех развертываемых наборов a , заключающимся в замене значения $\Phi(b)=dc$ на значение $\Psi(b)=\Phi(a)$.

Частичная функция называется *склеено-развернутой*, если в ней отсутствуют как склеиваемые, так и развертываемые наборы. Примеры не склеенных, не развернутых, склеенных и развернутых частичных функций приведены в табл.2.1. Так функция $\Phi_1(y)$ является не склеенной, но развернутой. В частности, наборы $(1,0), (1,1)$ являются склеиваемыми, поскольку функция $\Phi_1(y)$ принимает значение 1 на этих наборах, а на результате склеивания, т.е. наборе $(1,dc)$, функция принимает значение dc . При замене $\Phi_1(1,dc)=dc$ на $\Phi_1(1,dc)=1$ наборы становятся склеенными.

Функция $\Phi_2(y)$ является склеенной, но не развернутой. Так, на наборе $(1,dc)$ значение функции $\Phi_2(1,dc)=1$. На наборах $(1,0), (1,1)$

Таблица 2.1

Примеры частичных функций

| y_1 | y_2 | $\Psi(y)$ | $\Phi_1(y)$ | $\Phi_2(y)$ | $\Phi_3(y)$ |
|-------|-------|-----------|-------------|-------------|-------------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | dc | 1 |
| 0 | dc | 1 | dc | dc | dc |
| 1 | 0 | 1 | 1 | dc | 1 |
| 1 | 1 | 1 | 1 | dc | 1 |
| 1 | dc | dc | dc | 1 | 1 |
| dc | 0 | dc | dc | dc | dc |
| dc | 1 | 0 | dc | 1 | 1 |
| dc | dc | dc | dc | dc | dc |

значение функции равно dc . Набор $(1, dc)$ является развертываемым. Развертывание приводит к замене значения функции с dc на 1 на наборах $(1, 0)$, $(1, 1)$.

Функция $\Phi_3(y)$ является и склеенной и развернутой. В ней нет ни склеиваемых, ни развертываемых наборов. Легко видеть, что функция $\Phi_1(y)$ склеивается, а функция $\Phi_2(y)$ развертывается в функцию $\Phi_3(y)$.

Из возможности замены значения dc на значение 0 или 1 следует возможность сведения любой частичной функции $\Phi(y)$ к соответствующей не полностью определенной функции $\phi(x)$, в которой булевы переменные x_1, \dots, x_n заменяют частичные переменные y_1, \dots, y_n . Сначала $\Phi(y)$ трансформируется в склеено-развернутую частичную функцию $\Psi(y)$. Затем выбираются наборы a , являющиеся значениями переменной y и состоящие только из нулей и единиц, а значение частичной функции $\Psi(a)$ принимается в качестве значения $\phi(a)$ соответствующей не полностью определенной функции, где a является также значением булевой переменной x . Все три частичные функции $\Phi_1(y)$, $\Phi_2(y)$, $\Phi_3(y)$, определенные в табл.2.1, сводятся к одной не полностью (в данном примере полностью) определенной функции двухместная дизъюнкция.

Частичные функции, зависящих от n переменных и сводящиеся (преобразующиеся) к одной и той же не полностью определенной функции, называются *T-эквивалентными*. Все множество частичных функций разбивается на непересекающиеся подмножества или *классы T-эквивалентных частичных функций*. Если Π - частичная функция, то $\pi = T(\Pi)$ – соответствующая не полностью определенная

функция, к которой сводится функция П. Если π – не полностью определенная функция, то $T^{-1}(\pi) = \{\Pi_1, \dots, \Pi_k\}$ – подмножество частичных функций, сводящихся или преобразующихся к функции π . Функции из множества $T^{-1}(\pi)$ являются T-эквивалентными. Так, функции $\Phi_1(y)$, $\Phi_2(y)$, $\Phi_3(y)$ из табл.2.1 относятся к одному классу T-эквивалентных частичных функций.

2.2. Одноместные частичные операции

По аналогии с булевой алгеброй частичная алгебра строится, прежде всего, путем создания базиса из одноместных и двухместных *частичных операций*. *Одноместная* частичная операция $\Phi(y_1)$ зависит от скалярной частичной переменной y_1 , принимающей значения 0, 1 и *dc*. Число одноместных частичных операций, у которых область определения, так же как и область значений, включает три элемента, равно $3^3 = 27$. Число не полностью определенных одноместных операций равно $3^{2^1} = 9$, булевых операций – $2^{2^1} = 4$.

Все возможные одноместные частичные операции определены в табл.2.2. Темной заливкой показаны 10 не корректных операций, остальные 17 операций – корректные. Для некорректных операций существуют не ортогональные значения аргумента, на которых операция принимает различные значения 0 и 1. Без заливки показаны 12 частичных операций, сводящихся к полностью определенным булевым операциям. Светлой заливкой показаны 5 частичных операций, сводящихся к не полностью определенным булевым операциям, принимающим значение *dc* хотя бы на одном значении переменной. Не развернутыми являются 6 операций, не склеенными – 2 операции. Склеено-развернутыми являются 9 операций. Одновременно не склеенные и не развернутые операции отсутствуют.

В табл.2.3 описаны 9 не полностью определенных одноместных булевых операций. Среди них 4 операции (без заливки) являются полностью определенными, 5 операций (светлая заливка) принимают значение *dc*. Классы T-эквивалентности, соответствующие константам 0 и 1, включают по 5 частичных операций, что объясняется несущественностью переменной x_1 (константы 0 и 1 являются

Таблица 2.2

Одноместные частичные операции

| № | Переменная y_1 | | | Обозначение операции | Имя операции | Развернутая | Склеенная |
|----|------------------|----|----|----------------------|--------------|-------------|-----------|
| | 0 | 1 | dc | | | | |
| 1 | 0 | 0 | 0 | 0 | Константа 0 | да | да |
| 2 | 0 | 0 | 1 | | - | | |
| 3 | 0 | 0 | dc | 0 | Константа 0 | да | нет |
| 4 | 0 | 1 | 0 | | - | | |
| 5 | 0 | 1 | 1 | | - | | |
| 6 | 0 | 1 | dc | y_1 | Идентичность | да | да |
| 7 | 0 | dc | 0 | 0 | Константа 0 | нет | да |
| 8 | 0 | dc | 1 | | - | | |
| 9 | 0 | dc | dc | | н.п.о. | да | да |
| 10 | 1 | 0 | 0 | | - | | |
| 11 | 1 | 0 | 1 | | - | | |
| 12 | 1 | 0 | dc | $\sim y_1$ | Инверсия | да | да |
| 13 | 1 | 1 | 0 | | - | | |
| 14 | 1 | 1 | 1 | 1 | Константа 1 | да | да |
| 15 | 1 | 1 | dc | 1 | Константа 1 | да | нет |
| 16 | 1 | dc | 0 | | - | | |
| 17 | 1 | dc | 1 | 1 | Константа 1 | нет | да |
| 18 | 1 | dc | dc | | н.п.о. | да | да |
| 19 | dc | 0 | 0 | 0 | Константа 0 | нет | да |
| 20 | dc | 0 | 1 | | - | | |
| 21 | dc | 0 | dc | | н.п.о. | да | да |
| 22 | dc | 1 | 0 | | - | | |
| 23 | dc | 1 | 1 | 1 | Константа 1 | нет | да |
| 24 | dc | 1 | dc | | н.п.о. | да | да |
| 25 | dc | dc | 0 | 0 | Константа 0 | нет | да |
| 26 | dc | dc | 1 | 1 (#) | Константа 1 | нет | да |
| 27 | dc | dc | dc | | н.п.о. | да | да |

по существу 0-местными операциями). Остальные классы включают по одной частичной операции.

Частичные склеено-развернутые операции *константа 0*, *константа 1*, *константа dc*, *идентичность* и *отрицание* с номерами 1, 14, 27, 6 и 12 в табл.2.2, которым соответствуют не полностью определенные операции с номерами 1, 5, 9, 2 и 4 соответственно в

Таблица 2.3

Сведение частичных операций к не полностью определенным операциям

| № функции | Значения булевой переменной x_1 | | Обозначение не полностью определенной операции | Имя не полностью определенной операции | Классы T -эквивалентных одноместных частичных операций |
|-----------|-----------------------------------|----|--|--|--|
| | 0 | 1 | | | |
| 1 | 0 | 0 | 0 | Константа 0 | 1, 3, 7, 19, 25 |
| 2 | 0 | 1 | x_1 | Идентичность | 6 |
| 3 | 0 | dc | | | 9 |
| 4 | 1 | 0 | $\neg x_1$ | Отрицание | 12 |
| 5 | 1 | 1 | 1 | Константа 1 | 14, 15, 17, 23, 26 |
| 6 | 1 | dc | | | 18 |
| 7 | dc | 0 | | | 21 |
| 8 | dc | 1 | | | 24 |
| 9 | dc | dc | dc | Константа dc | 27 |

табл.2.3, выберем в качестве *базовых одноместных частичных операций*. В качестве базовой будем также рассматривать операцию с номером 26 в табл.2.2, которую назовем *частичной операцией идентификации неопределенности* и обозначим #. Назначение этой операции будет показано ниже.

2.3. Двухместные частичные операции

Двухместная частичная операция $\Phi(y)$ зависит от векторного аргумента y , состоящего из двух скалярных частичных переменных $y=(y_1, y_2)$, каждая из которых принимает значения из множества M . Вектор y принимает наборы (пары) значений из множества M^2 . Число таких наборов равно девяти. Поскольку любая операция $\Phi^2(y)$ также принимает значение их множества M , общее число двухместных частичных операций равно $3^{3^2}=19683$ (табл.2.4). Число не полностью определенных двухместных операций равно $3^{2^2}=81$, булевых операций – $2^{2^2}=16$. Для исследования операций была написана компьютерная программа, которая дала следующие результаты.

Таблица 2.4

Частичные двухместные операции

| № операции | Значения частичных переменных y_1 и y_2 | | | | | | | | |
|------------|---|----|----|----|----|----|----|----|----|
| | 0 | 0 | 0 | 1 | 1 | 1 | dc | dc | dc |
| | 0 | 1 | dc | 0 | 1 | dc | 0 | 1 | dc |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | dc |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ... | | | | | | | | | |
| 19683 | dc | dc | dc | dc | dc | dc | dc | dc | dc |

Как и одноместные частичные операции, двухместные операции могут быть корректно и некорректно определенными. В табл.2.4 операции с номерами 1, 3 определены корректно, операции с номерами 2, 4 определены некорректно.

Среди 19683 двухместных частичных операций 1361 операция является корректной (6.9%) и 18322 операций являются некорректными (93.1%). К 16 полностью определенным операциям сводятся 1072 корректные частичные операции. Остальные 289 корректных частичных операций сводятся к 65 не полностью определенным операциям, принимающим значение *dc* на некоторых наборах значений переменных. Не склеенными являются 636 частичных операций, склеенными – 725. Не развернутыми являются 1164 операции, развернутыми - 197. Одновременно не склеенными и не развернутыми являются 520 операций, 81 операция является склеено-развернутой. Одной не полностью определенной операции соответствует ровно одна склеено-развернутая частичная операция.

Табл.2.5 описывает 81 класс *T*-эквивалентных двухместных частичных операций. Из них 16 классов соответствуют 16 булевым полностью определенным операциям, описанным строками без заливки. Остальные 65 операций, описанных строками с заливкой, являются не полностью определенными, принимающими значение *dc* на некоторых наборах значений аргументов. Столбец $\alpha(k)$ описывает общее число частичных операций, сводящихся к не полностью определенной операции *k*, столбец $\beta(k)$ - число склеенных, а столбец $\gamma(k)$ - число развернутых частичных операций.

Таблица 2.5

Не полностью определенные двухместные операции

| № | Значения булевых переменных x_1, x_2 | | | | Обозначение | Имя операции | α | β | γ |
|----|--|----|----|----|-----------------------------|-----------------|----------|---------|----------|
| | 00 | 01 | 10 | 11 | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | Константа 0 | 417 | 17 | 169 |
| 2 | 0 | 0 | 0 | 1 | $x_1 \wedge x_2$ | Конъюнкция | 17 | 4 | 12 |
| 3 | 0 | 0 | 0 | dc | | | 17 | 4 | 12 |
| 4 | 0 | 0 | 1 | 0 | $\neg(x_1 \rightarrow x_2)$ | Отр. импликации | 17 | 4 | 12 |
| 5 | 0 | 0 | 1 | 1 | x_1 | Идентичность | 25 | 4 | 16 |
| 6 | 0 | 0 | 1 | dc | | | 5 | 2 | 4 |
| 7 | 0 | 0 | dc | 0 | | | 17 | 4 | 12 |
| 8 | 0 | 0 | dc | 1 | | | 5 | 2 | 4 |
| 9 | 0 | 0 | dc | dc | | | 5 | 2 | 4 |
| 10 | 0 | 1 | 0 | 0 | $\neg(x_1 \leftarrow x_2)$ | Отр. импликации | 17 | 4 | 12 |
| 11 | 0 | 1 | 0 | 1 | x_2 | Идентичность | 25 | 4 | 16 |
| 12 | 0 | 1 | 0 | dc | | | 5 | 2 | 4 |
| 13 | 0 | 1 | 1 | 0 | $x_1 \oplus x_2$ | Исключающее или | 1 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | $x_1 \vee x_2$ | Дизъюнкция | 17 | 4 | 12 |
| 15 | 0 | 1 | 1 | dc | | | 1 | 1 | 1 |
| 16 | 0 | 1 | dc | 0 | | | 1 | 1 | 1 |
| 17 | 0 | 1 | dc | 1 | | | 5 | 2 | 4 |
| 18 | 0 | 1 | dc | dc | | | 1 | 1 | 1 |
| 19 | 0 | dc | 0 | 0 | | | 17 | 4 | 12 |
| 20 | 0 | dc | 0 | 1 | | | 5 | 2 | 4 |
| 21 | 0 | dc | 0 | dc | | | 5 | 2 | 4 |
| 22 | 0 | dc | 1 | 0 | | | 1 | 1 | 1 |
| 23 | 0 | dc | 1 | 1 | | | 5 | 2 | 4 |
| 24 | 0 | dc | 1 | dc | | | 1 | 1 | 1 |
| 25 | 0 | dc | dc | 0 | | | 1 | 1 | 1 |
| 26 | 0 | dc | dc | 1 | | | 1 | 1 | 1 |
| 27 | 0 | dc | dc | dc | | | 1 | 1 | 1 |
| 28 | 1 | 0 | 0 | 0 | $x_1 \downarrow x_2$ | Стрелка Пирса | 17 | 4 | 12 |
| 29 | 1 | 0 | 0 | 1 | $x_1 \leftrightarrow x_2$ | Эквивалентность | 1 | 1 | 1 |
| 30 | 1 | 0 | 0 | dc | | | 1 | 1 | 1 |
| 31 | 1 | 0 | 1 | 0 | $\neg y_2$ | Отрицание | 25 | 4 | 16 |
| 32 | 1 | 0 | 1 | 1 | $x_1 \leftarrow x_2$ | Импликация | 17 | 4 | 12 |
| 33 | 1 | 0 | 1 | dc | | | 5 | 2 | 4 |
| 34 | 1 | 0 | dc | 0 | | | 5 | 2 | 4 |
| 35 | 1 | 0 | dc | 1 | | | 1 | 1 | 1 |
| 36 | 1 | 0 | dc | dc | | | 1 | 1 | 1 |
| 37 | 1 | 1 | 0 | 0 | $\neg x_1$ | Отрицание | 25 | 4 | 16 |
| 38 | 1 | 1 | 0 | 1 | $x_1 \rightarrow x_2$ | Импликация | 17 | 4 | 12 |

| | | | | | | | | | |
|----|----|----|----|----|-----------|----------------------|-----|----|-----|
| 39 | 1 | 1 | 0 | dc | | | 5 | 2 | 4 |
| 40 | 1 | 1 | 1 | 0 | x_1/x_2 | Штрих Шеффера | 17 | 4 | 12 |
| 41 | 1 | 1 | 1 | 1 | 1 | Константа 1 | 417 | 17 | 169 |
| 42 | 1 | 1 | 1 | dc | | | 17 | 4 | 12 |
| 43 | 1 | 1 | dc | 0 | | | 5 | 2 | 4 |
| 44 | 1 | 1 | dc | 1 | | | 17 | 4 | 12 |
| 45 | 1 | 1 | dc | dc | | | 5 | 2 | 4 |
| 46 | 1 | dc | 0 | 0 | | | 5 | 2 | 4 |
| 47 | 1 | dc | 0 | 1 | | | 1 | 1 | 1 |
| 48 | 1 | dc | 0 | dc | | | 1 | 1 | 1 |
| 49 | 1 | dc | 1 | 0 | | | 5 | 2 | 4 |
| 50 | 1 | dc | 1 | 1 | | | 17 | 4 | 12 |
| 51 | 1 | dc | 1 | dc | | | 5 | 2 | 4 |
| 52 | 1 | dc | dc | 0 | | | 1 | 1 | 1 |
| 53 | 1 | dc | dc | 1 | | | 1 | 1 | 1 |
| 54 | 1 | dc | dc | dc | | | 1 | 1 | 1 |
| 55 | dc | 0 | 0 | 0 | | | 17 | 4 | 12 |
| 56 | dc | 0 | 0 | 1 | | | 1 | 1 | 1 |
| 57 | dc | 0 | 0 | dc | | | 1 | 1 | 1 |
| 58 | dc | 0 | 1 | 0 | | | 5 | 2 | 4 |
| 59 | dc | 0 | 1 | 1 | | | 5 | 2 | 4 |
| 60 | dc | 0 | 1 | dc | | | 1 | 1 | 1 |
| 61 | dc | 0 | dc | 0 | | | 5 | 2 | 4 |
| 62 | dc | 0 | dc | 1 | | | 1 | 1 | 1 |
| 63 | dc | 0 | dc | dc | | | 1 | 1 | 1 |
| 64 | dc | 1 | 0 | 0 | | | 5 | 2 | 4 |
| 65 | dc | 1 | 0 | 1 | | | 5 | 2 | 4 |
| 66 | dc | 1 | 0 | dc | | | 1 | 1 | 1 |
| 67 | dc | 1 | 1 | 0 | | | 1 | 1 | 1 |
| 68 | dc | 1 | 1 | 1 | | | 17 | 4 | 12 |
| 69 | dc | 1 | 1 | dc | | | 1 | 1 | 1 |
| 70 | dc | 1 | dc | 0 | | | 1 | 1 | 1 |
| 71 | dc | 1 | dc | 1 | | | 5 | 2 | 4 |
| 72 | dc | 1 | dc | dc | | | 1 | 1 | 1 |
| 73 | dc | dc | 0 | 0 | | | 5 | 2 | 4 |
| 74 | dc | dc | 0 | 1 | | | 1 | 1 | 1 |
| 75 | dc | dc | 0 | dc | | | 1 | 1 | 1 |
| 76 | dc | dc | 1 | 0 | | | 1 | 1 | 1 |
| 77 | dc | dc | 1 | 1 | | | 5 | 2 | 4 |
| 78 | dc | dc | 1 | dc | | | 1 | 1 | 1 |
| 79 | dc | dc | dc | 0 | | | 1 | 1 | 1 |
| 80 | dc | dc | dc | 1 | | | 1 | 1 | 1 |
| 81 | dc | dc | dc | dc | dc | Константа dc | 1 | 1 | 1 |



Рисунок 2.1 – Количество частичных операций, сводящихся к одной не полностью определенной операции (точкам 1 и 41 горизонтальной оси соответствуют значения 417 вертикальной оси)

Число частичных операций, сводящихся к каждой из 81 не полностью определенной операции, описывается графиком, показанным на рис.2.1. Проанализируем сначала 65 классов T -эквивалентности, соответствующих чисто не полностью определенным операциям. Затем выполним анализ 16 классов, соответствующих полностью определенным операциям.

Для 8 не полностью определенных операций с номерами 3, 7, 19, 42, 44, 50, 55 и 68 значение $\alpha(k)=17$ является наибольшим, при этом $\beta(k)=4$ и $\gamma(k)=12$. На четырех входных наборах значений переменных эти операции принимают одно значение dc и три значения 0 либо 1. Для 24 операций $\alpha(k)=5$ при $\beta(k)=2$ и $\gamma(k)=4$. На четырех входных наборах значений переменных эти операции принимают два значения 0 (либо 1) и два значения dc либо одно значение dc и одно значение 1 (либо 0). Каждой из оставшихся 33 операций соответствует одна частичная склеено-развернутая операция.

В табл.2.6 приведены шестнадцать базовых частичных склеено-развернутых операций, обобщающих шестнадцать двухместных полностью определенных булевых операций и сводящихся к ним.

Таблица 2.6

Базовые двухместные частичные операции

| № операции | Значения частичных переменных y_1 и y_2 | | | | | | | | | Обозначение операции |
|------------|---|---|----|---|---|----|----|----|----|-----------------------------|
| | 0 | 0 | 0 | 1 | 1 | 1 | dc | dc | dc | |
| | 0 | 1 | dc | 0 | 1 | dc | 0 | 1 | dc | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | dc | 0 | dc | dc | $y_1 \& y_2$ |
| 3 | 0 | 1 | dc | 0 | 0 | 0 | 0 | dc | dc | $\sim(y_1 \Rightarrow y_2)$ |
| 4 | 0 | 0 | 0 | 1 | 1 | 1 | dc | dc | dc | y_1 |
| 5 | 0 | 0 | 0 | 1 | 0 | dc | dc | 0 | dc | $\sim(y_1 \Leftarrow y_2)$ |
| 6 | 0 | 1 | dc | 0 | 1 | dc | 0 | 1 | dc | y_2 |
| 7 | 0 | 1 | dc | 1 | 0 | dc | dc | dc | dc | $y_1 \odot y_2$ |
| 8 | 0 | 1 | dc | 1 | 1 | 1 | dc | 1 | dc | $y_1 + y_2$ |
| 9 | 1 | 0 | dc | 0 | 0 | 0 | dc | 0 | dc | $y_1 \oslash y_2$ |
| 10 | 1 | 0 | dc | 0 | 1 | dc | dc | dc | dc | $y_1 \Leftrightarrow y_2$ |
| 11 | 1 | 0 | dc | 1 | 0 | dc | 1 | 0 | dc | $\sim y_2$ |
| 12 | 1 | 1 | 1 | 0 | 1 | dc | dc | 1 | dc | $y_1 \Leftarrow y_2$ |
| 13 | 1 | 1 | 1 | 0 | 0 | 0 | dc | dc | dc | $\sim y_1$ |
| 14 | 1 | 0 | dc | 1 | 1 | 1 | 1 | dc | dc | $y_1 \Rightarrow y_2$ |
| 15 | 1 | 1 | 1 | 1 | 0 | dc | 1 | dc | dc | $y_1 \oslash y_2$ |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Имена частичных операций примем идентичными именам булевых операций, обозначения операций изменим: 1) константа 0; 2) конъюнкция (&); 3) отрицание прямой импликации; 4) идентичность; 5) отрицание обратной импликации; 6) идентичность; 7) исключающее или (\odot); 8) дизъюнкция (+); 9) стрелка Пирса (\oslash); 10) эквивалентность (\Leftrightarrow); 11) отрицание (\sim); 12) обратная импликация (\Leftarrow); 13) отрицание (\sim); 14) прямая импликация (\Rightarrow); 15) штрих Шеффера (\oslash); 16) константа 1.

Зависимость значения $\alpha(i)$ от номера i булевой операции представлена на рис.2.2. Для констант 0 и 1 значение $\alpha(i)=417$ является наибольшим. Причина состоит в том, что константы, являясь 0-местными операциями, используют две несущественные переменные. Вторым по величине значением $\alpha(i)=25$ обладают четыре 1-местные операции идентичности и отрицания, использующие по одной несущественной переменной.

Третьим значением $\alpha(i)=17$ обладают восемь по-настоящему двухместных операций, из которых главными четырьмя являются конъюнкция, дизъюнкция, прямая и обратная импликации, а четыре



Рисунок 2.2 – Количество $\alpha(i)$ частичных операций, приходящихся на одну булеву двухместную операцию i (точкам 1 и 16 горизонтальной оси соответствуют значения 417 вертикальной оси)

другие, включая штрих Шеффера и стрелку Пирса, являются инверсиями первых. Все восемь операций обладают одинаковыми возможностями оперирования неопределенностью.

Для операций исключаящее или и эквивалентность $\alpha(i)=1$. Удивительно, но у этих операций фактически полностью отсутствует возможность оперирования неопределенностью.

Зависимость значения $\beta(i)$ от номера булевой операции i дана на рис.2.3. Для констант 0 и 1 значение $\beta(i)=17$ является



Рисунок 2.3 – Количество склеенных частичных операций, приходящихся на одну булеву двухместную операцию



Рисунок 2.4 – Количество развернутых частичных операций, приходящихся на одну булеву двухместную операцию (точкам 1 и 16 горизонтальной оси соответствуют значения 169 вертикальной оси)

наибольшим. Для операций исключающее или и эквивалентность $\beta(i)=1$ является наименьшим. Для остальных 12 операций $\beta(i)=4$.

Зависимость значения $\gamma(i)$ от номера булевой операции i дана на рис.2.4. Для констант 0 и 1 значение $\gamma(i)=169$ является наибольшим. Для операций исключающее или и эквивалентность значение $\gamma(i)=1$ является наименьшим. Для четырех операций идентичность и отрицание $\gamma(i)=16$. Для остальных восьми операций $\gamma(i)=12$.

В итоге, для двухместных операций выполняются следующие закономерности. Число корректных частичных операций в 14.5 раза меньше общего числа частичных операций. В то же время число не полностью определенных операций в 16.8 раза меньше числа корректных частичных операций, а число полностью определенных операций в 5.06 раза меньше числа не полностью определенных операций и в 85.1 раза меньше числа корректных частичных операций. С ростом числа аргументов доля корректных частичных операций быстро сокращается, однако их абсолютное число быстро увеличивается по сравнению с числом не полностью и полностью определенных операций. Следовательно, возможность оперирования неопределенностью возрастает с увеличением числа аргументов.

Частичные операции константа 0 и константа 1 являются по существу 0-местными. Частичные операции идентичность и отри-

цание являются 1-местными. Остальные десять операций являются по-настоящему 2-местными. Примем их в качестве *базовых частичных двухместных операций*. Поскольку они обладают тем замечательным свойством, что сводятся к полностью определенным булевым операциям, применение этих операций к частичным переменным и функциям не может привести дополнительную неопределенность по сравнению с той, которая в этих переменных и функциях уже имеется.

2.4. *Формулы частичной алгебры*

Одноместные и двухместные частичные операции являются базисом для представления других частичных операций или функций, зависящих от произвольного числа переменных. Описание частичных функций выполняется посредством *формул частичной алгебры (термов, выражений)*. Формула (терм) частичной алгебры имеет следующее рекурсивное определение:

1. константа 0, 1, dc ;
2. частичная переменная u_i при $i=1,2,\dots$
3. терм $@t$, где $@$ - одноместная частичная операция идентичность или отрицание, t - формула частичной алгебры;
4. терм $t\$s$, где $\$$ - двухместная частичная операция; t, s - формулы частичной алгебры;
5. терм (t) , где t - формула частичной алгебры.

Порядок выполнения операций определяется расстановкой круглых скобок. Скобки могут опускаться. При этом устанавливается и используется *приоритет операций*, операции с более высоким приоритетом выполняются первыми. Операции с одинаковым приоритетом выполняются в порядке их следования слева направо. Одноместные операции имеют более высокий приоритет по сравнению с двухместными операциями. Приоритет двухместных частичных операций снижается в следующем порядке:

1. конъюнкция ($\&$);
2. дизъюнкция ($+$);
3. штрих Шеффера (\oslash), стрелка Пирса (\oslash);
4. прямая (\Rightarrow) и обратная (\Leftarrow) импликации;
5. исключаящее или (\oplus), эквивалентность (\Leftrightarrow).

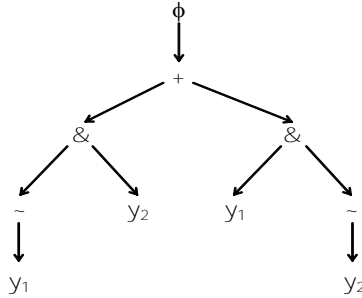


Рисунок 2.5 – Дерево разбора формулы Φ

Структура формулы представляется древовидным графом, называемым *деревом разбора*. Нетерминальными вершинами дерева являются частичные операции, входящие в формулу. Терминальными вершинами (листьями) дерева являются частичные константы и переменные. Корню дерева соответствует частичная функция, описываемая формулой. Значение формулы вычисляется в процессе обхода дерева разбора снизу вверх слева направо при условии, что частичные переменные получают конкретные значения.

Пример. Пусть частичная функция $\Phi(y_1, y_2)$ описывается формулой

$$\Phi = \sim y_1 \& y_2 + y_1 \& \sim y_2.$$

Дерево разбора формулы изображено на рис.2.5. При построении дерева учтено убывание приоритета частичных операций \sim , $\&$, $+$. Вычисление значения формулы Φ при различных значениях переменных y_1, y_2 показывает, что она описывает частичную двухместную операцию исключающее или $y_1 \odot y_2$ (строка 7 таблицы 2.5) через операции \sim , $\&$, $+$. \square

2.5. Сильные законы частичной алгебры

Сильный закон в частичной алгебре представляется равенством двух термов (формул), каждый из которых описывает одну и ту же частичную функцию. В частичной алгебре действуют следующие сильные законы.

Законы *ассоциативности* изменяют порядок выполнения частичных операций $\&$ и $+$:

$$y_1 \& (y_2 \& y_3) = (y_1 \& y_2) \& y_3, \quad (2.1)$$

$$y_1 + (y_2 + y_3) = (y_1 + y_2) + y_3. \quad (2.2)$$

Законы *коммутативности* переставляют операнды операций $\&$ и $+$:

$$y_1 \& y_2 = y_2 \& y_1, \quad (2.3)$$

$$y_1 + y_2 = y_2 + y_1. \quad (2.4)$$

Законы *дистрибутивности* перераспределяют операции $\&$ и $+$:

$$y_1 \& (y_2 + y_3) = (y_1 \& y_2) + (y_1 \& y_3), \quad (2.5)$$

$$y_1 + (y_2 \& y_3) = (y_1 + y_2) \& (y_1 + y_3). \quad (2.6)$$

Законы *идентичности* для частичных операций $\&$ и $+$ удаляют константы 0 и 1 из формул:

$$y \& 1 = y, \quad (2.7)$$

$$y + 0 = y. \quad (2.8)$$

Закон *аннигилятора* для частичных операций $\&$ и $+$ удаляют переменные из формул:

$$y \& 0 = 0. \quad (2.9)$$

$$y + 1 = 1. \quad (2.10)$$

Законы *идемпотентности* для частичных операций $\&$ и $+$ упрощают формулы:

$$y \& y = y, \quad (2.11)$$

$$y + y = y. \quad (2.12)$$

Законы *поглощения* для операций $\&$ и $+$ удаляют термы из формул:

$$y_1 \& (y_1 + y_2) = y_1, \quad (2.13)$$

$$y_1 + (y_1 \& y_2) = y_1. \quad (2.14)$$

Закон *двойного отрицания* исключает из формулы операцию \sim , имеющую два последовательных вхождения:

$$\sim \sim y = y. \quad (2.15)$$

Таблица 2.7

Доказательство законов де Моргана в частичной алгебре

| y_1 | y_2 | $\sim y_1$ | $\sim y_2$ | y_1+y_2 | $y_1&y_2$ | $\sim y_1+\sim y_2$ | $\sim y_1&\sim y_2$ | $\sim(y_1+y_2)$ | $\sim(y_1&y_2)$ |
|-------|-------|------------|------------|-----------|-----------|---------------------|---------------------|-----------------|-----------------|
| 1 | | 2 | | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | dc | 1 | dc | dc | 0 | 1 | dc | dc | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | dc | 0 | dc | 1 | dc | dc | 0 | 0 | dc |
| dc | 0 | dc | 1 | dc | 0 | 1 | dc | dc | 1 |
| dc | 1 | dc | 0 | 1 | dc | dc | 0 | 0 | dc |
| dc | dc | dc | dc | dc | dc | dc | dc | dc | dc |

Законы де Моргана взаимно заменяют операции $\&$ и $+$:

$$\sim y_1 \& \sim y_2 = \sim(y_1 + y_2), \quad (2.16)$$

$$\sim y_1 + \sim y_2 = \sim(y_1 \& y_2). \quad (2.17)$$

Другие законы для частичных операций $\&$, $+$, \Rightarrow и \odot :

$$0 \Rightarrow y = 1, \quad (2.18)$$

$$y_1 \Rightarrow y_2 = \sim y_2 \Rightarrow \sim y_1, \quad (2.19)$$

$$y + y \& dc = y, \quad (2.20)$$

$$y + \sim y \& dc = y + dc, \quad (2.21)$$

$$y \odot 1 = \sim y. \quad (2.22)$$

Законы 2.1-2.22 легко доказываются проверкой равенства значений левой и правой частей при всех значениях входящих в них переменных. Проиллюстрируем доказательство законов де Моргана, используя табл. 2.7. Так доказательство закона (2.16) следует из равенства столбцов 6 и 7 таблицы, закона (2.17) – из равенства столбцов 5 и 8. Поскольку законы выполняются при всех возможных комбинациях значений переменных, включая значение dc , они называются сильными.

Законы (2.1)-(2.22) используют, главным образом, три операции \sim , $\&$ и $+$. Следующие законы позволяют выразить другие частичные операции, определенные в табл. 2.6, через три указанные:

$$y_1 \Rightarrow y_2 = \sim y_1 + y_2, \quad (2.23)$$

$$y_1 \Leftarrow y_2 = y_1 + \sim y_2, \quad (2.24)$$

$$y_1 \odot y_2 = y_1 \& \sim y_2 + \sim y_1 \& y_2, \quad (2.25)$$

$$y_1 \Leftrightarrow y_2 = y_1 \& y_2 + \sim y_1 \& \sim y_2, \quad (2.26)$$

$$y_1 \overset{\text{♂}}{\circlearrowleft} y_2 = \sim(y_1 \& y_2), \quad (2.27)$$

$$y_1 \overset{\text{♀}}{\circlearrowright} y_2 = \sim(y_1 + y_2). \quad (2.28)$$

Прямая импликация \Rightarrow и обратная импликация \Leftarrow представляются через отрицание и дизъюнкцию. Исключающее или \odot и эквивалентность \Leftrightarrow представляются через отрицание, конъюнкцию и дизъюнкцию. Штрих Шеффера $\overset{\text{♂}}{\circlearrowleft}$ является отрицанием операции конъюнкция, стрелка Пирса $\overset{\text{♀}}{\circlearrowright}$ – отрицанием операции дизъюнкция, эквивалентность – отрицанием операции исключающее или.

2.6. Слабые законы частичной алгебры

В частичной алгебре в дополнение к сильным законам выполняются также слабые законы. *Слабый закон* представляется равенством двух термов (формул), описывающих частичные функции, принадлежащие одному классу Т-эквивалентности и сводящиеся к одной и той же не полностью определенной функции.

Законы *исключенного третьего* для частичных операций $\&$ и $+$

$$y \& \sim y = 0, \quad (2.29)$$

$$y + \sim y = 1. \quad (2.30)$$

выполняются не полностью, так как их левые части при $y = dc$ дают значение dc , а не константы 0 и 1 (табл.2.8). В то же время левая часть закона (2.29) описывает частичную функцию с номером 3 в табл.2.2, которая Т-эквивалентна константе 0 (строка 1 табл. 2.3). Аналогично левая часть закона (2.30) описывает частичную функцию с номером 15 в табл.2.2, которая Т-эквивалентна константе 1 (см. строку 5 табл. 2.3). Закон исключенного третьего является слабым в частичной логике и действует только с учетом возможности доопределения значения dc до значений 0 либо 1. Более точно,

Таблица 2.8

Доказательство законов исключенного третьего

| у | $y \& \sim y$ | Константа 0 | $y + \sim y$ | Константа 1 |
|----|---------------|-------------|--------------|-------------|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| dc | dc | 0 | dc | 1 |

Таблица 2.9

Доказательство слабых законов

| № | y_1 | y_2 | y_1+y_2 | $y_1 \& y_2 + y_1 \& \sim y_2$ | $y_1 + \sim y_1 \& y_2$ | $y_1 \odot y_2 \odot y_1 \& y_2$ |
|---|-------|-------|-----------|--------------------------------|-------------------------|----------------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | dc | dc | 0 | dc | dc |
| 4 | 1 | 0 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | dc | 1 | dc | 1 | dc |
| 7 | dc | 0 | dc | dc | dc | dc |
| 8 | dc | 1 | 1 | dc | dc | dc |
| 9 | dc | dc | dc | dc | dc | dc |

закон выполняется за счет возможности сведения различающихся частичных функций к одной и той же полностью определенной функции.

Из-за слабости закона исключенного третьего слабыми являются также и другие законы частичной алгебры (см. табл.2.9). Прежде всего, это закон *склеивания конъюнкции*:

$$y_1 \& y_2 + y_1 \& \sim y_2 = y_1 \& (y_2 + \sim y_2) = y_1 \& 1 = y_1. \quad (2.31)$$

Левая и правая части закона (2.31) описывают Т-эквивалентные частичные функции, определенные столбцами 4 и 1 табл.2.9. Функции имеют различия в значениях, стоящих на пересечении указанных столбцов со строкой 6.

Закон *поглощения литерала*:

$$y_1 + \sim y_1 \& y_2 = y_1 + y_1 \& y_2 + \sim y_1 \& y_2 = y_1 + y_2 \& (y_1 + \sim y_1) = y_1 + y_2 \& 1 = y_1 + y_2. \quad (2.32)$$

Левая и правая части закона (2.32) описывают Т-эквивалентные частичные функции, определенные столбцами 5 и 3 табл.2.9. Функции имеют различия в значениях, стоящих на пересечении указанных столбцов со строкой 8.

Закон *определения частичной дизъюнкции через исключаящее или*:

$$y_1 + y_2 = y_1 \odot y_2 \odot y_1 \& y_2. \quad (2.33)$$

Левая и правая части закона (2.33) описывают T -эквивалентные частичные функции, определенные столбцами 3 и 6 табл.2.9. Функции имеют различия в значениях, стоящих на пересечении указанных столбцов со строками 6 и 8.

2.7. Сильные формы представления частичных функций

Сильная форма представления $SF(\Phi)$ частичной функции $\Phi(y)$ описывает частичную функцию $\Psi(y)$ такую, что $\Psi(y)=\Phi(y)$. Другими словами, необходимо, чтобы исходная функция $\Phi(y)$ и функция $\Psi(y)$, описываемая формой SF , были идентичными, при этом не допускается замена значения функции dc на значение 0 или 1 и значения функции 0 или 1 на значение dc на каких либо наборах значений переменных. К сильным формам представления частичных функций относятся:

- *частичная совершенная дизъюнктивно-конъюнктивная форма (ЧСДКФ);*
- *частичная совершенная конъюнктивно-дизъюнктивная форма (ЧСКДФ).*

Пусть частичная функция $\Omega(y)$ задана в табличной форме, каждая строка которой определяет вектор значений $a=(a_1, \dots, a_n)$ частичных переменных y_1, \dots, y_n . На векторе a функция Ω принимает одно из трех значений: 0, 1, dc . Векторы a, b , на которых $\Omega(a)=1$ и $\Omega(b)=0$, являются ортогональными. Частичная совершенная дизъюнктивно-конъюнктивная форма частичной функции строится в виде

$$\text{ЧСДКФ}(\Omega) = \Omega_1 + (\sim \Omega_0) \& dc, \quad (2.34)$$

где частичные функции

$$\Omega_1 = \sum_{\Omega(a_1, \dots, a_n)=1} \left(\bigwedge_{i=1}^n y_i^{a_i} \right), \quad (2.35)$$

$$\Omega_0 = \sum_{\Omega(a_1, \dots, a_n)=0} \left(\bigwedge_{i=1}^n y_i^{a_i} \right), \quad (2.36)$$

где

$$y_i^{a_i} = \begin{cases} \sim y_i, & \text{если } a_i=0, \\ y_i, & \text{если } a_i=1, \\ \#y_i, & \text{если } a_i=dc. \end{cases} \quad (2.37)$$

Частичные функции Ω_1 и Ω_0 в ЧСДКФ(Ω) являются по существу дизъюнкциями элементарных частичных конъюнкций, построенных на наборах $a=(a_1, \dots, a_n)$ значений аргументов, на которых функции принимают значение 1 и 0 соответственно. Частичная элементарная конъюнкция есть конъюнкция литералов, представленных согласно (2.37) частичными переменными, которым могут предшествовать частичные одноместные операции отрицания (\sim) и идентификации неопределенности ($\#$). Частичная операция идентификации неопределенности определена в табл.2.10. Операция принимает значение 1 когда аргумент имеет значение dc , и принимает значение dc когда аргумент равен 0 или 1.

Корректность формулы (2.34) для ЧСДКФ(Ω) доказывают следующие рассуждения. Сначала проанализируем значения, принимаемые частичной функцией Ω_1 на наборе $a=(a_1, \dots, a_n)$ значений частичных переменных. Пусть $k_a=l_1 \& \dots \& l_n$ – элементарная конъюнкция дизъюнкции Ω_1 , построенная на наборе a , где l_i – i -ый литерал, построенный на значении a_i . Покажем, что значением конъюнкции k_a на наборе a является 1. Согласно (2.37), если $a_i=0$, то $l_i=\sim y_i$ и значением литерала является $value(\sim a_i)=1$. Если $a_i=1$, то $l_i=y_i$ и значением литерала снова является $value(a_i)=1$. Если $a_i=dc$, то $l_i=\#y_i$ и значение литерала определяется по табл.2.10. Им является $value(\#a_i)=1$. Поскольку $value(l_i)=1$ для всех литералов l_i конъюнкции $k(a)$ при $i=1, \dots, n$, значением конъюнкции $value(k_a(a))$ на наборе a также является 1.

Набор a называется *поглощаемым набором* b , если при $a \neq b_i$, $i=1, \dots, n$ всегда выполняется $b_i=dc$. В этом случае говорят также, что

Таблица 2.10
Частичная операция идентификации dc

| | |
|----|----|
| y | #y |
| 0 | dc |
| 1 | dc |
| dc | 1 |

набор b поглощает набор a . Если набор a поглощается набором b , то конъюнкция k_a называется *поглощаемой* конъюнкцией k_b .

Легко видеть, что если k_a поглощает k_b и $value(k_a(a))=1$, то $value(k_b(a))=dc$. Действительно, если k_a поглощает k_b , то найдется i такое, что $a_i=dc$ и $b_i=0$ либо $b_i=1$. Литерал l_i конъюнкции k_b имеет вид $\neg y_i$ при $b_i=0$, либо вид y_i при $b_i=1$. На значении $y_i=a_i=dc$ значением литерала l_i является dc в обоих случаях. Если значением хотя бы одного литерала конъюнкции k_b является dc , а значением остальных литералов является 1, значением конъюнкции является dc . Таким образом, значением конъюнкции k_b на наборе a является dc , если a поглощает b .

Легко также видеть, что если k_a поглощается k_b и $value(k_a(a))=1$, то $value(k_b(a))=dc$. Действительно, если k_a поглощается k_b , то найдется i такое, что $a_i=0$ либо $a_i=1$ и $b_i=dc$. Литерал l_i конъюнкции k_b имеет вид $\#y_i$. Как следует из табл.2.10, на значениях 0 и 1 переменной y_i этот литерал принимает значение dc . Если значением хотя бы одного литерала конъюнкции k_b является dc , а значением остальных литералов является 1, значением конъюнкции является dc . Таким образом, значением конъюнкции k_b на наборе a является dc , если a поглощается b .

Очевидно, что если наборы a и b являются ортогональными и $value(k_a(a))=1$, то $value(k_b(a))=0$. Другими словами, конъюнкции k_a и k_b являются ортогональными и, если конъюнкция k_a принимает значение 1 на наборе a , то конъюнкция k_b принимает значение 0 на этом же наборе.

Частичная функция Ω_0 строится аналогично частичной функции Ω_1 . Отличием является то, что построение ведется на наборах, на которых исходная частичная функция Ω принимает значение 0. Другими словами, Ω_0 строится на базе инверсии $\sim\Omega$. По этой причине функция Ω_0 входит в ЧСДКФ(Ω) (2.34) с отрицанием. Результаты анализа конъюнкций дизъюнкции Ω_1 относятся в полной мере к конъюнкциям дизъюнкции Ω_0 .

Между конъюнкциями дизъюнкции Ω_1 и Ω_0 имеются взаимосвязи, существенно влияющие на значение ЧСДКФ(Ω). Если конъюнкция k_a поглощает k_b , то обе конъюнкции входят одновременно либо в Ω_1 , либо в Ω_0 . Возможны девять комбинаций значений дизъюнкции Ω_1 и Ω_0 . При $\Omega_1=1$ всегда выполняется $\Omega_0=0$, при этом

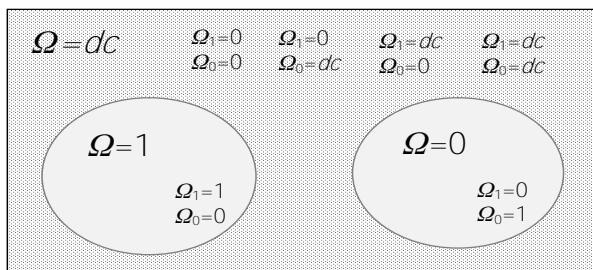


Рисунок 2.6 – Разбиение области определения частичной функции Ω в соответствии со значениями частичных функций Ω_1, Ω_0

ЧСДКФ(Ω)=1. При $\Omega_0=1$ всегда выполняется $\Omega_1=0$, при этом ЧСДКФ(Ω)=0. Это обусловлено попарной ортогональностью конъюнкций из Ω_1 и Ω_0 . Более того, появление комбинации $\Omega_1=1, \Omega_0=1$, комбинации $\Omega_1=1, \Omega_0=dc$ и комбинации $\Omega_1=dc, \Omega_0=1$ является невозможным. Комбинация $\Omega_1=0, \Omega_0=0$, комбинация $\Omega_1=0, \Omega_0=dc$, комбинация $\Omega_1=dc, \Omega_0=0$ и комбинация $\Omega_1=dc, \Omega_0=dc$ дают значение ЧСДКФ(Ω)=dc. Влияние шести допустимых комбинаций из девяти возможных на правильное поведение ЧСДКФ(Ω) иллюстрируется рис.2.6.

Пример 2.1. Пусть задана частичная функция $\Omega(y_1, y_2, y_3)$, определенная в табл.2.11. Эта функция является не склеенной и не развернутой. Не склеенными являются наборы 13 и 14, на которых функция Ω принимает значение 1. На наборе 15, являющемся результатом склеивания, функция принимает значение dc. Не развернутым является набор 20, на котором функция принимает значение 1. Он разворачивается в наборы 2 и 11, причем на наборе 2 функция принимает значение dc.

Выбирая из табл.2.11 конститутенты 1, по формуле (2.35) строим функцию

$$\Omega_1 = y_1 \& \sim y_2 \& y_3 + y_1 \& y_2 \& \sim y_3 + y_1 \& y_2 \& y_3 + y_1 \& \# y_2 \& y_3 + \# y_1 \& \sim y_2 \& y_3. \quad (2.38)$$

Выбирая из табл.2.11 конститутенты 0, по формуле (2.36) строим функцию

Таблица 2.11

Частичная совершенная дизъюнктивно-конъюнктивная форма

| № | y_1 | y_2 | y_3 | Функция Ω | Ω_1 | Ω_0 | ЧСДКФ(Ω) |
|----|-------|-------|-------|---------------------|------------|------------|-------------------|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | dc | 0 | 0 | dc |
| 3 | 0 | 0 | dc | dc | dc | dc | dc |
| 4 | 0 | 1 | 0 | dc | 0 | 0 | dc |
| 5 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 6 | 0 | 1 | dc | dc | 0 | dc | dc |
| 7 | 0 | dc | 0 | dc | 0 | dc | dc |
| 8 | 0 | dc | 1 | dc | dc | dc | dc |
| 9 | 0 | dc | dc | dc | dc | dc | dc |
| 10 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 12 | 1 | 0 | dc | dc | dc | dc | dc |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 14 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | dc | dc | dc | 0 | dc |
| 16 | 1 | dc | 0 | dc | dc | dc | dc |
| 17 | 1 | dc | 1 | 1 | 1 | 0 | 1 |
| 18 | 1 | dc | dc | dc | dc | dc | dc |
| 19 | dc | 0 | 0 | 0 | 0 | 1 | 0 |
| 20 | dc | 0 | 1 | 1 | 1 | 0 | 1 |
| 21 | dc | 0 | dc | dc | dc | dc | dc |
| 22 | dc | 1 | 0 | dc | dc | 0 | dc |
| 23 | dc | 1 | 1 | dc | dc | dc | dc |
| 24 | dc | 1 | dc | dc | dc | dc | dc |
| 25 | dc | dc | 0 | dc | dc | dc | dc |
| 26 | dc | dc | 1 | dc | dc | dc | dc |
| 27 | dc | dc | dc | dc | dc | dc | dc |

$$\Omega_0 = \sim y_1 \& \sim y_2 \& \sim y_3 + \sim y_1 \& y_2 \& y_3 + y_1 \& \sim y_2 \& \sim y_3 + \# y_1 \& \sim y_2 \& \sim y_3. \quad (2.39)$$

Используя формулу (2.34), строим совершенную дизъюнктивно-конъюнктивную форму частичной функции Ω :

$$\begin{aligned} \text{ЧСДКФ}(\Omega) = & (y_1 \& \sim y_2 \& y_3 + y_1 \& y_2 \& \sim y_3 + \\ & y_1 \& y_2 \& y_3 + y_1 \& \# y_2 \& y_3 + \# y_1 \& \sim y_2 \& y_3 \\ &) + \sim (\sim y_1 \& \sim y_2 \& \sim y_3 + \sim y_1 \& y_2 \& y_3 + \\ & y_1 \& \sim y_2 \& \sim y_3 + \# y_1 \& \sim y_2 \& \sim y_3 \\ &) \& \text{dc}. \end{aligned} \quad (2.40)$$

С целью проверки правильности построения ЧСДКФ(Ω), в табл.2.11 приведены значения функций Ω_1 и Ω_0 , вычисленные по формулам (2.38) и (2.39) при различных значениях переменных y_1, y_2, y_3 . Вычисление функции $\Omega(y_1, y_2, y_3)$ по формуле (2.34) дает последний столбец табл.2.11, который, как мы видим, полностью идентичен столбцу, описывающему исходную частичную функцию Ω .

Пример 2.2. Воспользуемся ЧСДКФ для представления одноместных частичных операций, описанных строками 9, 18, 21, 24, 27 табл. 2.1 и сводящихся к соответствующим не полностью определенным операциям, описанным строками 3, 6-9 табл. 2.2:

$$\Phi_9 = 0 + \sim(\sim y) \& dc = y \& dc, \quad (2.41)$$

$$\Phi_{18} = \sim y + \sim(0) \& dc = \sim y + dc, \quad (2.42)$$

$$\Phi_{21} = 0 + \sim(y) \& dc = \sim y \& dc, \quad (2.43)$$

$$\Phi_{24} = y + \sim(0) \& dc = y + dc, \quad (2.44)$$

$$\Phi_{27} = 0 + \sim(0) \& dc = dc. \quad (2.45)$$

Как видим, все эти одноместные частичные операции могут быть выражены через константу dc , инверсию \sim , конъюнкцию $\&$ и дизъюнкцию $+$.

Для получения частичной совершенной конъюнктивно-дизъюнктивной формы (ЧСКДФ) представим в ЧСДКФ частичную функцию $\sim\Omega$:

$$\text{ЧСДКФ}(\sim\Omega) = \Omega_0 + (\sim\Omega_1) \& dc. \quad (2.46)$$

Применяя операцию отрицания к обеим частям равенства (2.46), получаем ЧСКДФ частичной функции Ω :

$$\text{ЧСКДФ}(\Omega) = \sim(\Omega_0 + (\sim\Omega_1) \& dc). \quad (2.47)$$

После применения к формуле (2.47) законов де Моргана (2.16) и (2.17), закона двойного отрицания (2.15), закона коммутативности (2.3) и соотношения $\sim dc = dc$ получаем:

$$\text{ЧСКДФ}(\Omega) = (\sim\Omega_0) \& (\Omega_1 + dc). \quad (2.48)$$

Эквивалентное преобразование формул частичной алгебры и получение эквивалентных сильных форм представления частичных

функций выполняется с использованием сильных законов, описанных в разделе 2.5. Формулы, получаемые на всех этапах преобразования, описывают частичную функцию, полностью идентичную исходной частичной функции.

2.8. Слабые формы представления частичных функций

Слабая форма представления $WF(\Phi)$ частичной функции $\Phi(y)$ описывает частичную функцию $\Psi(y)$ такую, что $T(\Psi) = T(\Phi)$. Другими словами, для слабой формы необходимо, чтобы исходная функция $\Phi(y)$ и функция $\Psi(y)$, описываемая формой WF , принадлежали одному классу T -эквивалентности и описывали одну и ту же не полностью определенную функцию. При этом допускается замена значения функции dc на значение 0 или 1 и значения функции 0 или 1 на значение dc на некоторых наборах значений переменных при сохранении корректности функции $\Psi(y)$. К слабым формам представления частичных функций относятся:

- частичная дизъюнктивно-конъюнктивная форма (ЧДКФ),
- частичная конъюнктивно-дизъюнктивная форма (ЧКДФ),
- частичная сокращенная дизъюнктивно-конъюнктивная форма (ЧСКДФ),
- частичная минимальная дизъюнктивно-конъюнктивная форма (ЧМДФ) и другие.

Частичная одноместная операция $\#$ идентификации неопределенности dc (см. табл.2.10) относится к классу T -эквивалентности константа 1. Она может быть заменена на одноместную частичную операцию под номером 14 в табл.2.2 с сохранением не полностью определенной функции, к которой эта операция сводится. Подстановка константы 1 вместо литерала $\#y$ в формулы (2.35), (2.36), (2.37) изменит частичную функцию Ω , описываемую формулой (2.34), на некоторую другую частичную функцию Λ , однако сохранит равенство $T(\Lambda) = T(\Omega)$ не полностью определенных функций, к которым эти частичные функции сводятся. Таким образом, сильная форма ЧСДФ(Ω) представления частичной функции трансформируется в слабую *частичную дизъюнктивно-конъюнктивную форму* ЧДФ(Ω) представления этой же функции:

$$\text{ЧДКФ}(\Omega) = \Lambda = \Lambda_1 + (\sim \Lambda_0) \& dc, \quad (2.49)$$

где частичные функции

$$\Lambda_1 = \sum_{\Omega(a_1, \dots, a_n)=1} \left(\&_{\substack{i=1, \dots, n, \\ a_i \neq dc}} y_i^{a_i} \right), \quad (2.50)$$

$$\Lambda_0 = \sum_{\Omega(a_1, \dots, a_n)=0} \left(\&_{\substack{i=1, \dots, n, \\ a_i \neq dc}} y_i^{a_i} \right) \quad (2.51)$$

где

$$y_i^{a_i} = \begin{cases} \sim y_i, & \text{если } a_i=0, \\ y_i, & \text{если } a_i=1. \end{cases} \quad (2.52)$$

Функция Λ , описываемая формулой (2.49), является развернутой, но не склеенной частичной функцией. Переход от ЧСДКФ(Ω) к ЧДКФ(Ω) приводит к развертыванию всех развертываемых наборов значений частичной векторной переменной. Склеивание склеиваемых наборов при этом не происходит.

Пример 2.3. Построим частичную дизъюнктивно-конъюнктивную форму частичной функции $\Omega(y_1, y_2, y_3)$, описанной табл.2.11. Эта функция является не склеенной и не развернутой. Используя формулы (2.49)-(2.52), получаем:

$$\begin{aligned} \text{ЧДКФ}(\Omega) = & (y_1 \& \sim y_2 \& y_3 + y_1 \& y_2 \& \sim y_3 + \\ & y_1 \& y_2 \& y_3 + y_1 \& y_3 + \sim y_2 \& y_3 \\ &) + \sim (\sim y_1 \& \sim y_2 \& \sim y_3 + \sim y_1 \& y_2 \& y_3 + \\ & y_1 \& \sim y_2 \& \sim y_3 + \sim y_2 \& \sim y_3 \\ &) \& dc. \end{aligned} \quad (2.53)$$

Табл.2.12 описывает частичные функции Λ_1 , Λ_0 и Λ , представленные формулой (2.53). Функции Ω и Λ имеют различные значения на наборе 2, на котором функция Λ поменяла значение dc на значение 1.

Частичная конъюнктивно-дизъюнктивная форма (ЧКДФ) частичной функции Ω выводится из совершенной конъюнктивно-дизъюнктивной формы (2.48) путем замены частичных функций Ω_1 , Ω_0 , на частичные функции Λ_1 , Λ_0 :

Таблица 2.12

Частичная дизъюнктивно-конъюнктивная форма

| № | y_1 | y_2 | y_3 | Λ_1 | Λ_0 | Λ | Ω | Δ |
|----|-------|-------|-------|-------------|-------------|-----------|----------|----------|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 | 0 | 1 | dc | 1 |
| 3 | 0 | 0 | dc | dc | dc | dc | dc | dc |
| 4 | 0 | 1 | 0 | 0 | 0 | dc | dc | dc |
| 5 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | dc | 0 | dc | dc | dc | dc |
| 7 | 0 | dc | 0 | 0 | dc | dc | dc | dc |
| 8 | 0 | dc | 1 | dc | dc | dc | dc | dc |
| 9 | 0 | dc | dc | dc | dc | dc | dc | dc |
| 10 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 0 | dc | dc | dc | dc | dc | dc |
| 13 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 14 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 15 | 1 | 1 | dc | dc | 0 | dc | dc | 1 |
| 16 | 1 | dc | 0 | dc | dc | dc | dc | dc |
| 17 | 1 | dc | 1 | 1 | 0 | 1 | 1 | 1 |
| 18 | 1 | dc | dc | dc | dc | dc | dc | dc |
| 19 | dc | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 20 | dc | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 21 | dc | 0 | dc | dc | dc | dc | dc | dc |
| 22 | dc | 1 | 0 | dc | 0 | dc | dc | dc |
| 23 | dc | 1 | 1 | dc | dc | dc | dc | dc |
| 24 | dc | 1 | dc | dc | dc | dc | dc | dc |
| 25 | dc | dc | 0 | dc | dc | dc | dc | dc |
| 26 | dc | dc | 1 | dc | dc | dc | dc | dc |
| 27 | dc | dc | dc | dc | dc | dc | dc | dc |

$$\text{ЧКДФ}(\Omega) = (\sim \Lambda_0) \& (\Lambda_1 + dc). \quad (2.54)$$

ЧКДФ также как и ЧДКФ описывает развернутую, но не склеенную частичную функцию Λ .

Частичная сокращенная дизъюнктивно-конъюнктивная форма (ЧСкДКФ) частичной функции Ω строится по ЧДКФ выполнением трех операций:

1. склеиванием всех склеиваемых наборов частичной функции Λ , описываемой ЧДКФ; в результате склеивания выполняется переход от частичной функции Λ к T -эквивалентной частичной склеено-развернутой функции Δ ;

- поглощением всех поглощаемых наборов функции Δ с сохранением простых импликант; если $\Delta(dc, \dots, dc) = dc$, то все наборы, на которых функция Δ принимает значение dc , поглощаются набором (dc, \dots, dc) ; все поглощенные наборы удаляются из таблицы, описывающей функцию Δ ;
- построением формулы ЧСкДКФ по оставшимся в таблице наборам:

$$\text{ЧСкДКФ}(\Omega) = \Delta = \Delta_1 + (\sim \Delta_0) \& dc \quad (2.55)$$

где функции Δ_1, Δ_0 описываются аналогично функциям Λ_1, Λ_0 (см. формулы (2.50)-(2.52)).

Заметим, что операции склеивания и поглощения наборов (конъюнкций) частичной функции аналогичны одноименным операциям, разработанным Мак-Класки и Квайном применительно к булевым функциям. Существенным отличием является то, в формулу (2.55) входят не булевы операции отрицания, конъюнкции, дизъюнкции, а одноименные частичные операции.

Если к сокращенным частичным дизъюнкциям конъюнкций Δ_1, Δ_0 применить метод построения матрицы Квайна и найти минимальное столбцовое покрытие всех конститuent 1 и всех конститuent 0, то в результате построим *частичную минимальную дизъюнктивно-конъюнктивную форму* (ЧМДКФ).

Пример 2.4. Построим ЧСкДКФ частичной функции Ω , описанной табл.2.12. Склеивание всех склеиваемых конъюнкций приводит к построению функции Δ (последний столбец табл.2.12). На наборе 15 функция Δ получила значение 1 вместо значения dc . Поглощение всех поглощаемых наборов оставляет наборы 5, 15, 17, 19, 20, 27 не поглощенными. Все поглощенные наборы отмечены заливкой в столбце Δ табл.2.12. Табл.2.13 описывает сокращенную табличную форму частичной функции Ω . Соответствующая ЧСкДКФ, построенная по формулам (2.50)-(2.52), (2.55), имеет следующий вид:

$$\begin{aligned} \text{СкДКФ}(\Omega) = & (y_1 \& y_2 + y_1 \& y_3 + \sim y_2 \& y_3 \\ &) + \sim(\sim y_1 \& y_2 \& y_3 + \sim y_2 \& \sim y_3 \\ &) \& dc. \end{aligned} \quad (2.56)$$

Таблица 2.13

Сокращенная табличная форма частичной функции Ω

| № | y_1 | y_2 | y_3 | Функция Δ |
|---|-------|-------|-------|------------------|
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 1 | dc | 1 |
| 3 | 1 | dc | 1 | 1 |
| 4 | dc | 0 | 0 | 0 |
| 5 | dc | 0 | 1 | 1 |
| 6 | dc | dc | dc | dc |

Таблица 2.14

Матрица Квайна для частичной функции Δ_1

| | $y_1 \& y_2$ | $y_1 \& y_3$ | $\sim y_2 \& y_3$ |
|-------------------------------|--------------|--------------|-------------------|
| $\sim y_1 \& \sim y_2 \& y_3$ | | | 1 |
| $y_1 \& \sim y_2 \& y_3$ | | 1 | 1 |
| $y_1 \& y_2 \& \sim y_3$ | 1 | | |
| $y_1 \& y_2 \& y_3$ | 1 | 1 | |

Таблица 2.15

Матрица Квайна для частичной функции Δ_0

| | $\sim y_1 \& y_2 \& y_3$ | $\sim y_2 \& \sim y_3$ |
|------------------------------------|--------------------------|------------------------|
| $\sim y_1 \& \sim y_2 \& \sim y_3$ | | 1 |
| $\sim y_1 \& y_2 \& y_3$ | 1 | |
| $y_1 \& \sim y_2 \& \sim y_3$ | | 1 |

Для построения ЧМДКФ формируем матрицы Квайна для двух частичных функций: Δ_1 (табл.2.14) и Δ_0 (табл.2.15). Столбцы соответствуют простым импликантам, строки соответствуют конститuentам 1 и 0. Нахождение минимальных столбцовых покрытий дает

$$\begin{aligned} \text{ЧМДКФ}(\Omega) = & (y_1 \& y_2 + \sim y_2 \& y_3 \\ &) + \sim(\sim y_1 \& y_2 \& y_3 + \sim y_2 \& \sim y_3 \\ &) \& dc. \end{aligned} \quad (2.57)$$

Преобразование формул и получение эквивалентных слабых форм представления частичных функций выполняется с использованием слабых и сильных законов, описанных в разделах 2.6, 2.5.

При использовании слабых законов, получаемые формулы описывают Т-эквивалентные частичные функции.

2.9. Функционально полные базисы

Функционально полные базисы представления частичных функций можно разделить с учетом полученных выше результатов на сильные (сокращенно SB) и слабые (сокращенно WB). Анализ совершенных дизъюнктивно-конъюнктивной и конъюнктивно-дизъюнктивной форм представления частичных функций приводит к построению сильного базиса $SB_1 = \{0, dc, \sim, \#, \&, +\}$, включающего константы 0 и dc , одноместные операции отрицания \sim и идентификации неопределенности $\#$, двухместные операции конъюнкции $\&$ и дизъюнкции $+$.

Расширенные законы де Моргана позволяют выразить дизъюнкцию $+$ через конъюнкцию $\&$ и отрицание \sim и выразить конъюнкцию $\&$ через дизъюнкцию $+$ и отрицание \sim . Следовательно, следующие два базиса $SB_2 = \{0, dc, \sim, \#, \&\}$ и $SB_3 = \{0, dc, \sim, \#, +\}$ также являются сильными и включают меньшее число операций.

Более компактные сильные базисы строятся с использованием двухместных частичных операций штрих Шеффера и стрелка Пирса. Так, частичная одноместная операция отрицания \sim выражается через операцию штрих Шеффера и операцию стрелка Пирса как:

$$\sim y = y \overset{\delta}{\delta} y = y \underset{\#}{\#} y. \quad (2.58)$$

Доказательство соотношения (2.58) непосредственно вытекает из табл.2.6, строки 9, 15. Частичная конъюнкция:

$$y_1 \& y_2 = (y_1 \overset{\delta}{\delta} y_2) \overset{\delta}{\delta} (y_1 \overset{\delta}{\delta} y_2) = (y_1 \underset{\#}{\#} y_1) \underset{\#}{\#} (y_2 \underset{\#}{\#} y_2). \quad (2.59)$$

Частичная дизъюнкция:

$$y_1 + y_2 = (y_1 \overset{\delta}{\delta} y_1) \overset{\delta}{\delta} (y_2 \overset{\delta}{\delta} y_2) = (y_1 \underset{\#}{\#} y_2) \underset{\#}{\#} (y_1 \underset{\#}{\#} y_2). \quad (2.60)$$

Доказательство соотношений (2.59), (2.60) вытекает из сильных законов (2.27), (2.28) и соотношения (2.58). Поскольку через частичные отрицание, конъюнкцию, дизъюнкцию выражаются частичный штрих Шеффера и частичная стрелка Пирса, следующие

два функционально полных базиса являются сильными: $SB_4 = \{0, dc, \#, \exists\}$, $SB_5 = \{0, dc, \#, \varphi\}$.

Переход к слабым дизъюнктивно-конъюнктивной и конъюнктивно-дизъюнктивной формам приводит к исключению из рассмотрения частичной операции идентификации неопределенности $\#$ и к построению слабых функционально полных базисов $WB_1 = \{0, dc, \sim, \&, +\}$, $WB_2 = \{0, dc, \sim, \&\}$, $WB_3 = \{0, dc, \sim, +\}$, $WB_4 = \{0, dc, \exists\}$, $WB_5 = \{0, dc, \varphi\}$.

Далее по тексту под термином частичная функция понимается корректно определенная частичная функция.

2.10. *Историческая справка*

Трехзначную частичную логику и частичные логические функции ввел С.К. Клини с целью формализации рассуждений о частичных рекурсивных предикатах, определенных на множестве натуральных чисел [38, 92 (1952)]. В работе М. Кербера и М. Колхазе [37 (1996)] предложен способ механизации логики Клини для частичных функций. В работах А.А. Прихожего [93-95 (1991), 53 (1997), 108 (2012)] исследована взаимосвязь частичных, не полностью определенных и булевых функций. Частичные функции использованы в качестве механизма построения представлений, конструирования разложений частично определенных функций, а также оперирования неопределенностью в логических системах. В настоящей книге проведен детальный анализ одноместных и двухместных частичных операций, изучены их свойства, введено понятие Т-эквивалентности частичных функций, дана классификация законов и форм представления частичных функций на сильные и слабые, исследованы функционально полные базисы.

Глава 3. ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ. СМЕШАННАЯ АЛГЕБРА

3.1. Кодирование частичных переменных парами булевых переменных значения и области определенности

Формой «значение/область определенности» (*Value-Care Representation – VCR* или *Value-Domain Representation – VDR*) называется следующее кодирование частичной переменной y_i парой полностью определенных булевых переменных v_i и d_i :

$$y_i = \begin{cases} 0, & \text{если } v_i=0 \text{ и } d_i=1, \\ 1, & \text{если } v_i=1 \text{ и } d_i=1, \\ dc, & \text{если } v_i \in \{0,1\} \text{ и } d_i=0. \end{cases} \quad (3.1)$$

Булева переменная v_i называется *переменной значения*, булева переменная d_i называется *переменной области определенности*. При $d_i=1$ частичная переменная y_i принимает такое же определенное значение 0 или 1 как и булева переменная v_i . При $d_i=0$ переменная y_i принимает неопределенное значение dc независимо от значения переменной v_i .

Таким образом, частичной переменной y_i ставится в соответствие не полностью определенная функция $\varphi(v_i, d_i)$, зависящая от двух булевых переменных. Эта функция, описанная табл.3.1, записываемая в дальнейшем по тексту с использованием вертикальной черты в форме $y_i = (v_i | d_i)$.

Для функции $(v_i | d_i)$ имеют место следующие частные случаи:
 $(v_i | 0) = (0 | 0) = (1 | 0)$ – константа dc не зависимо от значения v_i ;
 $(v_i | 1)$ – полностью определенная переменная v_i ;
 $(0 | 1)$ – полностью определенная константа 0;
 $(1 | 1)$ – полностью определенная константа 1.

Таблица 3.1

Не полностью определенная функция $y_i = \varphi(v_i, d_i)$

| v_i | d_i | $y_i = \varphi(v_i, d_i)$ |
|-------|-------|---------------------------|
| 0 | 0 | dc |
| 0 | 1 | 0 |
| 1 | 0 | dc |
| 1 | 1 | 1 |

3.2. Частично определенные функции

Пусть $\Psi(y_1, \dots, y_m)$ – частичная логическая функция, зависящая от частичных переменных y_1, \dots, y_m . Благодаря кодированию частичных переменных y_i парами булевых переменных v_i и d_i при $i=1, \dots, m$ частичная функция Ψ трансформируется в *не полностью определенную функцию* ψ , зависящую от $2 \times m$ булевых переменных $v_1, d_1, \dots, v_m, d_m$.

Поставим в соответствие переменным v_i и d_i при $i=1, \dots, m$ булевы функции $f_i(x_1, \dots, x_n)$ и $h_i(x_1, \dots, x_n)$ соответственно, зависящие от первичных булевых переменных x_1, \dots, x_n каждая. Тогда, частичной переменной y_i поставлена в соответствие не полностью определенная функция $\varphi(f_i(x_1, \dots, x_n), h_i(x_1, \dots, x_n))$, зависящая от пары булевых функций и, в конечном счете, зависящая от первичных булевых переменных x_1, \dots, x_n . Согласно табл.3.1 эта функция принимает значения на множестве M .

Не полностью определенная функция φ разбивает булево пространство на три части как показано на рис.3.1:

- часть φ^{on} представляет собой подмножество значений векторного аргумента x , на котором $\varphi(x)=1$; это подмножество описывается множеством конституент 1 конъюнкции двух булевых функций f_i и h_i : $\varphi^{on} = (f_i \wedge h_i)^{on}$;
- часть φ^{off} представляет собой подмножество значений векторного аргумента x , на котором $\varphi(x)=0$; это подмножество описывается множеством конституент 1 конъюнкции двух булевых функций $\neg f_i$ и h_i : $\varphi^{off} = (\neg f_i \wedge h_i)^{on}$;

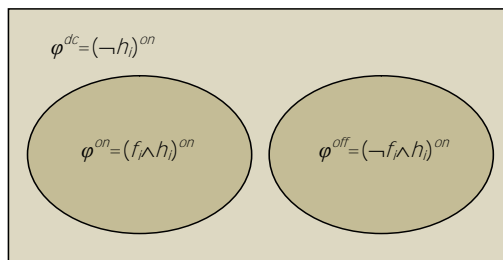


Рисунок 3.1 – Частично определенная функция в булевом пространстве

- часть φ^{dc} представляет собой подмножество значений векторного аргумента x , на котором $\varphi(x)=dc$; это подмножество описывается множеством конституент 1 булевой функции $\neg h_i$: $\varphi^{dc}=(\neg h_i)^{an}$;

В свою очередь частичная функция Ψ также может быть проинтерпретирована как не полностью определенная функция $\psi(x_1, \dots, x_n)$, зависящая от булевых переменных x_1, \dots, x_n . Функцию ψ , описанную посредством одновременного использования булевых переменных, а также частичных, не полностью определенных и булевых полностью определенных функций, будем называть *частично определенной логической функцией*. Частично определенные функции представляются смешанными формулами (термами или выражениями), использующими операции частичной и булевой алгебр, а также VCR форму.

Пример 3.1. Частичная функция Ψ , зависящая от двух частичных переменных y_1, y_2 , определена табл.3.2. Закодируем частичные переменные y_1, y_2 парами булевых переменных v_1, d_1, v_2, d_2 . По частичной функции Ψ построим частично определенную функцию ψ , описанную табл.3.3. Введем первичные булевы переменные x_1, x_2, x_3 и поставим в соответствие булевым переменным v_1, d_1, v_2, d_2 булевы функции f_1, h_1, f_2, h_2 , зависящие от первичных переменных x_1, x_2, x_3 . Функции f_1, h_1, f_2, h_2 определены в табл.3.4. Зная значения функций на каждом из наборов переменных x_1, x_2, x_3 , вычислим значения частичных переменных y_1, y_2 и частично определенной функции ψ и поместим их в табл.3.4. В результате получим новую частично определенную функцию ψ' , зависящую от первичных булевых переменных x_1, x_2, x_3 . □

Таблица 3.2

Пример частичной функции Ψ

| y_1 | y_2 | $\Psi(y_1, y_2)$ |
|-------|-------|------------------|
| 0 | 0 | 0 |
| 0 | 1 | dc |
| 0 | dc | dc |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | dc | 1 |
| dc | 0 | dc |
| dc | 1 | dc |
| dc | dc | dc |

Таблица 3.3

Частично определенная функция ψ

| Булевы переменные | | | | Функция ψ |
|-------------------|-------|-------|-------|----------------|
| v_1 | d_1 | v_2 | d_2 | |
| 0 | 0 | 0 | 0 | dc |
| 0 | 0 | 0 | 1 | dc |
| 0 | 0 | 1 | 0 | dc |
| 0 | 0 | 1 | 1 | dc |
| 0 | 1 | 0 | 0 | dc |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | dc |
| 0 | 1 | 1 | 1 | dc |
| 1 | 0 | 0 | 0 | dc |
| 1 | 0 | 0 | 1 | dc |
| 1 | 0 | 1 | 0 | dc |
| 1 | 0 | 1 | 1 | dc |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Таблица 3.4

Пример частично определенной функции ψ'

| x_1 | x_2 | x_3 | f_1 | h_1 | f_2 | h_2 | y_1 | y_2 | ψ' |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | dc | dc | dc |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | dc | 1 | dc |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | dc | 1 | dc |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | dc | 1 | dc |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | dc | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

В паре $y_i=(f_i|h_i)$ булевых функций, функция h_i , описывающая область определенности частичной функции y_i , фиксирована. Функция f_i , описывающая значение частичной функции y_i , может быть заменена на другую булеву функцию f_i' такую что $(f_i'|h_i)=(f_i|h_i)$. Ус-

ловием выполнения равенства двух частичных функций, представленных в VCR-форме с одной и той же функцией области определенности h_i , является выполнение равенства

$$f_i' \wedge h_i = f_i \wedge h_i \quad (3.2)$$

для булевых функций f_i, f_i', h_i .

3.3. Одноместные операции над частично определенными функциями

Перейдем от рассмотрения одноместной частичной операции $\Phi^1(y)$, зависящей от скалярной частичной переменной y (см. раздел 2.2), к рассмотрению одноместной операции $\Phi^1((v)d)$ над частично определенными функциями, представленными в VCR-форме парами булевых функций v и d . В обоих случаях функция Φ^1 принимает значения из множества M .

Кодирование значений из множества M парами значений из множества B показано на рис.3.2. Каждому частичному значению $0, 1 \in M$ соответствует одна пара значений из множества B^2 . Частичному значению dc соответствуют две пары значений из B^2 .

В табл.2.3 определены девять классов Т-эквивалентных одноместных частичных операций, сводящихся к девяти не полностью определенным функциям, зависящим от одной булевой переменной. Для каждого класса мы выберем из табл.2.2 по одной одноместной частичной склеено-развернутой операции и включим ее в табл.3.5. В первом столбце табл.3.5 даны номера частичных операций из табл.2.2. Операции с номерами 1, 6, 12, 14 сводятся к полностью определенным булевым операциям. Операции с номерами 9, 18, 21, 24, 27 сводятся к не полностью определенным булевым операциям.

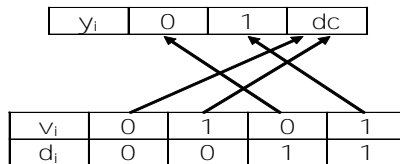


Рисунок 3.2 – Кодирование частичных значений парами булевых значений

Таблица 3.5

Одноместные операции над частично определенными функциями

| № частичной операции | Булевы переменные v, d | | | | Имя операции |
|----------------------|-----------------------------|----|----|----|---------------------------|
| | 01 | 11 | 00 | 10 | |
| 1 | 0 | 0 | 0 | 0 | Константа 0 |
| 6 | 0 | 1 | dc | dc | Идентичность |
| 9 | 0 | dc | dc | dc | Не полностью определенная |
| 12 | 1 | 0 | dc | dc | Инверсия |
| 14 | 1 | 1 | 1 | 1 | Константа 1 |
| 18 | 1 | dc | dc | dc | Не полностью определенная |
| 21 | dc | 0 | dc | dc | Не полностью определенная |
| 24 | dc | 1 | dc | dc | Не полностью определенная |
| 27 | dc | dc | dc | dc | Константа dc |

Выполним кодирование в форме VCR частичной скалярной переменной y , от которой зависит частичная одноместная операция, парой булевых переменных v и d . Тогда определение одноместных частичных операций из табл.2.2 трансформируется в соответствующее определение не полностью определенных двухместных функций как показано в табл.3.5. Для построения табл.3.5 использованы правила, описываемые соотношением (3.1). Проинтерпретируем не полностью определенные функции из табл.3.5 как частично определенные функции, представленные в VCR форме с использованием формул булевой алгебры:

$$\varphi_1 = (0|1), \quad (3.3)$$

$$\varphi_6 = (v|d), \quad (3.4)$$

$$\varphi_9 = (v|\neg v \wedge d), \quad (3.5)$$

$$\varphi_{12} = (\neg v|d), \quad (3.6)$$

$$\varphi_{14} = (1|1), \quad (3.7)$$

$$\varphi_{18} = (\neg v|\neg v \wedge d), \quad (3.8)$$

$$\varphi_{21} = (\neg v|v \wedge d), \quad (3.9)$$

$$\varphi_{24} = (v|v \wedge d), \quad (3.10)$$

$$\varphi_{27} = (v|0). \quad (3.11)$$

Доказательство соотношений (3.3)-(3.11) легче всего выполняется подстановкой всех возможных значений переменных v и d , вычислением значения соответствующей частично определенной функции по формуле (3.1) и сопоставлением этого значения с соответствующим значением из табл.2.2. Другой способ доказательства состоит в представлении результата операции в VCR форме ($f|h$) и построении булевых функций f и h , зависящих от переменных v и d .

Приведем в качестве примера доказательство соотношения (3.5) для функции φ_9 . Построим табл.3.6, первые три столбца которой повторяют определение функции φ_9 . Столбцы 4 и 5 определяют функцию значения f и функцию области определенности h в зависимости от v и d . Если значение функции h однозначно на всех наборах значений переменных v и d (если $\varphi_9=dc$, то $h=0$, в противном случае $h=1$), то значение функции f может варьироваться при $\varphi_9=dc$. Легко видеть, что функция области определенности описывается формулой $h=v \wedge d$. Выбирая для функции значения $f=0$ на наборе (0,0) и $f=1$ на наборах (1,0) и (1,1), получаем $f=v$, что доказывает соотношение (3.5). Однако, при выборе $f=0$ на всех наборах (0,0), (1,0) и (1,1) получаем $f=0$. Следовательно, функция φ_9 может быть представлена также в виде $\varphi_9=(0|_{\neg v \wedge d})$.

Анализ соотношений (3.3)-(3.11) выявляет следующую закономерность относительно функции области определенности h . Для частично определенных функций $\varphi_1, \varphi_6, \varphi_{12}, \varphi_{14}$, сводящихся к полностью определенным функциям, а также для функции-константы φ_{27} (dc), функция h не зависит от булевой переменной значения v . Для остальных частично определенных функций $\varphi_9, \varphi_{18}, \varphi_{21}, \varphi_{24}$ функция h существенно зависит от булевой переменной значения v .

Как видим, все одноместные частичные операции могут быть выражены через форму VCR, константы 0 и 1, булевы инверсию \neg и конъюнкцию \wedge .

Таблица 3.6

Доказательство соотношения (3.5)

| v | d | φ_9 | f | h | f' | f'' |
|---|---|-------------|-----|---|------|-------|
| 0 | 0 | dc | 0,1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | dc | 0,1 | 0 | 1 | 0 |
| 1 | 1 | dc | 0/1 | 0 | 1 | 0 |

3.4. Двухместные операции над частично определенными функциями

Перейдем от рассмотрения двухместной частичной операции $\Phi^2(y_1, y_2)$, зависящей от двух скалярных частичных переменных y_1 и y_2 (см. раздел 2.3), к рассмотрению двухместной операции над частично определенными функциями $\Phi^2((v_1|d_1), (v_2|d_2))$, каждая из которых представлена в VCR-форме парой булевых переменных (функций). В обоих случаях функция Φ^2 принимает значения из множества M .

Кодирование пар значений из множества M двумя парами значений из множества B показано на рис.3.3. Если кодируемая пара строится из значений $0, 1 \in M$, то ей соответствует один набор из четырех значений переменных v_1, d_1, v_2, d_2 . Если одним из элементов кодируемой пары является dc , то ей соответствует два набора из четырех значений булевых переменных. Если кодируемая пара имеет вид (dc, dc) , то ей соответствует четыре набора из четырех значений булевых переменных.

Выполним кодирование в форме VCR пары частичных переменных y_1, y_2 , от которых зависит частичная двухместная операция, двумя парами $(v_1|d_1)$ и $(v_2|d_2)$ булевых переменных. Тогда определенные десять базовых двухместных частичных операций из табл.2.6

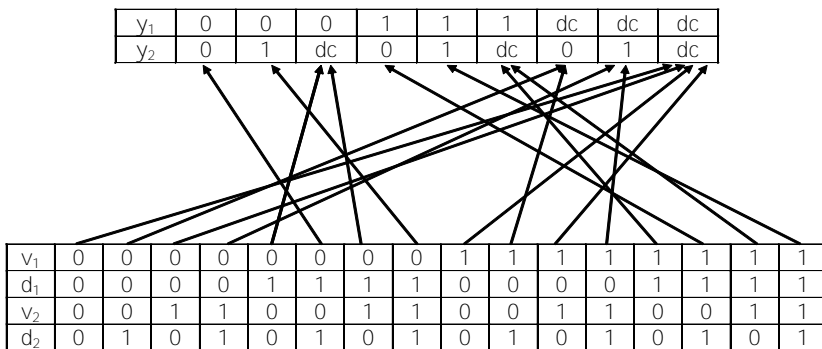


Рисунок 3.3 – Кодирование пар истинностных значений частичной алгебры двумя парами булевых значений

трансформируется в соответствующее определение не полностью определенных четырехместных функций как показано в табл.3.7. Для построения табл.3.7 использованы правила, описанные соотношением (3.1) и и представленные рис.3.3.

Проинтерпретируем не полностью определенные функции из табл.3.7 как частично определенные функции, представленные в VCR форме ($f|h$) с использованием формул булевой алгебры для функций f и h :

$$(v_1|d_1)\&(v_2|d_2) = (v_1 \wedge v_2 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2), \quad (3.12)$$

$$\sim((v_1|d_1)\Rightarrow(v_2|d_2)) = (\neg(v_1 \rightarrow v_2) | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee v_2 \wedge d_2), \quad (3.13)$$

$$\sim((v_1|d_1)\Leftarrow(v_2|d_2)) = (\neg(v_1 \leftarrow v_2) | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee \neg v_2 \wedge d_2), \quad (3.14)$$

$$(v_1|d_1)\ddot{\times}(v_2|d_2) = (v_1 \oplus v_2 | d_1 \wedge d_2), \quad (3.15)$$

$$(v_1|d_1)+(v_2|d_2) = (v_1 \vee v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee v_2 \wedge d_2), \quad (3.16)$$

$$(v_1|d_1)\bar{\cap}(v_2|d_2) = (v_1 \downarrow v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee v_2 \wedge d_2), \quad (3.17)$$

$$(v_1|d_1)\Leftrightarrow(v_2|d_2) = (v_1 \leftrightarrow v_2 | d_1 \wedge d_2), \quad (3.18)$$

$$(v_1|d_1)\Leftarrow(v_2|d_2) = (v_1 \leftarrow v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee \neg v_2 \wedge d_2), \quad (3.19)$$

$$(v_1|d_1)\Rightarrow(v_2|d_2) = (v_1 \rightarrow v_2 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee v_2 \wedge d_2), \quad (3.20)$$

$$(v_1|d_1)\bar{\delta}(v_2|d_2) = (v_1 / v_2 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2), \quad (3.21)$$

Докажем соотношение (3.12) для частичной конъюнкции. Соотношения (3.13)-(3.21) для других частичных операций доказываются аналогичным образом. Построим для частичной конъюнкции табл.3.8. Первые пять столбцов табл.3.8 повторяют определение частичной операции $\&$. Столбец 6 описывает возможные значения булевой функции f , а столбец 7 описывает значения булевой функции области определенности h на всех наборах значений булевых переменных v_1, d_1, v_2, d_2 . Значение функции h фиксировано на всех наборах значений переменных. Значение функции f варьируется на значениях dc частичной операции $\&$. Запишем функцию h в совершенной дизъюнктивной нормальной форме в соответствии со столбцом 7 табл.3.8:

$$\begin{aligned} h = & \neg v_1 \wedge \neg d_1 \wedge \neg v_2 \wedge d_2 \vee \neg v_1 \wedge d_1 \wedge \neg v_2 \wedge \neg d_2 \vee \\ & \neg v_1 \wedge d_1 \wedge \neg v_2 \wedge d_2 \vee \neg v_1 \wedge d_1 \wedge v_2 \wedge \neg d_2 \vee \\ & \neg v_1 \wedge d_1 \wedge v_2 \wedge d_2 \vee v_1 \wedge \neg d_1 \wedge \neg v_2 \wedge d_2 \vee \\ & v_1 \wedge d_1 \wedge \neg v_2 \wedge d_2 \vee v_1 \wedge d_1 \wedge v_2 \wedge d_2. \end{aligned} \quad (3.22)$$

Таблица 3.7

Двухместные операции над частично определенными функциями

| Булевы переменные | | | | Операция | | | | | | | | | |
|-------------------|-------|-------|-------|----------|--------------------|-------------------|---------|----|------------|-------------------|--------------|---------------|------------|
| v_1 | d_1 | v_2 | d_2 | & | $\sim \Rightarrow$ | $\sim \Leftarrow$ | \odot | + | ♀ | \Leftrightarrow | \Leftarrow | \Rightarrow | ♂ |
| 0 | 0 | 0 | 0 | dc | dc | dc | dc | dc | dc | dc | dc | dc | dc |
| 0 | 0 | 0 | 1 | 0 | dc | 0 | dc | dc | dc | dc | 1 | dc | 1 |
| 0 | 0 | 1 | 0 | dc | dc | dc | dc | dc | dc | dc | dc | dc | dc |
| 0 | 0 | 1 | 1 | dc | 0 | dc | dc | 1 | 0 | dc | dc | 1 | dc |
| 0 | 1 | 0 | 0 | 0 | 0 | dc | dc | dc | dc | dc | dc | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | dc | dc | dc | dc | dc | dc | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | dc | dc | 0 | dc | dc | dc | dc | 1 | dc | dc |
| 1 | 0 | 0 | 1 | 0 | dc | 0 | dc | dc | dc | dc | 1 | dc | 1 |
| 1 | 0 | 1 | 0 | dc | dc | dc | dc | dc | dc | dc | dc | dc | dc |
| 1 | 0 | 1 | 1 | dc | 0 | dc | dc | 1 | 0 | dc | dc | 1 | dc |
| 1 | 1 | 0 | 0 | dc | dc | 0 | dc | 1 | 0 | dc | 1 | dc | dc |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | dc | dc | 0 | dc | 1 | 0 | dc | 1 | dc | dc |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

Таблица 3.8

Доказательство соотношения (3.12)

| v_1 | d_1 | v_2 | d_2 | & | f | h | f' |
|-------|-------|-------|-------|----|-----|---|----|
| 0 | 0 | 0 | 0 | dc | 0,1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | dc | 0,1 | 0 | 0 |
| 0 | 0 | 1 | 1 | dc | 0,1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | dc | 0,1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | dc | 0,1 | 0 | 1 |
| 1 | 0 | 1 | 1 | dc | 0,1 | 0 | 1 |
| 1 | 1 | 0 | 0 | dc | 0,0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | dc | 0,1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Выполняя минимизацию дизъюнктивной формы (3.22) путем склеивания и поглощения элементарных конъюнкций, получаем:

$$h = d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2. \quad (3.23)$$

Легко видеть, что правая часть соотношения (3.23) совпадает с булевой функцией h области определенности правой части соотношения (3.12).

Выбирая из всех возможных значений конкретные значения функции f , как это показано в последнем столбце табл.3.5, получаем функцию f' , для которой совершенная дизъюнктивная нормальная форма имеет вид:

$$f' = v_1 \wedge \neg d_1 \wedge v_2 \wedge \neg d_2 \vee v_1 \wedge \neg d_1 \wedge v_2 \wedge d_2 \vee v_1 \wedge d_1 \wedge v_2 \wedge \neg d_2 \vee v_1 \wedge d_1 \wedge v_2 \wedge d_2. \quad (3.24)$$

Минимизация дизъюнктивной формы (3.24) дает:

$$f' = v_1 \wedge v_2. \quad (3.25)$$

Отличительной особенностью функции f' является то, что она существенно зависит только от переменных v_1, v_2 значения и не зависит от переменных d_1, d_2 области определенности. Легко видеть, что правая часть соотношения (3.25) совпадает с булевой функцией значения f в правой части соотношения (3.12).

Таким образом, при кодировании частичных переменных u_1, u_2 парами $(v_1|d_1)$ и $(v_2|d_2)$ булевых переменных частичная операция конъюнкция представляется парой булевых функций, первой из которой является булева конъюнкция (3.25), второй – дизъюнкция (3.23) трех двухлитеральных конъюнкций, зависящих в совокупности от всех четырех булевых переменных v_1, d_1, v_2, d_2 . Область определенности результирующей частично определенной функции в общем случае зависит не только от переменных d_1, d_2 исходных областей определенности, но и от переменных v_1, v_2 исходных значений аргументов. Эта область шире, чем пересечение областей определенности двух исходных частично определенных функций. Это расширение описывается выражением $\neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2$.

Анализ закономерностей (3.12)-(3.21) позволяет сделать следующие выводы.

1. для каждой частичной двухместной операции значение соответствующей частично определенной функции определяется одноименной булевой операцией (для частичной конъюнкции $\&$ – булевой конъюнкцией \wedge , для частичной дизъюнкции $+$ – булевой дизъюнкцией \vee и т.д.);
2. области определенности десяти двухместных частично определенных операций описываются двумя классами булевых формул: а) формулами вида $a_1 \wedge a_2 \vee b_1 \wedge b_2$, где a_1 – положительный или отрицательный литерал, построенный на переменной v_1 , a_2 – положительный или отрицательный литерал, построенный на переменной v_2 ; б) формулами вида $a \wedge b$; формулы вида b используются для частичных операций исключаящее или и эквивалентность, формулы вида a используются для остальных восьми частичных операций;
3. если одна частичная операция определяется путем отрицания другой частичной операции, функция результирующей области определенности не меняется, она остается такой же; так одну результирующую область определенности имеют следующие пары операций: конъюнкция $\&$ и штрих Шеффера \oslash , дизъюнкция $+$ и стрелка Пирса \oslash , прямая импликация \Rightarrow и ее отрицание $\sim\Rightarrow$, обратная импликация \Leftarrow и ее отрицание $\sim\Leftarrow$, исключаящее или \odot и эквивалентность \Leftrightarrow .

Если операнды двухместной частичной операции имеют одну область определенности, описываемую характеристической функцией d , то легко видеть, что результаты всех двухместных частичных операций также имеют область определенности, описываемую d :

$$(v_1|d)\&(v_2|d) = (v_1 \wedge v_2|d), \quad (3.26)$$

$$\sim((v_1|d)\Rightarrow(v_2|d)) = (\neg(v_1 \rightarrow v_2)|d), \quad (3.27)$$

$$\sim((v_1|d)\Leftarrow(v_2|d)) = (\neg(v_1 \leftarrow v_2)|d), \quad (3.28)$$

$$(v_1|d)\odot(v_2|d) = (v_1 \oplus v_2|d), \quad (3.29)$$

$$(v_1|d)+(v_2|d) = (v_1 \vee v_2|d), \quad (3.30)$$

$$(v_1|d) \wp (v_2|d) = (v_1 \downarrow v_2|d), \quad (3.31)$$

$$(v_1|d) \Leftrightarrow (v_2|d) = (v_1 \leftrightarrow v_2|d), \quad (3.32)$$

$$(v_1|d) \Leftarrow (v_2|d) = (v_1 \leftarrow v_2|d), \quad (3.33)$$

$$(v_1|d) \Rightarrow (v_2|d) = (v_1 \rightarrow v_2|d), \quad (3.34)$$

$$(v_1|d) \wp^{\circ} (v_2|d) = (v_1 / v_2|d). \quad (3.35)$$

Таким образом, разнообразие результирующих областей определенности вытекает из различий в областях определенности исходных операндов. Нет различий – нет и разнообразия.

В правых частях равенств (3.12)-(3.21) булевы функции области определенности представлены дизъюнкциями элементарных конъюнкций. В следующих соотношениях характеристические функции области определенности представлены конъюнкциями элементарных дизъюнкций:

$$(v_1|d_1) \& (v_2|d_2) = (v_1 \wedge v_2 | (d_1 \vee d_2) \wedge (\neg v_2 \vee d_1) \wedge (\neg v_1 \vee d_2)), \quad (3.36)$$

$$\sim((v_1|d_1) \Rightarrow (v_2|d_2)) = (\neg(v_1 \rightarrow v_2) | (d_1 \vee d_2) \wedge (v_2 \vee d_1) \wedge (\neg v_1 \vee d_2)), \quad (3.37)$$

$$\sim((v_1|d_1) \Leftarrow (v_2|d_2)) = (\neg(v_1 \leftarrow v_2) | (d_1 \vee d_2) \wedge (\neg v_2 \vee d_1) \wedge (v_1 \vee d_2)), \quad (3.38)$$

$$(v_1|d_1) \wp^{\circ} (v_2|d_2) = (v_1 \oplus v_2 | d_1 \wedge d_2), \quad (3.39)$$

$$(v_1|d_1) + (v_2|d_2) = (v_1 \vee v_2 | (d_1 \vee d_2) \wedge (v_2 \vee d_1) \wedge (v_1 \vee d_2)), \quad (3.40)$$

$$(v_1|d_1) \wp (v_2|d_2) = (v_1 \downarrow v_2 | (d_1 \vee d_2) \wedge (v_2 \vee d_1) \wedge (v_1 \vee d_2)), \quad (3.41)$$

$$(v_1|d_1) \Leftrightarrow (v_2|d_2) = (v_1 \leftrightarrow v_2 | d_1 \wedge d_2), \quad (3.42)$$

$$(v_1|d_1) \Leftarrow (v_2|d_2) = (v_1 \leftarrow v_2 | (d_1 \vee d_2) \wedge (\neg v_2 \vee d_1) \wedge (v_1 \vee d_2)), \quad (3.43)$$

$$(v_1|d_1) \Rightarrow (v_2|d_2) = (v_1 \rightarrow v_2 | (d_1 \vee d_2) \wedge (v_2 \vee d_1) \wedge (\neg v_1 \vee d_2)), \quad (3.44)$$

$$(v_1|d_1) \wp^{\circ} (v_2|d_2) = (v_1 / v_2 | (d_1 \vee d_2) \wedge (\neg v_2 \vee d_1) \wedge (\neg v_1 \vee d_2)). \quad (3.45)$$

Если $d_1 = d_2 = 1$, то частичные операции сводятся к полностью определенным булевым операциям. В частности, частичная конъюнкция $(v_1|1) \& (v_2|1)$ преобразуется в полностью определенную булеву конъюнкцию $v_1 \wedge v_2$ следующим образом: $(v_1|1) \& (v_2|1) = (v_1 \wedge v_2 | 1 \vee \neg v_1 \wedge 1 \vee \neg v_2 \wedge 1) = (v_1 \wedge v_2 | 1) = v_1 \wedge v_2$.

Некоторые двухместные частичные операции легко обобщаются на многоместные операции. Так двухместная частичная конъюнкция

юнкция (3.12) при увеличении числа аргументов с 2 до n трансформируется в:

$$\& (v_i | d_i) = \left(\bigwedge_{i=1}^n v_i \mid \left(\bigwedge_{i=1}^n d_i + \bigvee_{i=1}^n \neg v_i \wedge d_i \right) \right).$$

В правой части равенства функция значения представлена n -местной конъюнкцией переменных v_i . Функция области определенности представлена n -местной конъюнкцией переменных d_i и n -местной дизъюнкцией конъюнкций $\neg v_i \wedge d_i$. Двухместная частичная дизъюнкция (3.16) трансформируется в n -местную дизъюнкцию:

$$\oplus (v_i | d_i) = \left(\bigvee_{i=1}^n v_i \mid \left(\bigwedge_{i=1}^n d_i + \bigvee_{i=1}^n v_i \wedge d_i \right) \right).$$

Структура правой части равенства та же, что и структура правой части предыдущего равенства.

3.5. Смешанная алгебра

Структура смешанной алгебры представлена на рис.3.4. Смешанная алгебра строится путем взаимодействия частичной алгебры, описанной в главе 2, с классической булевой алгеброй. Взаимодействие происходит посредством VCR формы представления частичных переменных парами булевых переменных и представления частично определенных функций посредством частичных операций, выполняемых над парами булевых функций.

Элементами смешанной алгебры являются:

- множество значений $M = \{0, 1, dc\}$;
- частичные переменные y_1, \dots, y_m ;
- булевы переменные x_1, \dots, x_n ;
- кодирование частичных переменных y парами булевых переменных $(v|d)$;
- частичные одноместные и двухместные операции;
- смешанные одноместные и двухместные операции;
- булевы одноместные и двухместные операции;
- частичные функции от частичных переменных;

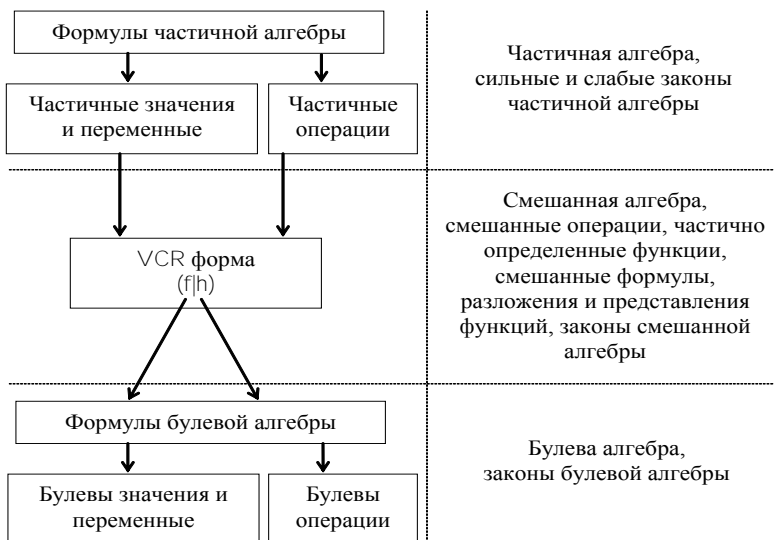


Рисунок 3.4 – Структура смешанной алгебры

- частично определенные функции от булевых переменных в VCR форме $(f|h)$;
- булевы функции от булевых переменных;
- формулы частичной алгебры;
- формулы булевой алгебры;
- формулы смешанной алгебры.

Формулы смешанной алгебры строятся на базе формул частичной алгебры, формул булевой алгебры, VCR формы. Законы смешанной алгебры строятся на базе законов частичной и булевой алгебр. На базе законов смешанной алгебры строятся новые разложения и формы представления частично определенных функций. Новые разложения частично определенных функций позволяют построить новые ранее не известные разложения булевых функций.

Преобразование формул смешанной алгебры базируется на использовании законов смешанной алгебры. Принципиальное значение имеют два основных направления преобразования:

- любая формула смешанной алгебры преобразуется в VCR форму некоторой частично определенной функции, пред-

ставленной парой булевых функций, путем использования соответствующих законов смешанной алгебры;

- VCR форма частично определенной функции преобразуется в формулу смешанной алгебры путем использования разнообразных разложений.

Пример 3.2. Данный пример является продолжением примера 3.1, в котором все частичные, частично определенные и булевы функции представлены в табличной форме. В данном примере все функции представляются в алгебраической форме. Выполним построение и преобразование частичной формулы в смешанную формулу и далее в VCR форму частично определенной функции. Сначала представим частичную функцию Ψ в совершенной дизъюнктивно-конъюнктивной форме (2.34)-(2.37):

$$\Psi = (y_1 \& \sim y_2 + y_1 \& y_2 + y_1 \& \# y_2) + \sim(\sim y_1 \& \sim y_2) \& dc.$$

Затем полученную форму преобразуем к дизъюнктивно-конъюнктивной форме (2.49)-(2.52):

$$\Psi = (y_1) + \sim(\sim y_1 \& \sim y_2) \& dc.$$

После кодирования частичных переменных парами булевых переменных в VCR форме $y_1 = (v_1 | d_1)$, $y_2 = (v_2 | d_2)$ получаем смешанную формулу для частично определенной функции ψ :

$$\psi = (v_1 | d_1) + \sim(\sim(v_1 | d_1) \& \sim(v_2 | d_2)) \& (0 | 0) = \\ (v_1 | d_1) + ((v_1 | d_1) + (v_2 | d_2)) \& (0 | 0).$$

Подставляя вместо переменных v_1 , d_1 , v_2 , d_2 булевы функции f_1 , h_1 , f_2 , h_2 , преобразованные из табл.3.4 в формульный вид, получаем смешанную формулу для частично определенной функции ψ' :

$$\psi' = (x_2 \vee \neg x_3 | x_1) + ((x_2 \vee \neg x_3 | x_1) + (\neg x_1 | x_2 \vee x_3)) \& (0 | 0).$$

Используя законы (3.12), (3.16) для частичных операций $\&$ и $+$, получаем:

$$\begin{aligned} \psi' &= (x_2 \vee \neg x_3 | x_1) + \\ &(\neg x_1 \vee x_2 \vee x_3 | x_1 \wedge x_2 \vee x_1 \wedge x_3 \vee x_1 \wedge \neg x_3 \vee \neg x_1 \wedge x_2 \vee \neg x_1 \wedge x_3) \& (0 | 0) = \\ &(x_2 \vee \neg x_3 | x_1) + (\neg x_1 \vee x_2 \vee \neg x_3 | x_1 \vee x_2 \vee x_3) \& (0 | 0) = \\ &(x_2 \vee \neg x_3 | x_1) + (0 | x_1 \wedge \neg x_2 \wedge x_3) = \\ &(x_2 \vee \neg x_3 | x_1). \end{aligned} \tag{3.46}$$

Очевидно, что смешанная формула (3.46) описывает ту же частично определенную функцию ψ' , что и табл.3.4. \square

3.6. Обзор законов смешанной алгебры

Законы смешанной алгебры отражают либо свойства VCR формы представления частично определенных функций либо свойства частичных операций над VCR формами.

Важнейшим свойством VCR формы ($\vee d$) является допустимость вариации булевой функции значения \vee в диапазоне от $\vee \wedge d$ до $\vee \neg d$. Преобразование функции \vee в указанном диапазоне может преследовать различные цели. Прежде всего, это цель упрощения или минимизации представления функции \vee , напрямую связанная с уменьшением стоимости аппаратной или программной реализации цифровых и программных систем. Законы преобразования в этом случае могут учитывать, но могут и не учитывать, в какой из исходных форм представлена функция \vee . Целью преобразования может быть также улучшение временных или энергетических параметров цифровых систем, а также улучшение качества других технических параметров.

Законы смешанной алгебры, отражающие свойства частичных операций, конкретизируют, прежде всего, сильные и слабые законы частичной алгебры в расчете на VCR форму. Другими словами, частичная алгебра, являющаяся одной из составляющих смешанной алгебры, служит богатым источником законов смешанной алгебры. При замене частичных переменных в законах частичной алгебры (см. разделы 2.5, 2.6) на пары булевых переменных в VCR форме получаем законы смешанной алгебры, которые могут выполняться как безусловно так и условно. При этом оказывается, что законы смешанной алгебры, построенные на базе сильных законов частичной алгебры, выполняются безусловно, а законы смешанной алгебры, построенные на базе слабых законов частичной алгебры, выполняются условно.

Смешанная алгебра, имеющая существенные отличия от частичной и булевой алгебр, имеет также свои собственные законы. Эти законы являются базой для построения новых ранее не известных разложений и форм представления частично определенных

функций. Эти разложения и формы представления мы изучим в последующих главах. При этом покажем, что новые результаты в смешанной алгебре приводят также к новым результатам в булевой алгебре.

3.7. Законы преобразования функции значения

Важнейшими законами смешанной алгебры являются законы замены булевой функции v в паре (v/d) любой другой булевой функцией v' такой, что $v' \wedge d = v \wedge d$. Этот закон устанавливает требование равенства функций v и v' в области определенности, характеризуемой функцией d . Рассмотрим несколько важнейших случаев преобразования функции v , связанных с различными формами представления функций v и d :

1. функции v и d представлены простейшими формулами с использованием булевых операций;
2. функция v имеет общий вид, функция d имеет конкретный вид;
3. функции v и d представлены в табличной форме;
4. функции v и d представлены стандартными алгебраическими формами;
5. функции v и d представлены диаграммами решений в графовой форме.

Функции v и d представлены простейшими формулами. Следующие законы обеспечивают упрощение булевой функции значения в специфических конфигурациях VCR форм:

$$(\neg v|v) = (0|v), \quad (3.47)$$

$$(v \wedge \neg d|d) = (0|d), \quad (3.48)$$

$$(v|\neg v \wedge d) = (0|\neg v \wedge d), \quad (3.49)$$

$$(v|v) = (1|v), \quad (3.50)$$

$$(v \vee d|d) = (1|d), \quad (3.51)$$

$$(v|v \wedge d) = (1|v \wedge d), \quad (3.52)$$

$$(v \wedge d|d) = (v|d), \quad (3.53)$$

$$(v \vee \neg d|d) = (v|d). \quad (3.54)$$

Законы (3.47)-(3.49) сводят функцию V к константе 0. Законы (3.50)-(3.52) сводят функцию V к константе 1. Законы (3.53)-(3.54) упрощают формульное представление функции значения до V .

Функция V имеет общий вид, функция d имеет конкретный вид. Пусть булева функция $V(x_1, \dots, x_n)$ общего вида зависит от переменных x_1, \dots, x_n . Два следующих закона устанавливают механизм преобразования булевой функции значения при условии, что булева функция области определенности описывается некоторой переменной $d=x_i$ или ее отрицанием $d=\neg x_i$:

$$(V|\neg x_i) = (V_{x_i=0}|\neg x_i), \quad (3.55)$$

$$(V|x_i) = (V_{x_i=1}|x_i), \quad (3.56)$$

где остаточные функции $V_{x_i=0}$ и $V_{x_i=1}$ получаются в результате подстановки вместо переменной x_i константных значений 0 и 1 соответственно и называются *отрицательным* и *положительным кофактором* булевой функции V по переменной x_i . Переменная x_i , являющаяся существенной в функции V , становится несущественной в кофакторах $V_{x_i=0}$ и $V_{x_i=1}$. Законы (3.55), (3.56) допускают использование кофакторов вместо полной функции значения при установлении ограничения на вид функции области определенности. Кофакторы $V_{x_i=0}$ и $V_{x_i=1}$ используются в разложениях Шеннона и Давио и широко применяются на практике при решении логических задач.

Пример 3.2. В табл.3.9 определена булева функция V , зависящая от трех булевых переменных x_1, x_2, x_3 . Функция описывается

Таблица 3.9

Пример кофакторов $V_{x_1=0}$ и $V_{x_1=1}$

| № | x_1 | x_2 | x_3 | Функция V | Функция $d=\neg x_1$ | Кофактор $V_{x_1=0}$ | Функция $d=x_1$ | Кофактор $V_{x_1=1}$ |
|---|-------|-------|-------|-------------|----------------------|----------------------|-----------------|----------------------|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

формулой $V = \neg X_1 \wedge X_2 \wedge \neg X_3 \vee X_1 \wedge X_3$. Подставляя в формулу значение переменной $X_1=0$, находим, что отрицательный кофактор функции V описывается формулой $V_{X_1=0} = X_2 \wedge \neg X_3$. Аналогично находим положительный кофактор, описываемый формулой $V_{X_1=1} = X_3$. \square

Следующие два закона допускают замену одной булевой переменной X_i другой переменной X_j или ее отрицанием $\neg X_j$ в формуле функции значения V , если область определенности описывается посредством булевой операции исключающее или \oplus или посредством булевой операции эквивалентность \leftrightarrow :

$$(V|X_i \oplus X_j) = (V_{X_i=\neg X_j} | X_i \oplus X_j), \quad (3.57)$$

$$(V|X_i \leftrightarrow X_j) = (V_{X_i=X_j} | X_i \leftrightarrow X_j). \quad (3.58)$$

Булевы функции $V_{X_i=\neg X_j}$ и $V_{X_i=X_j}$, являющиеся остаточными от булевой функции V при замене переменной X_i отрицанием переменной $\neg X_j$ или самой переменной X_j , назовем *исключающим отрицательным кофактором* и *исключающим положительным кофактором* соответственно. Замена X_i на X_j не является эквивалентной замене X_j на X_i . Кофакторы $V_{X_i=\neg X_j}$ и $V_{X_i=X_j}$ отличаются от кофакторов $V_{X_j=\neg X_i}$ и $V_{X_j=X_i}$ уже тем, что в двух первых несущественной становится переменная X_i , в то время как в двух других несущественной становится переменная X_j . Заметим, что в обозначениях кофакторов знак $=$ используется не как знак равенства, но как направление замены.

Кофакторы $V_{X_i=\neg X_j}$, $V_{X_i=X_j}$, $V_{X_j=\neg X_i}$ и $V_{X_j=X_i}$ не использовались в теории булевых функций. Автору не известно разложений функций, в основе которых лежит использование этих кофакторов.

При построении функциональных диаграмм решений используется еще один кофактор вида

$$V_{X_i=0 \oplus 1} = V_{X_i=0} \oplus V_{X_i=1}, \quad (3.59)$$

построенный из кофакторов $V_{X_i=0}$ и $V_{X_i=1}$ посредством операции исключающее или. Следует заметить, что в общем случае для кофакторов выполняются неравенства:

$$V_{X_i=\neg X_j} \neq V_{X_i=0} \oplus V_{X_i=1},$$

$$V_{X_i=X_j} \neq V_{X_i=0} \leftrightarrow V_{X_i=1},$$

Таблица 3.10

Пример кофакторов $V_{x_1=\neg x_2}$, $V_{x_2=\neg x_1}$, $V_{x_1=x_2}$, $V_{x_2=x_1}$

| № | x_1 | x_2 | x_3 | v | $d=x_1\oplus x_2$ | $V_{x_1=\neg x_2}$ | $V_{x_2=\neg x_1}$ | $d=x_1\leftrightarrow x_2$ | $V_{x_1=x_2}$ | $V_{x_2=x_1}$ |
|---|-------|-------|-------|-----|-------------------|--------------------|--------------------|----------------------------|---------------|---------------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

Пример 3.3. В табл.3.10 определена та же булева функция $V=\neg x_1 \wedge x_2 \wedge \neg x_3 \vee x_1 \wedge x_3$, что и в примере 3.2. Подставляя в формулу для функции V значение переменной $x_1=\neg x_2$ и преобразуя ее, находим, что исключающий отрицательный кофактор описывается формулой $V_{x_1=\neg x_2}=x_2 \oplus x_3$. Аналогично находим, что исключающий положительный кофактор описывается формулой $V_{x_1=x_2}=x_2 \wedge x_3$. Подставляя в формулу для функции V значение переменной $x_2=\neg x_1$ и преобразуя ее, находим, что исключающий отрицательный кофактор описывается формулой $V_{x_2=\neg x_1}=x_1 \leftrightarrow x_3$. Аналогично находим, что при подстановке $x_2=x_1$ исключающий положительный кофактор описывается формулой $V_{x_2=x_1}=x_1 \wedge x_3$. Из двух кофакторов $V_{x_1=0}$ и $V_{x_1=1}$ получаем кофактор $V_{x_1=0 \oplus 1}=(x_2 \wedge \neg x_3) \oplus (x_3)=x_3 \vee x_2 \wedge \neg x_3=x_2 \vee x_3$. □

Следующие два закона допускают замену одной булевой переменной x_i целыми выражениями в формуле функции V , если область определенности описывается посредством булевой n -местной операции исключающее или \oplus :

$$(V|x_i \oplus \dots \oplus x_n) = (V_{x_i=\neg x_n}|x_i \oplus \dots \oplus x_n), \quad (3.60)$$

$$(V|\neg(x_i \oplus \dots \oplus x_n)) = (V_{x_i=x_n}|\neg(x_i \oplus \dots \oplus x_n)). \quad (3.61)$$

Булева функция $V_{x_i=\neg x_n}$ является остаточной от булевой функции V при замене переменной x_i выражением $\neg(x_{i+1} \oplus \dots \oplus x_n)$, а булева функция $V_{x_i=x_n}$ является остаточной от булевой функции V при замене переменной x_i выражением $x_{i+1} \oplus \dots \oplus x_n$. Остаточную функцию $V_{x_i=\neg x_n}$ назовем *полным исключающим отрицательным кофактором*, а остаточную функцию $V_{x_i=x_n}$ назовем *полным исключающим положительным кофактором*.

Таблица 3.11

Пример кофакторов $V_{x_1=-x_2}, V_{x_2=-x_1}, V_{x_1=x_2}, V_{x_2=x_1}$

| № | x_1 | x_2 | x_3 | v | $d=x_1 \oplus x_2 \oplus x_3$ | $V_{x_1=-x_1}$ | $d=\neg(x_1 \oplus x_2 \oplus x_3)$ | $V_{x_1=x_1}$ |
|---|-------|-------|-------|-----|-------------------------------|----------------|-------------------------------------|---------------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Пример 3.4. В табл.3.11 определена та же булева функция $V=\neg x_1 \wedge x_2 \wedge \neg x_3 \vee x_1 \wedge x_3$, что и в примере 3.2. Подставляя в формулу для функции V значение переменной $x_1=\neg(x_2 \oplus x_3)$ и преобразуя ее, находим, что полный исключающий отрицательный кофактор описывается формулой $V_{x_1=-x_1}=x_2$. Подставляя в формулу значение переменной $x_1=x_2 \oplus x_3$, находим полный исключающий положительный кофактор $V_{x_1=x_1}=\neg x_2 \wedge x_3$. □

Функции v и d представлены в табличной форме. Рассмотрим законы (3.55)-(3.56) применительно к совместному табличному представлению функций v и d . Первые столбцы таблицы соответствуют булевым переменным $x_i, i=1, \dots, n$, последующие столбцы соответствуют булевым функциям, включая v и все кофакторы. Строки таблицы соответствуют наборам значений булевых переменных. Функция $d=\neg x_i$ описывается столбцом, в строке k которого находится значение, инверсное значению, стоящему на пересечении строки k и столбца переменной x_i . Функция $d=x_i$ описывается столбцом, идентичным столбцу, описывающему переменную x_i . Построение функции кофактора $V_{x_i=0}$ выполняется следующим образом. На наборе $(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n)$ значений переменных, где $x_i=0$, значение кофактора $V_{x_i=0}$ принимается равным значению функции v на этом же наборе, поскольку здесь $d=\neg x_i=1$. На наборе $(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n)$ значений переменных, где $x_i=1$, значение кофактора $V_{x_i=0}$ принимается равным значению функции v на наборе $(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n)$ поскольку здесь $d=\neg x_i=0$ и значение функции может

быть произвольным. Благодаря такому построению кофактора переменная x_i становится несущественной, а функция V упрощается. Аналогичным образом происходит построение кофактора $V_{x_i=1}$ при $d=x_i$.

Пример 3.5. В табл.3.9 светлая заливка показывает кофактор $V_{x_1=0}$, темная заливка показывает кофактор $V_{x_1=1}$. Переменная x_1 не находится ни под светлой ни под темной заливкой, так как она является несущественной для обоих кофакторов. Значение кофактора $V_{x_1=0}$ на четырех верхних наборах, для которых $x_1=0$, совпадают со значениями функции V , так как на этих наборах функция области определенности $d=\neg x_1=1$. Значение кофактора на четырех нижних наборах, для которых $x_1=1$, может быть выбрано произвольным образом, поскольку на этих наборах функция области определенности $d=\neg x_1=0$. Мы выбираем значения на четырех нижних наборах идентичными значениям на четырех верхних наборах. Именно благодаря этому переменная x_1 становится несущественной. Кофактор $V_{x_1=1}$ строится аналогично кофактору $V_{x_1=0}$. □

Рассмотрим теперь законы (3.57)-(3.58) применительно к совместному табличному представлению функций V и d . Функция $d=x_i \oplus x_j$ описывается столбцом, в строке k которого находится 1, если в этой строке переменные x_i и x_j имеют различные значения, и находится 0, если переменные имеют одинаковые значения. Функция $d=x_i \leftrightarrow x_j$ описывается столбцом, элементом которого является 1, если переменные x_i и x_j имеют одинаковые значения, и является 0, если переменные имеют различные значения.

Построение функции кофактора $V_{x_i=\neg x_j}$ выполняется следующим образом. На наборе $a=(a_1, \dots, a_i, \dots, a_j, \dots, a_n)$ значений переменных, на котором $a_i \neq a_j$, значение кофактора $V_{x_i=\neg x_j}$ принимается равным значению функции $V(a)$ на этом же наборе поскольку здесь $d=x_i \oplus x_j=1$. На наборе $(a_1, \dots, a_i, \dots, a_j, \dots, a_n)$ значений переменных, на котором $a_i = a_j$, значение кофактора $V_{x_i=\neg x_j}$ принимается равным значению функции V на наборе $(a_1, \dots, \neg a_i, \dots, a_j, \dots, a_n)$ поскольку здесь $d=x_i \oplus x_j=0$ и значение функции может быть произвольным. Благодаря такому построению кофактора переменная x_i становится несуще-

ственной, а функция V упрощается. Аналогичным образом происходит построение кофактора $V_{x_i=x_j}$ при $d=x_i \leftrightarrow x_j$.

Пример 3.6. В табл.3.7 светлая заливка показывает кофакторы $V_{x_1=-x_2}$ и $V_{x_2=-x_1}$ темная заливка показывает кофакторы $V_{x_1=x_2}$ и $V_{x_2=x_1}$. Кофакторы $V_{x_1=-x_2}$ и $V_{x_1=x_2}$ зависят от переменных x_2 и x_3 . Кофакторы $V_{x_2=-x_1}$ и $V_{x_2=x_1}$ зависят от переменных x_1 и x_3 . Значения кофактора $V_{x_1=-x_2}$ совпадают с четырьмя средними значениями функции V на тех наборах значений аргументов, в которых переменные x_1 и x_2 имеют различные значения. В этом случае наборы принадлежат области определенности, описываемой характеристической булевой функцией $d=x_1 \oplus x_2=1$. На двух верхних и двух нижних наборах переменные x_1 и x_2 имеют одинаковые значения, следовательно $d=x_1 \oplus x_2=0$. На этих наборах значения кофактора $V_{x_1=-x_2}$ произвольные. Выбираем такие значения, чтобы они совпадали со средними значениями функции, и переменная x_1 стала несущественной. Отличие кофактора $V_{x_2=-x_1}$ от $V_{x_1=-x_2}$ состоит в том, что на двух верхних и двух нижних наборах выбираются такие значения функции, что несущественной становится переменная x_2 . Кофакторы $V_{x_1=x_2}$ и $V_{x_2=x_1}$ строятся аналогично кофакторам $V_{x_1=-x_2}$ и $V_{x_2=-x_1}$. \square

Пример 3.7. В табл.3.8 светлая заливка показывает кофактор $V_{x_1=-x_3}$, темная заливка показывает кофактор $V_{x_1=x_3}$. Кофакторы зависят от переменных x_2 и x_3 . Значения кофактора $V_{x_1=-x_2}$ совпадают со значениями функции V на четырех наборах значений аргументов, которые отмечены светлой заливкой, для которых $d=x_1 \oplus x_2 \oplus x_3=1$. На других наборах значение кофактора $V_{x_1=-x_2}$ произвольно. Выбираем такие значения, чтобы они совпадали с четырьмя первыми значениями, и переменная x_1 стала несущественной. Кофактор $V_{x_1=x_2}$ строится аналогично кофактору $V_{x_1=-x_2}$. \square

Функции V и d представлены стандартными алгебраическими формами. К стандартным алгебраическим формам относятся дизъюнктивные нормальные формы, конъюнктивные нормальные формы, формы Рида-Маллера, частным случаем которых является полином Жегалкина. Преобразование функции значения выполняется, прежде всего, в случаях, когда функции V и d представлены в одной и той же форме. Так, функция V может быть минимизирована методом Квайна-Мак-Класки, в котором построение множества простых

импликант выполняется на булевой функции $\mathcal{W}\neg d$, а нахождение минимального покрытия выполняется для конституент 1 функции $\mathcal{V}\wedge d$.

Функции \mathcal{V} и d представлены диаграммами решений в графовой форме. Наибольший интерес представляют сокращенные упорядоченные двоичные диаграммы решений и положительные и отрицательные функциональные диаграммы решений.

3.8. Сильные законы частичной алгебры в смешанной алгебре

Сильные законы частичной алгебры действуют в полном объеме и в смешанной алгебре. Переход от законов частичной алгебры к законам смешанной алгебры осуществляется посредством замены частичных переменных парами булевых переменных в VCR форме, причем каждой частичной переменной ставится в соответствие своя область определенности. Так законы ассоциативности для частичных операций $\&$ и $+$ приобретают вид:

$$(\mathcal{V}_1|d_1)\&((\mathcal{V}_2|d_2)\&(\mathcal{V}_3|d_3)) = ((\mathcal{V}_1|d_1)\&(\mathcal{V}_2|d_2))\&(\mathcal{V}_3|d_3), \quad (3.62)$$

$$(\mathcal{V}_1|d_1)+((\mathcal{V}_2|d_2)+(\mathcal{V}_3|d_3)) = ((\mathcal{V}_1|d_1)+(\mathcal{V}_2|d_2))+(\mathcal{V}_3|d_3). \quad (3.63)$$

Законы коммутативности для частичных операций $\&$ и $+$:

$$(\mathcal{V}_1|d_1)\&(\mathcal{V}_2|d_2) = (\mathcal{V}_2|d_2)\&(\mathcal{V}_1|d_1), \quad (3.64)$$

$$(\mathcal{V}_1|d_1)+(\mathcal{V}_2|d_2) = (\mathcal{V}_2|d_2)+(\mathcal{V}_1|d_1). \quad (3.65)$$

Законы дистрибутивности для частичных операций $\&$ и $+$:

$$(\mathcal{V}_1|d_1)\&((\mathcal{V}_2|d_2)+(\mathcal{V}_3|d_3)) = ((\mathcal{V}_1|d_1)\&(\mathcal{V}_2|d_2))+((\mathcal{V}_1|d_1)\&(\mathcal{V}_3|d_3)), \quad (3.66)$$

$$(\mathcal{V}_1|d_1)+((\mathcal{V}_2|d_2)\&(\mathcal{V}_3|d_3)) = ((\mathcal{V}_1|d_1)+(\mathcal{V}_2|d_2))\&((\mathcal{V}_1|d_1)+(\mathcal{V}_3|d_3)). \quad (3.67)$$

Законы идентичности для частичных операций $\&$ и $+$ с использованием констант (0|1) и (1|1) (напомним, что константа (0|1) представляет полностью определенный 0, а константа (1|1) представляет полностью определенную 1):

$$(\mathcal{V}|d)\&(1|1) = (\mathcal{V}|d), \quad (3.68)$$

$$(\mathcal{V}|d)+(0|1) = (\mathcal{V}|d). \quad (3.69)$$

Закон аннигилятора для частичных операций $\&$ и $+$:

$$(\nu|d) \& (0|1) = (0|1). \quad (3.70)$$

$$(\nu|d) + (1|1) = (1|1). \quad (3.71)$$

Законы *идемпотентности* для частичных операций $\&$ и $+$:

$$(\nu|d) \& (\nu|d) = (\nu|d), \quad (3.72)$$

$$(\nu|d) + (\nu|d) = (\nu|d). \quad (3.73)$$

Законы *поглощения* для операций $\&$ и $+$:

$$(\nu_1|d_1) \& ((\nu_1|d_1) + (\nu_2|d_2)) = (\nu_1|d_1), \quad (3.74)$$

$$(\nu_1|d_1) + ((\nu_1|d_1) \& (\nu_2|d_2)) = (\nu_1|d_1). \quad (3.75)$$

Закон *двойного отрицания*:

$$\sim \sim (\nu|d) = (\nu|d). \quad (3.76)$$

Законы *де Моргана* для операций $\&$ и $+$:

$$\sim (\nu_1|d_1) \& \sim (\nu_2|d_2) = \sim ((\nu_1|d_1) + (\nu_2|d_2)), \quad (3.77)$$

$$\sim (\nu_1|d_1) + \sim (\nu_2|d_2) = \sim ((\nu_1|d_1) \& (\nu_2|d_2)). \quad (3.78)$$

Следующие законы позволяют выразить другие частичные операции, определенные в табл. 3.4, через операции \sim , $\&$ и $+$:

$$(\nu_1|d_1) \Rightarrow (\nu_2|d_2) = \sim (\nu_1|d_1) + (\nu_2|d_2), \quad (3.79)$$

$$(\nu_1|d_1) \Leftarrow (\nu_2|d_2) = (\nu_1|d_1) + \sim (\nu_2|d_2), \quad (3.80)$$

$$(\nu_1|d_1) \overset{\circ}{\times} (\nu_2|d_2) = (\nu_1|d_1) \& \sim (\nu_2|d_2) + \sim (\nu_1|d_1) \& (\nu_2|d_2), \quad (3.81)$$

$$(\nu_1|d_1) \Leftrightarrow (\nu_2|d_2) = (\nu_1|d_1) \& (\nu_2|d_2) + \sim (\nu_1|d_1) \& \sim (\nu_2|d_2), \quad (3.82)$$

$$(\nu_1|d_1) \overset{\circ}{\delta} (\nu_2|d_2) = \sim ((\nu_1|d_1) \& (\nu_2|d_2)), \quad (3.83)$$

$$(\nu_1|d_1) \overset{\circ}{\eta} (\nu_2|d_2) = \sim ((\nu_1|d_1) + (\nu_2|d_2)). \quad (3.84)$$

Законы (3.79)-(3.84) аналогичны законам булевой алгебры при замене булевых переменных парами булевых переменных в VCR форме и замене булевых операций аналогичными частичными операциями.

Другие законы для частичных операций $\&$, $+$, \Rightarrow и $\overset{\circ}{\times}$ с использованием констант $(0|0)$, $(1|1)$:

$$(\nu|d) + (\nu|d) \& (0|0) = (\nu|d), \quad (3.85)$$

$$(\nu|d) + \sim (\nu|d) \& (0|0) = (\nu|d) + (0|0), \quad (3.86)$$

$$(0|1) \Rightarrow (\nu|d) = (1|1), \quad (3.87)$$

$$(\nu|d) \Rightarrow (\nu|d) = (1|1), \quad (3.88)$$

$$(\psi d) \Rightarrow (1|1) = (1|1), \quad (3.89)$$

$$(\psi_1|d_1) \Rightarrow (\psi_2|d_2) = \sim(\psi_2|d_2) \Rightarrow \sim(\psi_1|d_1), \quad (3.90)$$

$$(\psi d) \star (1|1) = \sim(\psi d). \quad (3.91)$$

Проиллюстрируем доказательство законов на соотношении (3.86). Преобразуем левую и правую части равенства (3.86) к VCR форме посредством законов (3.12)-(3.21) смешанной алгебры и законов булевой алгебры. Левая часть:

$$\begin{aligned} (\psi d) + \sim(\psi d) \&(0|0) &= (\psi d) + (\sim\psi d) \&(0|0) = \\ (\psi d) + \sim(\sim\psi 0|d \wedge 0 \vee \psi \wedge d \vee 1 \wedge 0) &= (\psi d) + (0|\psi \wedge d) = \\ (\psi 0|d \wedge \psi \wedge d \vee \psi \wedge d \vee 0 \wedge \psi \wedge d) &= (\psi \vee \wedge d). \end{aligned}$$

Правая часть:

$$(\psi d) + (0|0) = (\psi 0|d \wedge 0 \vee \psi \wedge d \vee 0 \wedge 0) = (\psi \vee \wedge d).$$

Поскольку левая и правая части функционально эквивалентны, равенство (3.86) выполняется.

3.9. Слабые законы частичной алгебры в смешанной алгебре

Слабые законы частичной алгебры приобретают несколько иное звучание в смешанной алгебре. Слабость частичных законов проявляется в смешанной алгебре в наличии тех или иных ограничений на области определенности операндов, при которых законы выполняются. Для каждого из слабых законов мы точно определим эти ограничения. При равенстве областей определенности операндов законы выполняются для любых функций значения.

Так законы *исключенного третьего* для частичных операций $\&$ и $+$ приобретают вид:

$$(\psi d) \& \sim(\psi d) = (0|d), \quad (3.92)$$

$$(\psi d) + \sim(\psi d) = (1|d). \quad (3.93)$$

Равенство результата конъюнкции константе 0 и результата дизъюнкции константе 1 выполняется не на всей области, а только на области определенности, описываемой характеристической функцией d . В результате правая часть законов не представлена

полностью определенными константами (0|1) и (1|1), но представлена частично определенными константами (0| d) и (1| d) с областью определенности d . Доказательство законов базируется на использовании свойств частичных операций $\&$ и $+$ и законов исключенного третьего булевой алгебры.

Закон *склеивания конъюнкций* при равенстве областей определенности исходных операндов $d_1 = d_2 = d$:

$$(v_1|d)\&(v_2|d) + (v_1|d)\&\sim(v_2|d) = (v_1|d). \quad (3.94)$$

Доказательство закона (3.94) базируется на использовании аналогичного закона склеивания конъюнкций булевой алгебры.

Закон *склеивания конъюнкций* при различных областях определенности операндов

$$(v_1|d_1)\&(v_2|d_2) + (v_1|d_1)\&\sim(v_2|d_2) = (v_1|d_1). \quad (3.95)$$

Выполняется тогда, когда

$$d_1 \rightarrow (d_2 \vee \neg v_1). \quad (3.96)$$

Для доказательства закона преобразуем левую часть равенства (3.95) к VCR форме:

$$\begin{aligned} & (v_1|d_1)\&(v_2|d_2) + (v_1|d_1)\&\sim(v_2|d_2) = \\ & (v_1 \wedge v_2 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2) + (v_1 \wedge \neg v_2 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee v_2 \wedge d_2) = \\ & (v_1 \wedge v_2 \vee v_1 \wedge \neg v_2 | (d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2) \wedge (d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee v_2 \wedge d_2)) \\ & \quad \vee (v_1 \wedge v_2) \wedge (d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2) \\ & \quad \vee (v_1 \wedge \neg v_2) \wedge (d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee v_2 \wedge d_2) = \\ & (v_1 | d_1 \wedge d_2 \vee \neg v_1 \wedge d_1) = (v_1 | d_1 \wedge (d_2 \vee \neg v_1)). \end{aligned}$$

Булевы функции значения, представленные переменной v_1 , идентичны в левой и правой частях равенства (3.95). Функцию области определенности $d_1 \wedge (d_2 \vee \neg v_1)$ можно заменить на d_1 при выполнении условия (3.96), а это и доказывает соотношение (3.95).

Закон *поглощения литерала* при равенстве областей определенности операндов:

$$(v_1|d) + \sim(v_1|d)\&(v_2|d) = (v_1|d) + (v_2|d). \quad (3.97)$$

Доказательство закона (3.97) базируется на использовании аналогичного закона поглощения литерала в булевой алгебре.

Закон *поглощения литерала* при различных областях определенности операндов:

$$(v_1|d_1)+\sim(v_1|d_1)\&(v_2|d_2) = (v_1|d_1)+(v_2|d_2) \quad (3.98)$$

имеет место при условии выполнения

$$d_1\wedge d_2\vee v_1\wedge d_1\rightarrow v_2\wedge d_2. \quad (3.99)$$

С целью доказательства закона преобразуем левую и правую части равенства (3.98) к VCR форме. Левая часть:

$$\begin{aligned} (v_1|d_1)+\sim(v_1|d_1)\&(v_2|d_2) &= (v_1|d_1)+(\neg v_1\wedge v_2|d_1\wedge d_2\vee v_1\wedge d_1\neg v_2\wedge d_2) = \\ (v_1\neg v_1\wedge v_2|d_1\wedge d_2\vee v_1\wedge d_1) &= (v_1\vee v_2|d_1\wedge d_2\vee v_1\wedge d_1). \end{aligned}$$

Правая часть:

$$(v_1|d_1)+(v_2|d_2) = (v_1\vee v_2|d_1\wedge d_2\vee v_1\wedge d_1\vee v_2\wedge d_2).$$

Булевы функции значения, представленные формулой $v_1\vee v_2$, идентичны в левой и правой частях. Функцию области определенности $d_1\wedge d_2\vee v_1\wedge d_1\vee v_2\wedge d_2$ можно заменить на $d_1\wedge d_2\vee v_1\wedge d_1$ при выполнении условия (3.99), а это и доказывает соотношение (3.98).

Закон *определения частичной дизъюнкции через исключаящее или* при равенстве областей определенности операндов:

$$(v_1|d)+(v_2|d) = (v_1|d)\dot{\times}(v_2|d)\dot{\times}(v_1|d)\&(v_2|d). \quad (3.100)$$

Доказательство закона (3.100) базируется на использовании аналогичного закона булевой алгебры.

Закон *определения частичной дизъюнкции через исключаящее или* при различных областях определенности операндов

$$(v_1|d_1)+(v_2|d_2) = (v_1|d_1)\dot{\times}(v_2|d_2)\dot{\times}(v_1|d_1)\&(v_2|d_2) \quad (3.101)$$

выполнения тогда, когда выполняется

$$d_1\wedge d_2\rightarrow v_1\wedge d_1\vee v_2\wedge d_2. \quad (3.102)$$

Для доказательства закона преобразуем правую часть равенства (3.101) к VCR форме:

$$\begin{aligned} (v_1|d_1)\dot{\times}(v_2|d_2)\dot{\times}(v_1|d_1)\&(v_2|d_2) = \\ (v_1\oplus v_2|d_1\wedge d_2)\dot{\times}(v_1\wedge v_2|d_1\wedge d_2\neg v_1\wedge d_1\neg v_2\wedge d_2) = \end{aligned}$$

$$\begin{aligned} & (v_1 \oplus v_2 \oplus (v_1 \wedge v_2)) | (d_1 \wedge d_2) \wedge (d_1 \wedge d_2 \vee \neg v_1 \wedge d_1 \vee \neg v_2 \wedge d_2) = \\ & (v_1 \vee v_2 | d_1 \wedge d_2) \end{aligned}$$

Левая часть равенства (3.101):

$$(v_1 | d_1) + (v_2 | d_2) = (v_1 \vee v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee v_2 \wedge d_2).$$

Булевы функции значения, представленные формулой $v_1 \vee v_2$, идентичны в левой и правой частях. Функцию области определенности $d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee v_2 \wedge d_2$ можно заменить на $d_1 \wedge d_2$ при выполнении условия (3.102), а это и доказывает соотношение (3.101).

3.10. Собственные законы смешанной алгебры

Смешанная алгебра, использующая VCR форму представления частичных переменных и частично определенных функций, имеет также свои *собственные законы*, отсутствующие отдельно в булевой алгебре и в частичной алгебре.

Следующие законы преобразуют взаимосвязанные функции значения и функции области определенности в представлениях частично определенных функций, получаемых путем выполнения частичных операций:

$$(v | d_1) \& (v | d_2) = (v | d_1 \wedge d_2 \vee \neg v \wedge (d_1 \vee d_2)) \quad (3.103)$$

$$(v | d_1) + (v | d_2) = (v | d_1 \wedge d_2 \vee v \wedge (d_1 \vee d_2)) \quad (3.104)$$

Законы (3.103), (3.104) показывают, как преобразуется результирующая область определенности при равенстве функций значения в операндах, над которыми выполняются частичные операции конъюнкции и дизъюнкции.

Следующие законы выполняют простейшие разложения частично определенных функций по частичным операциям конъюнкции и дизъюнкции в более простые частично определенные функции, представленные в VCR форме:

$$(v | v \wedge d) = (v | d) + (v | 0) \quad (3.105)$$

$$(v | \neg v \wedge d) = (v | d) \& (v | 0) \quad (3.106)$$

$$(v | v \vee d) = (v | 1) + (v | d) \quad (3.107)$$

$$(\psi \neg \omega d) = (\psi 1) \& (1 | d) \quad (3.108)$$

$$(\neg \psi \omega d) = (\neg \psi 1) \& (1 | d) \quad (3.109)$$

$$(\neg \psi \neg \omega d) = (\neg \psi 1) + (0 | d) \quad (3.110)$$

В законах (3.105)-(3.110) в исходной частично определенной функции булева функция значения представлена одним литералом: переменной ψ или ее отрицанием $\neg \psi$. Булева функция области определенности представлена с помощью булевых операций отрицания \neg , конъюнкции $\&$ и дизъюнкции \vee . Правые части законов содержат выражения, построенные с использованием частичных операций конъюнкции $\&$ и дизъюнкции $+$, а также с использованием операндов (ψd) , $(\psi 0)$, $(\psi 1)$, $(\neg \psi 1)$, $(0 | d)$, $(1 | d)$. Видим, что замена булевых конъюнкции и дизъюнкции на частичные конъюнкцию и дизъюнкцию происходит нетривиальным образом.

Оказывается, что смешанная алгебра обеспечивает построение других новых законов, описывающих, например, различные разложения частично определенных функций. Смешанная алгебра является также базисом для построения новых форм представления частично определенных функций. Изучим эти законы, разложения и представления в последующих главах.

3.11. Историческая справка

Слияние булевой алгебры с частичной алгеброй и построение смешанной алгебры предложено в работах А.А. Прихожего [53, 54, 56, 58-60, 63, 67, 92-94, 97-99, 105, 106, 108], опубликованных с 1991 по 2012 год. Базовой идеей явилось использование кодирования не полностью определенной булевой функции парой полностью определенных булевых функций таким образом, что стало возможным эффективно представлять частичные одноместные и двухместные операции через булевы одноместные и двухместные операции. Частично определенные булевы функции отличаются от не полностью определенных булевых функций тем, что они представляются формулами смешанной алгебры. Впервые сформулированы законы, выражающие частичные операции через булевы операции, а также законы, позволяющие генерировать различные представления од-

ной и той же частично определенной функции. Сильные и слабые законы частичной алгебры трансформированы в сильные и слабые законы смешанной алгебры. Введенные в перечисленных работах исключающий положительный и исключающий отрицательный ко-факторы ранее не применялись в теории переключательных схем и в разложениях булевых функций.

Глава 4. РАЗЛОЖЕНИЯ ЧАСТИЧНО ОПРЕДЕЛЕННЫХ ФУНКЦИЙ

Под *разложением булевой функции* в булевой алгебре понимается представление этой функции в виде формулы, построенной из булевых операций и более простых булевых функций. Примерами разложения булевой функции являются разложение Шеннона по одной переменной и разложения Давио.

В смешанной алгебре мы рассматриваем *разложения частично определенных функций*. Такие разложения представляются в виде формул смешанной алгебры, построенных из частичных операций и более простых частично определенных функций обычно с меньшей степенью определенности. Как мы уже знаем, уменьшение степени определенности частично определенной функции ведет к более компактному и экономичному представлению булевой функции значения, являющейся компонентом VCR формы частично определенной функции. Далее по тексту, разложения функций в смешанной алгебре будем называть также *частичными разложениями*.

4.1. Частичные разложения, обусловленные частичными операциями

Разрабатываемые в этом параграфе разложения будем называть РЕО от английского Partial Expansion on Operation.

Пусть булева функция f определена через булевы функции g и h посредством операции булевой конъюнкции: $f = g \wedge h$. Все булевы функции являются полностью определенными, следовательно, в VCR форме их можно представить как $(f|1)$, $(g|1)$, $(h|1)$. Тогда функцию $(f|1)$ можно выразить эквивалентным образом через функции $(g|1)$ и $(h|1)$ посредством частичной конъюнкции $\&$: $(f|1) = (g|1) \& (h|1)$.

Теперь мы задаемся вопросом, можно ли увеличить степень неопределенности операндов операции $\&$, получая тот же результат $(f|1)$? Ответ на этот вопрос положительный. Следующие разложения функции $(f|1)$ увеличивают степень неопределенности одного из операндов:

$$(f|1) = (g|h) \& (h|1), \quad (4.1)$$

$$(f|1) = (g|1) \& (h|g). \quad (4.2)$$

В разложении РЕО.4.1 полностью определенный операнд $(g|1)$ заменен частично определенным операндом $(g|h)$. В разложении РЕО.4.2 полностью определенный операнд $(h|1)$ заменен частично определенным операндом $(h|g)$. Докажем равенство РЕО.4.1 с помощью закона (3.12), выполнив следующие преобразования:

$$(g|h)\&(h|1) = (g\wedge h|h\wedge 1\vee\neg g\wedge h\vee\neg h\wedge 1) = (g\wedge h|1) = (f|1).$$

Аналогично доказывается равенство РЕО.4.2. Если булева функция f определена через булевы функции g и h посредством булевых операций отрицания прямой импликации \rightarrow , отрицания обратной импликации \leftarrow , дизъюнкции \vee , стрелки Пирса \downarrow , обратной импликации \leftarrow , прямой импликации \rightarrow и штриха Шеффера $/$, то разложения по соответствующим частичным операциям выглядят следующим образом:

$$(f|1) = \neg((g| \neg h)\Rightarrow(h|1)), \quad \text{если } f = \neg(g \rightarrow h), \quad (4.3)$$

$$(f|1) = \neg((g|h)\leftarrow(h|1)), \quad \text{если } f = \neg(g \leftarrow h), \quad (4.4)$$

$$(f|1) = (g| \neg h) + (h|1), \quad \text{если } f = g \vee h, \quad (4.5)$$

$$(f|1) = (g| \neg h) \Downarrow (h|1), \quad \text{если } f = g \downarrow h, \quad (4.6)$$

$$(f|1) = (g|h)\leftarrow(h|1), \quad \text{если } f = g \leftarrow h, \quad (4.7)$$

$$(f|1) = (g| \neg h)\Rightarrow(h|1), \quad \text{если } f = g \rightarrow h, \quad (4.8)$$

$$(f|1) = (g|h)\oslash(h|1), \quad \text{если } f = g/h. \quad (4.9)$$

В разложениях РЕО.4.3-РЕО.4.9 полностью определенный операнд $(g|1)$ заменен одним из двух частично определенных операндов $(g|h)$ или $(g| \neg h)$. Разложения РЕО.4.3-РЕО.4.9 доказываются аналогично разложению РЕО.4.1.

Следующий вопрос состоит в том, можно ли увеличить степень неопределенности второго операнда? Ответ положительный и на этот вопрос. Следующие разложения функции $(f|1)$ увеличивают степень неопределенности обоих операндов:

$$(f|1) = (g|h)\&(h|g\nu\neg h), \quad \text{если } f = g\wedge h, \quad (4.10)$$

$$(f|1) = \neg((g| \neg h)\Rightarrow(h|g\nu h)), \quad \text{если } f = \neg(g \rightarrow h), \quad (4.11)$$

$$(f|1) = \neg((g|h)\leftarrow(h|\neg g\nu\neg h)), \quad \text{если } f = \neg(g \leftarrow h), \quad (4.12)$$

$$(f|1) = (g| \neg h) + (h|\neg g\nu h), \quad \text{если } f = g \vee h, \quad (4.13)$$

$$(f|1) = (g| \neg h) \Downarrow (h|\neg g\nu h), \quad \text{если } f = g \downarrow h, \quad (4.14)$$

$$(f|1) = (g|h)\leftarrow(h|\neg g\nu\neg h), \quad \text{если } f = g \leftarrow h, \quad (4.15)$$

$$(f1) = (g|\neg h) \Rightarrow (h|g \vee h), \quad \text{если } f = g \rightarrow h, \quad (4.16)$$

$$(f1) = (g|h) \delta^{\circ} (h|g \vee \neg h), \quad \text{если } f = g|h. \quad (4.17)$$

Если степень неопределенности первого операнда увеличена до $(g|h)$ или $(g|\neg h)$, то степень неопределенности второго операнда увеличена до $g \vee h$, $g \vee \neg h$, $\neg g \vee h$ или $\neg g \vee \neg h$. Докажем, например, равенство РЕО.4.12 с помощью закона (3.14), выполнив следующие преобразования:

$$\begin{aligned} \sim((g|h) \Leftarrow (h|\neg g \vee \neg h)) &= \\ \sim(g \Leftarrow h|h \wedge \neg g \vee h \wedge \neg h \vee g \wedge h \vee \neg h \wedge \neg g \vee \neg h \wedge \neg h) &= \\ (\neg(g \Leftarrow h)|h \vee \neg h) = (\neg(g \Leftarrow h)|1) &= (f1). \end{aligned}$$

Корректность остальных разложений РЕО.4.10-РЕО.4.17 доказывается аналогичным образом с использованием законов (3.12)-(3.21).

Для операций исключаящее или \oplus и эквивалентность \leftrightarrow построить аналогичные разложения функции $(f1)$ не удастся. Причина кроется в том, что, согласно законам (3.15), (3.18), результирующая область определенности для этих операций есть пересечение областей определенности операндов. Как следствие, если результирующая область является полной, то области определенности операндов также должны быть полными.

Однако если для представления операций \oplus и \leftrightarrow использовать операции \neg , \wedge , \vee , то становится возможным построение следующих разложений:

$$(f1) = (g|\neg h) \& (\neg h|g) + (\neg g|h) \& (h|\neg g), \quad \text{если } f = g \oplus h, \quad (4.18)$$

$$(f1) = (g|h) \& (h|g) + (\neg g|\neg h) \& (\neg h|\neg g), \quad \text{если } f = g \leftrightarrow h, \quad (4.19)$$

Докажем справедливость разложения РЕО.4.18:

$$\begin{aligned} (g|\neg h) \& (\neg h|g) + (\neg g|h) \& (h|\neg g) &= \\ (g \wedge \neg h | g \wedge \neg h \vee \neg g \wedge \neg h \vee g \wedge h) + (\neg g \wedge h | \neg g \wedge h \vee g \wedge \neg h) &= \\ (g \wedge \neg h | g \vee \neg h) + (\neg g \wedge h | \neg g \vee h) &= \\ (g \wedge \neg h \vee \neg g \wedge h | g \wedge \neg h \vee \neg g \wedge \neg h \vee g \wedge \neg h) &= (g \oplus h | 1) = (f1). \end{aligned}$$

Аналогично доказывается разложение РЕО.4.19.

Таблица 4.1

Пример частичного разложения

| x_1 | x_2 | x_3 | g | h | $f=g\wedge h$ | g' | $g\nu\bar{h}$ | h' |
|-------|-------|-------|-----|-----|---------------|------|---------------|------|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Пример 4.1. Пусть даны две булевы функции g и h , зависящие от трех булевых переменных x_1, x_2, x_3 и определенные табл.4.1. В минимальной дизъюнктивной нормальной форме $g = \neg x_1 \wedge \neg x_2 \vee \neg x_1 \wedge x_3 \vee \neg x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge \neg x_3$ и $h = x_1 \wedge x_2 \vee x_2 \wedge x_3$. Пусть далее булева функция f строится как $f = g \wedge h$. В соответствии с разложением РЕО.4.10, необходимо найти такие функции g' и h' , также представленные в минимальной дизъюнктивной нормальной форме, что $(g'|h) = (g|h)$ и $(h'|g\nu\bar{h}) = (h|g\nu\bar{h})$. Функция g' принимает такое же значение, что и функция g , если $h=1$ (три залитые клетки столбца g' табл.4.1). Если $h=0$, функция g' принимает произвольное значение (пять не залитых клеток). В табл.4.1 функция g' построена таким образом, что ее минимальная дизъюнктивная нормальная форма имеет наиболее компактное представление: $g' = \neg x_1 \vee \neg x_2 \vee \neg x_3$. Функция h' принимает такое же значение, что и функция h , если $g\nu\bar{h}=1$ (семь залитых клеток столбца h' табл.4.1). Если $g\nu\bar{h}=0$, функция h' принимает произвольное значение (только одна не залитая клетка). С целью получения наиболее компактного представления минимальной дизъюнктивной нормальной формы функции h' выбираем 1 в качестве значения не залитой клетки. Функция h остается без изменения: $h' = x_1 \wedge x_2 \vee x_2 \wedge x_3$. Таким образом, размер представления функции g' существенно сократился, функции h' – нет. □

Все рассмотренные выше разложения относятся к определенной на всей области функции $(f|1)$, в которой $f = g \diamond h$, где \diamond – некоторая двухместная булева операция. Пусть теперь исходная функция определена как $(f|d)$ не на всей области, а на области d . Законы

(3.26)-(3.35) доказывают, что для исходной функции имеет место разложение:

$$(f|d) = (g|d) \blacklozenge (h|d), \quad (4.20)$$

где \blacklozenge - частичная двухместная операция, построенная на базе булевой операции \diamond . Следующие разложения позволяют уменьшить степень определенности обоих операндов $(g|d)$ и $(h|d)$ в разложении (4.20):

$$(f|d) = (g|d \wedge h) \& (h|d \wedge (g \vee \neg h)), \quad \text{если } f = g \wedge h, \quad (4.21)$$

$$(f|d) = \neg((g|d \wedge \neg h) \Rightarrow (h|d \wedge (g \vee h))), \quad \text{если } f = \neg(g \rightarrow h), \quad (4.22)$$

$$(f|d) = \neg((g|d \wedge h) \Leftarrow (h|d \wedge (\neg g \vee \neg h))), \quad \text{если } f = \neg(g \leftarrow h), \quad (4.23)$$

$$(f|d) = (g|d \wedge \neg h) + (h|d \wedge (\neg g \vee h)), \quad \text{если } f = g \vee h, \quad (4.24)$$

$$(f|d) = (g|d \wedge \neg h) \wp (h|d \wedge (\neg g \vee h)), \quad \text{если } f = g \downarrow h, \quad (4.25)$$

$$(f|d) = (g|d \wedge h) \Leftarrow (h|d \wedge (\neg g \vee \neg h)), \quad \text{если } f = g \leftarrow h, \quad (4.26)$$

$$(f|d) = (g|d \wedge \neg h) \Rightarrow (h|d \wedge (g \vee h)), \quad \text{если } f = g \rightarrow h, \quad (4.27)$$

$$(f|d) = (g|d \wedge h) \wp (h|d \wedge (g \vee \neg h)), \quad \text{если } f = g|h. \quad (4.28)$$

Очевидно, что разложения ПЕО.4.21-ПЕО.4.28 построены по разложениям ПЕО.4.10-ПЕО.4.17. Разложения ПЕО.4.18, ПЕО.4.19 также легко обобщаются при переходе от полностью определенной функции $(f|1)$ к частично определенной функции $(f|d)$.

Для интерпретации разложений ПЕО.4.10-ПЕО.4.17 воспользуемся диаграммами Вейча. На диаграмме прямоугольником представляется все булево пространство, овалом представляется множество конститuent 1, т.е. область булева пространства, на котором функция принимает значение 1.

Анализ разложений показывает, имеются четыре пары операндов с одинаковыми областями определенности. Эти пары показаны на рис.4.1-4.4. Каждая диаграмма разбита ровно на четыре области: $g \wedge h$, $g \wedge \neg h$, $\neg g \wedge h$, $\neg g \wedge \neg h$. Каждой из областей присваивается частичное значение 0, 1, dc . Каждая диаграмма определяется вектором из четырех значений. Например, левая диаграмма на рис.4.1 определяется вектором $(1, dc, 0, dc)$. Она совпадает с левой диаграммой рис.4.3. Левые диаграммы на рис.4.2 и рис.4.4 определяются вектором $(dc, 1, dc, 0)$. Правые диаграммы отличаются друг от друга и определяются векторами $(1, 0, dc, 0)$, $(1, 0, 1, dc)$, $(dc, 0, 1, 0)$, $(1, dc, 1, 0)$

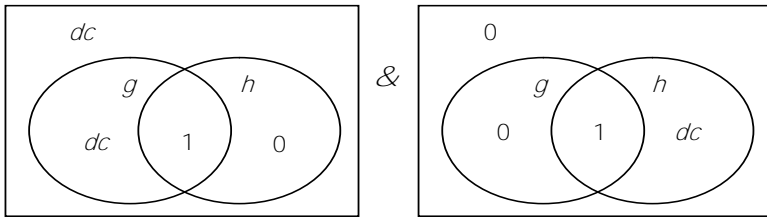


Рисунок 4.1 – Интерпретация операндов разложений (4.10) и (4.17) диаграммами Вейча

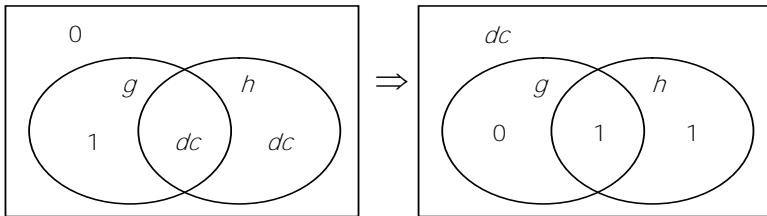


Рисунок 4.2 – Интерпретация операндов разложений (4.11) и (4.16) диаграммами Вейча

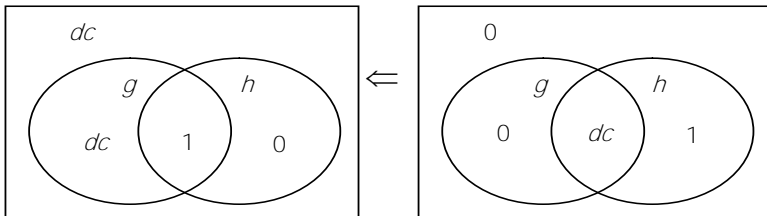


Рисунок 4.3 – Интерпретация операндов разложений (4.12) и (4.15) диаграммами Вейча

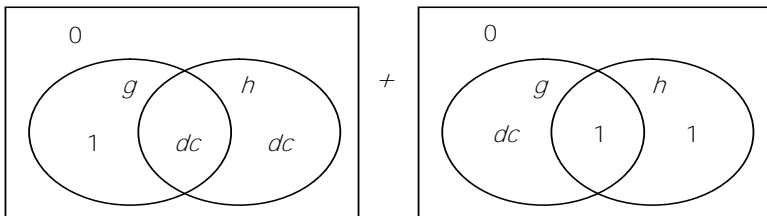


Рисунок 4.4 – Интерпретация операндов разложений (4.13) и (4.14) диаграммами Вейча

соответственно. Количество правых диаграмм определяется числом положений значения dc в четырех элементном векторе.

Интересно отметить, что для левых диаграмм вектор содержит два значения dc , для правых – одно. Как следствие, неопределенность левых операндов выше, чем правых. Более того, легко видеть, что для всех пар диаграмм увеличение области неопределенности правого операнда влечет уменьшение области неопределенности левого операнда, и наоборот. Области неопределенности левых и правых операндов подобны в своем устройстве и функционировании паре сообщающихся сосудов. Как следствие, увеличение возможности минимизации правого операнда уменьшает возможность минимизации левого операнда, и наоборот.

Построенные диаграмм Вейча легко обобщаются для разложений РЕО.4.21-РЕО.4.28. Учет характеристической функции d увеличивает число областей, на которые разбивается булево пространство. Однако, принцип построения диаграмм областей определенности остается прежним.

4.2. Разложения частично определенной функции по характеристической булевой функции

Разрабатываемые в этом параграфе разложения будем называть РЕФ от английского Partial Expansion on Function.

Пусть дана полностью определенная функция $(f|1)$, при этом ничего не известно о структуре булевой функции значения f . Зададимся вопросом, можно ли структурировать функцию $(f|1)$ посредством построения разложений, использующих частичные двухместные операции и операнды с сокращенными областями определенности? Введем для этого в рассмотрение произвольную булеву функцию c , определенную на том же множестве булевых переменных, на каком определена функция f . Никаких специальных требований к функции c не предъявляется.

Построим сначала разложение, в основе которого лежит частичная операция конъюнкция $\&$:

$$(f|1) = (f|f \vee c) \& (f|f \vee \neg c). \quad (4.29)$$

Следующие преобразования с использованием закона (3.12) доказывают справедливость разложения PEF.4.29:

$$\begin{aligned} (\uparrow\downarrow c) \& (\uparrow\downarrow \neg c) &= (f \wedge \uparrow(\downarrow c) \wedge (\downarrow \neg c) \vee \neg f \wedge (\downarrow c) \vee \neg f \wedge (\downarrow \neg c)) = \\ (\uparrow\downarrow \neg c \wedge c \vee \neg f \wedge \neg c) &= (\uparrow\downarrow \neg \uparrow) = (\uparrow 1). \end{aligned}$$

В разложении PEF.4.29 степень определенности операндов уменьшена, в общем случае, по сравнению с исходной функцией. Исключением являются два крайних случая, когда $c=1$ и $c=0$. В этих случаях один операнд становится самой функцией ($\uparrow 1$), другой - константой единица.

Разложения по другим двухместным частичным операциям, включая отрицание прямой импликации $\sim \Rightarrow$, отрицание обратной импликации $\sim \Leftarrow$, дизъюнкцию $+$, стрелку Пирса \Downarrow , обратную импликацию \Leftarrow , прямую импликацию \Rightarrow и штрих Шеффера \Updownarrow , строятся аналогично разложению (4.29) следующим образом:

$$(\uparrow 1) = \sim((\uparrow\downarrow c) \Rightarrow (\neg\uparrow\downarrow \neg c)), \quad (4.30)$$

$$(\uparrow 1) = \sim((\neg\uparrow\downarrow c) \Leftarrow (\uparrow\downarrow \neg c)), \quad (4.31)$$

$$(\uparrow 1) = (\uparrow \neg \downarrow c) + (\uparrow \neg \downarrow \neg c), \quad (4.32)$$

$$(\uparrow 1) = (\neg\uparrow\downarrow c) \Downarrow (\neg\uparrow\downarrow \neg c), \quad (4.33)$$

$$(\uparrow 1) = (\uparrow \neg \downarrow c) \Leftarrow (\neg\uparrow \neg \downarrow \neg c), \quad (4.34)$$

$$(\uparrow 1) = (\neg\uparrow \neg \downarrow c) \Rightarrow (\uparrow \neg \downarrow \neg c), \quad (4.35)$$

$$(\uparrow 1) = (\neg\uparrow \neg \downarrow c) \Updownarrow (\neg\uparrow \neg \downarrow \neg c). \quad (4.36)$$

Разложения PEF.4.30-PEF.4.36 доказываются аналогично разложению PEF.4.29, но с использованием законов (3.13)-(3.21).

Операнды разложений PEF.4.29-PEF.4.36 иллюстрируются диаграммами Вейча, изображенными на рис.4.5-4.12. С принятием той же что и в предыдущем параграфе нумерации областей, пары диаграмм операндов частичных операций определяются следующими парами векторов из четырех элементов:

$$(1, 1, 0, dc) - (1, 1, dc, 0),$$

$$(1, 1, 0, dc) - (0, 0, dc, 1),$$

$$(0, 0, 1, dc) - (1, 1, dc, 0),$$

$$(1, dc, 0, 0) - (dc, 1, 0, 0),$$

$$(0, 0, 1, dc) - (0, 0, dc, 1),$$

$$(1, dc, 0, 0) - (dc, 0, 1, 1),$$

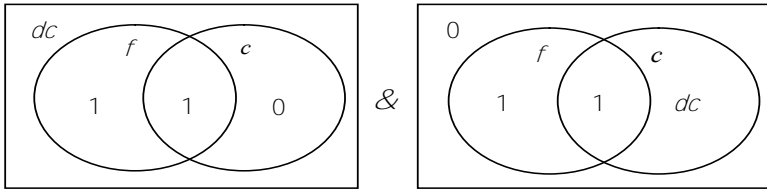


Рисунок 4.5 – Интерпретация операндов разложения (4.29) диаграммами Вейча

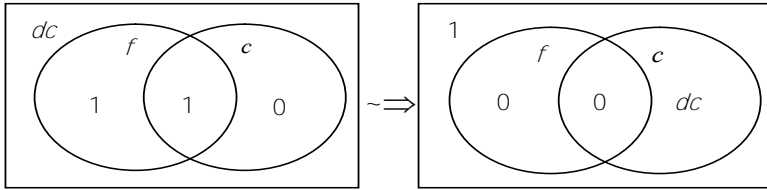


Рисунок 4.6 – Интерпретация операндов разложения (4.30) диаграммами Вейча

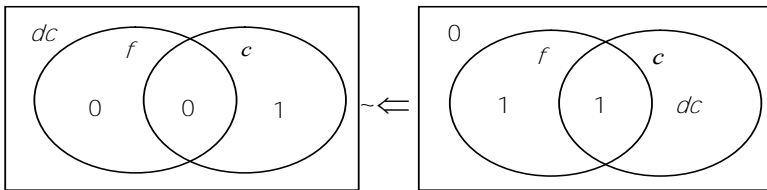


Рисунок 4.7 – Интерпретация операндов разложения (4.31) диаграммами Вейча

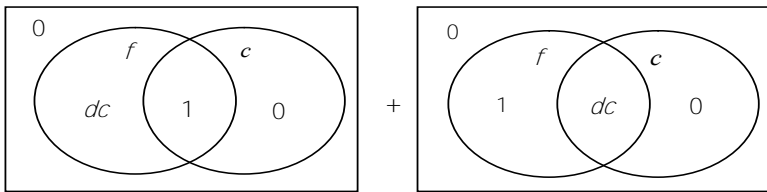


Рисунок 4.8 – Интерпретация операндов разложения (4.32) диаграммами Вейча

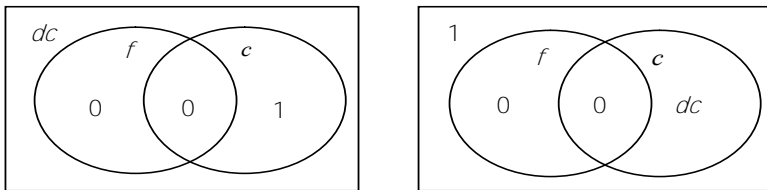


Рисунок 4.9 – Интерпретация операндов разложения (4.33) диаграммами Вейча

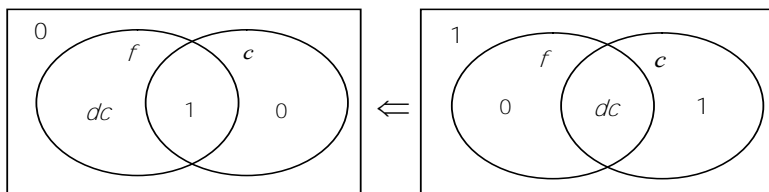


Рисунок 4.10 – Интерпретация операндов разложения (4.34) диаграммами Вейча

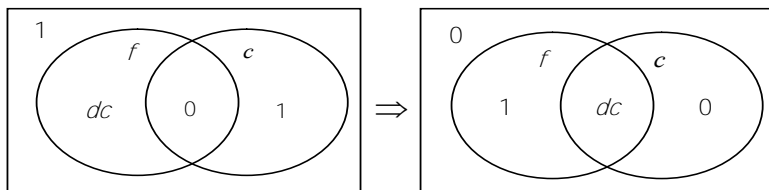


Рисунок 4.11 – Интерпретация операндов разложения (4.35) диаграммами Вейча

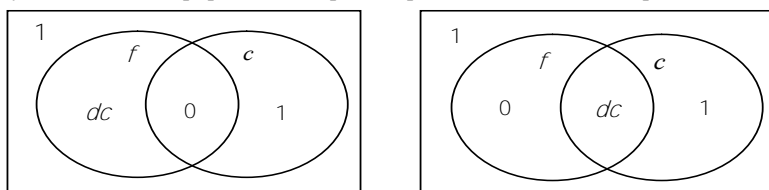


Рисунок 4.12 – Интерпретация операндов разложения (4.36) диаграммами Вейча

$$(0, dc, 1, 1) - (dc, 1, 0, 0),$$

$$(0, dc, 1, 1) - (dc, 0, 1, 1).$$

Легко видеть, что всего существует четыре различных вектора слева, каждый из которых повторяется дважды. Точно также, существует четыре различных вектора справа, каждый из которых также повторяется дважды. Однако не существует ни одной повторяющейся пары векторов. Все восемь пар уникальны. В этом смысле уникальным является каждое разложение по одной из восьми двухместных частичных операций.

Возникает закономерный вопрос, если одну функцию можно разложить восемью альтернативными способами, то какой из способов предпочтительнее в каждой конкретной ситуации? Прежде

чем попытаться ответить на этот вопрос, рассмотрим особенности операндов различных операций.

Прежде всего, следует отметить, что исходная функция f является заданной и фиксированной, в то время как функция c не фиксирована и выбирается нужным нам образом. Другими существенными свойствами операндов при анализе предпочтительного применения того или иного разложения являются:

- строится ли формульное представление операнда по множеству конституент 0 или по множеству конституент 1 функции значения $\vee CR$ формы операнда;
- какая область f или $\neg f$ используется в разложении для непосредственного построения функции значения операнда;
- каково соотношение размеров областей f и $\neg f$;
- пересекается ли или не пересекается область неопределенности операнда с областью f .

Приведем несколько примеров. Так дизъюнктивная нормальная форма булевой функции строится по множеству конституент 1, в то же время конъюнктивная нормальная форма строится по множеству конституент 0. Четыре левых операнда восьми разложений и четыре правых операнда используют область f для построения функции значения. Ровно столько же левых и правых операндов используют область $\neg f$. В четырех левых операндах и четырех правых операндах область неопределенности пересекается с областью f . В таком же количестве левых и правых операндов область неопределенности не пересекается с областью f .

С точки зрения построения операндов с максимальной степенью неопределенности полезными оказываются следующие рассуждения. Пусть дана исходная функция, для которой область f меньше по размеру (возможно существенно) области $\neg f$. Пусть далее формульное представление функции значения строится по множеству конституент 1. Какие из разложений могут дать операнды с максимальной степенью неопределенности? Очевидно это разложения PEF.4.29-PEF.4.31 и PEF.4.33, так как область неопределенности обоих операндов строится как часть большой области $\neg f$. В разложениях PEF.4.32, PEF.4.34-PEF.4.36 области неопределенности операндов строятся как часть малой области f , поэтому они не могут быть большими. Среди первых четырех разложений, разложе-

ние PEF.4.29 предпочтительнее тем, что дает малую область конститuent 1 и большую область неопределенности обоих операндов. Разложения PEF.4.30, PEF.4.31 дают большую область конститuent 1 одного из операндов. Разложение PEF.4.33 дает большую область конститuent 1 обоих операндов.

Особенность разложения PEF.4.29 состоит в том, что область C может не пересекаться с областью f , от этого результат разложения не зависит. Результат зависит только от того, насколько удачно и сбалансированно область $\neg f$ разбивается областью C на две части, на которых функция f может принять дополнительно в каждом из операндов значение 1.

Сравним восемь разложений PEO.4.10-PEO.4.17 по булевым операциям с восемью разложениями PEF.4.29-PEF.4.36 по произвольной булевой функции. В разложениях PEF.4.29-PEF.4.36 область неопределенности всех операндов представлена одной подобластью из четырех. В разложениях PEO.4.10-PEO.4.17 область неопределенности первого операнда представлена двумя подобластями из четырех, второго операнда - одной подобластью. Как видим сила разложений PEO.4.10-PEO.4.17 выше силы разложений PEF.4.29- PEF.4.36. Это обусловлено тем, что в первых разложениях заранее известна структура исходной функции, во вторых разложениях структура функции заранее не известна.

Пример 4.2. Пусть задана функция $(f1)$, где булева функция значения f зависит от трех булевых переменных x_1, x_2, x_3 и определяется табл.4.2. В дизъюнктивной нормальной форме $f = \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \vee \neg x_1 \wedge x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge \neg x_3$. Область конститuent 1 функции f меньше области конститuent 0, поэтому воспользуемся разложением PEF.4.29 по частичной конъюнкции $\&$. Начинаем с выбора булевой функции C . Множество конститuent 0 функции f разбиваем с помощью C на две части как показано в табл.4.2. На двух конститuentaх 0 функция $C=1$, на остальных наборах $C=0$.

Далее строим функции $f \vee C$ и $f \vee \neg C$, описывающие области определенности операндов. Функцию $(f1)$ представляем согласно разложению PEF.4.29 в виде $(f' | f \vee C) \& (f'' | f \vee \neg C)$, где f' и f'' - минимизированные в классе дизъюнктивных нормальных форм булевы функции значения по сокращенным областям определенности. В табл.4.2

Таблица 4.2

Пример частичного разложения

| x_1 | x_2 | x_3 | f | c | $\bar{f}c$ | f' | $\bar{f}\bar{c}$ | f'' |
|-------|-------|-------|-----|-----|------------|------|------------------|-------|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

темной заливкой показано, что функция f' принимает на пяти наборах значений аргументов значение, совпадающее со значением функции f . На трех оставшихся наборах значение может быть произвольным, выбираем значение 1. Построение минимальной дизъюнктивной нормальной формы дает $f' = x_1 \vee x_2 \wedge x_3 \vee \bar{x}_2 \wedge \bar{x}_3$. Темной заливкой показано также, что $f'' = f$ на шести наборах значений аргументов. На двух оставшихся наборах значение произвольно, выбираем значение 1. Построение минимальной дизъюнктивной нормальной формы дает $f'' = \bar{x}_1 \vee x_2 \wedge \bar{x}_3$. В обоих операндах получаем существенное сокращение сложности функции значения. Исходя из построенного частичного разложения, функцию f можно выразить через функции f' и f'' как $f = (x_1 \vee x_2 \wedge x_3 \vee \bar{x}_2 \wedge \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \wedge \bar{x}_3)$. \square

Перейдем от определенной на всей области функции ($f \uparrow 1$) к частично определенной функции ($f \downarrow d$), в которой область определенности описывается характеристической функцией d . Разложения PEF.4.29-PEF.4.36 обобщаются для функции ($f \downarrow d$) следующим образом:

$$(f \downarrow d) = (f \downarrow d \wedge (\bar{f}c)) \& (f \downarrow d \wedge (\bar{f}\bar{c})). \quad (4.37)$$

$$(f \downarrow d) = \sim((f \downarrow d \wedge (\bar{f}c)) \Rightarrow (\bar{f} \downarrow d \wedge (\bar{f}\bar{c}))), \quad (4.38)$$

$$(f \downarrow d) = \sim(\bar{f} \downarrow d \wedge (\bar{f}c)) \Leftarrow (f \downarrow d \wedge (\bar{f}\bar{c})), \quad (4.39)$$

$$(f \downarrow d) = (f \downarrow d \wedge (\bar{f}\bar{c})) + (f \downarrow d \wedge (\bar{f}c)), \quad (4.40)$$

$$(f \downarrow d) = (\bar{f} \downarrow d \wedge (\bar{f}c)) \wp (\bar{f} \downarrow d \wedge (\bar{f}\bar{c})), \quad (4.41)$$

$$(f \downarrow d) = (f \downarrow d \wedge (\bar{f}\bar{c})) \Leftarrow (\bar{f} \downarrow d \wedge (\bar{f}c)), \quad (4.42)$$

$$(f \downarrow d) = (\bar{f} \downarrow d \wedge (\bar{f}c)) \Rightarrow (f \downarrow d \wedge (\bar{f}\bar{c})), \quad (4.43)$$

$$(f \downarrow d) = (\bar{f} \downarrow d \wedge (\bar{f}\bar{c})) \wp (\bar{f} \downarrow d \wedge (\bar{f}c)). \quad (4.44)$$

Благодаря использованию операции булевой конъюнкции, степень неопределенности операндов всех восьми разложений меньше чем степень неопределенности исходной функции (f_d).

4.3. Частичные разложения по двум частичным операциям

Разрабатываемые в этом параграфе разложения будем называть PE2F от английского Partial Expansion on Two Operations and Function.

Все рассмотренные выше частичные разложения построены с использованием одной двухместной частичной операции. В настоящем разделе рассмотрим, как строятся частичные разложения с использованием нескольких двухместных частичных операций. Прежде всего, интерес представляют пары частичных операций, такие как конъюнкция и дизъюнкция, двойное применение операции штрих Шеффера, двойное применение операции стрелка Пирса.

Пусть, как и прежде, (f_1) – полностью определенная функция, в которой ничего не известно о структуре булевой функции f . Пусть далее c – произвольная булева функция, определенная на том же множестве булевых переменных, что и f .

Выберем для начала пару частичных операций дизъюнкция и конъюнкция. Применим к функции (f_1) следующее очевидное разложение по частичной дизъюнкции +

$$(f_1) = (f \wedge d_1) + (f \wedge \neg d_1), \quad (4.45)$$

получив в результате пару операндов, представленных функциями, подлежащими дальнейшему разложению по частичной конъюнкции &. Для первой функции строим разложение

$$(f \wedge d_1) = (d_1) \& (f c). \quad (4.46)$$

Докажем его:

$$(d_1) \& (f c) = (c \wedge f_1 \wedge c \vee \neg c \wedge 1 \vee \neg f \wedge c) = (c \wedge f_1).$$

Для второй функции строим разложение

$$(f \wedge \neg d_1) = \neg(d_1) \& (f \neg c). \quad (4.47)$$

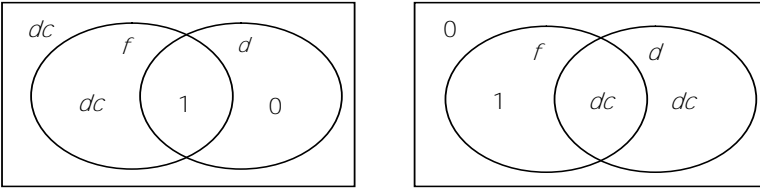


Рисунок 4.13 – Диаграммы Вейча операндов разложения (4.48)

Докажем его:

$$\sim(d1) \& (f \neg c) = (\neg c \wedge f1 \wedge \neg c \vee c \wedge 1 \vee \neg f \wedge \neg c) = (\neg c \wedge f1).$$

Собирая воедино разложения (4.45)-(4.47), получаем следующее важное разложение:

$$(f1) = (c1) \& (fc) + \sim(d1) \& (f \neg c). \quad (4.48)$$

Построим для его двух основных операндов диаграммы Вейча, показанные на рис.4.13. Диаграммы определяются парой векторов

$$(1, dc, 0, dc) - (dc, 1, dc, 0).$$

Сравнивая диаграммы полученного разложения с диаграммами всех ранее построенных разложений, можно сделать следующие выводы:

- степень неопределенности каждого из основных операндов полученного разложения описывается двумя подобластями диаграммы из четырех – это больше чем во всех ранее рассмотренных разложениях;
- стратегия построения функции c должна искать сбалансированное разбиение как множества конституент 0 так и множества конституент 1 функции f ;
- очевидное преимущество в увеличении степени неопределенности основных операндов может компенсироваться недостатком вхождения в значимую часть разложения функции c (для смягчения недостатка необходимо стремиться выбирать простую в реализации функцию c).

Рассмотрим частные случаи разложения PE2F.4.48. Простейшим полезным видом функции c является x_i . Подстановка x_i вместо c в разложении PE2F.4.48 дает:

$$(f1) = (x_i|1) \& (f|x_i) + \sim(x_i|1) \& (f|\neg x_i). \quad (4.49)$$

Используя законы смешанной алгебры, изложенные в разделе 3.7, из разложения РЕ2F.4.49 получаем:

$$(f1) = (x_i|1) \& (f_{x_i=1}|x_i) + \sim(x_i|1) \& (f_{x_i=0}|\neg x_i). \quad (4.50)$$

где $f_{x_i=1}$ и $f_{x_i=0}$ есть остаточные функции, называемые положительным и отрицательным кофактором функции f по переменной x_i . Легко видеть, что разложение РЕ2F.4.50 есть по существу разложение Шеннона булевой функции по одной переменной.

Другим важнейшим частным случаем является $C = x_i \oplus x_j$. Разложение РЕ2F.4.48 приобретает вид

$$(f1) = (x_i \oplus x_j|1) \& (f|x_i \oplus x_j) + \sim(x_i \oplus x_j|1) \& (f|\neg(x_i \oplus x_j)). \quad (4.51)$$

Снова используя законы смешанной алгебры, сформулированные в разделе 3.7, преобразуем разложение РЕ2F.4.51 к разложению

$$(f1) = (x_i \oplus x_j|1) \& (f_{x_i=\neg x_j}|x_i \oplus x_j) + \sim(x_i \oplus x_j|1) \& (f_{x_i=x_j}|\neg(x_i \oplus x_j)), \quad (4.52)$$

где $f_{x_i=\neg x_j}$ и $f_{x_i=x_j}$ есть остаточные функции, называемые исключающим отрицательным и исключающим положительным кофактором функции f по переменным x_i и x_j .

Замечательным свойством функций $C = x_i$ и $C = x_i \oplus x_j$ является то, что они, будучи простыми по своей структуре, разбивают булево пространство на две равные части, что, как правило, дает сбалансированность областей определенности операндов разложений. Кроме того, получаемые в результате остаточные функции зависят существенно от меньшего числа переменных. Какое из разложений РЕ2F.4.50 или РЕ2F.4.52 предпочтительнее? Это зависит от вида каждой конкретной функции f .

Если в качестве C выбрать функции $C = x_i \wedge x_j$, $C = x_i \vee x_j$ или $C = x_i \rightarrow x_j$, то в этом случае получаем асимметричное разбиение булевого пространства. Это, очевидно, приводит к тому, что один операнд разложения будет иметь высокую степень неопределенности, второй операнд – низкую степень неопределенности, хотя разложение в целом может оказаться эффективным. В качестве C может быть также выбрана произвольная булева функция, специально построенная для конкретной функции f .

Пример 4.3. Выполним разложение по соотношениям PE2F.4.50 и PE2F.4.52 функции $(f1)$ из примера 4.2, где f определяется табл.4.2 и описывается формулой $f = \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \vee \neg x_1 \wedge x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge \neg x_3$. Применение разложения PE2F.4.50 по переменной x_1 и кофакторам $f_{x_1=1}$ и $f_{x_1=0}$ дает

$$(f1) = (x_1|1) \& (x_2 \wedge \neg x_3 | x_1) + \sim (x_1|1) \& (\neg x_2 \wedge \neg x_3 \vee x_2 \wedge x_3 | \neg x_1).$$

Применение разложения PE2F.4.52 по переменным x_2 и x_3 и кофакторам $f_{x_3=\neg x_2}$ и $f_{x_3=x_2}$ дает

$$(f1) = (x_2 \oplus x_3 | 1) \& (x_1 \wedge x_2 | x_2 \oplus x_3) + \sim (x_2 \oplus x_3 | 1) \& (\neg x_1 | \neg (x_2 \oplus x_3)).$$

Сравнивая два полученных разложения, делаем вывод о том, что кофакторы второго разложения проще кофакторов первого разложения, хотя функция, по которой выполняется разложение, проще в первом случае. \square

Продолжим рассмотрение частичных операций дизъюнкция и конъюнкция для построения разложений функции $(f1)$. При этом изменим порядок применения операций: сначала воспользуемся операцией конъюнкция, затем воспользуемся операцией дизъюнкция:

$$(f1) = ((d1) + (f \neg c)) \& (\sim (d1) + (f c)). \quad (4.53)$$

Доказательство разложения PE2F.4.53:

$$\begin{aligned} & ((d1) + (f \neg c)) \& (\sim (d1) + (f c)) = \\ & (c \vee f1 \wedge \neg c \vee c \wedge 1 \vee f \wedge \neg c) \& (\neg c \vee f1 \wedge c \vee \neg c \wedge 1 \vee f \wedge c) = \\ & (c \vee f1) \& (\neg c \vee f1) = (f1). \end{aligned}$$

Рассмотрим теперь частичное разложение, построенное на двойном применении (на двух уровнях иерархии) частичной операции штрих Шеффера к функции $(f1)$:

$$(f1) = (d1) \delta (f c) \delta \sim (d1) \delta (f \neg c). \quad (4.54)$$

Доказательство разложения PE2F.4.54:

$$\begin{aligned} & ((d1) \delta (f c)) \delta (\sim (d1) \delta (f \neg c)) = \\ & (d f1 \wedge c \vee \neg c \wedge 1 \vee \neg f \wedge c) \delta (\neg d f1 \wedge \neg c \vee c \wedge 1 \vee \neg f \wedge \neg c) = \\ & (d f1) \delta (\neg d f1) = ((d f) / (\neg d f) | 1) = (f1). \end{aligned}$$

Построим частичное разложение, основанное на двойном применении частичной операции стрелка Пирса к функции $(f1)$:

$$(f1) = (d1) \wp (f \neg c) \wp \sim (d1) \wp (fc). \quad (4.55)$$

Доказательство разложения PE2F.4.55:

$$\begin{aligned} & ((d1) \wp (f \neg c) \wp \sim (d1) \wp (fc)) = \\ & (c \downarrow f1 \wedge \neg c \vee c \wedge 1 \vee f \wedge \neg c) \wp (\neg c \downarrow f1 \wedge c \vee \neg c \wedge 1 \vee f \wedge c) = \\ & (c \downarrow f1) \wp (\neg c \downarrow f1) = ((c \downarrow f) \downarrow (\neg c \downarrow f) | 1) = (f1). \end{aligned}$$

Все четыре разложения PE2F.4.48, PE2F.4.53-PE2F.4.55 построены на использовании одних и тех же операндов с одной и той же степенью неопределенности, поэтому имеют одинаковую выразительную силу. Результаты анализа разложения PE2F.4.48, проведенного выше, в полной мере относятся к разложениям PE2F.4.53-PE2F.4.55.

Пусть теперь исходная функция является не полностью, а частично определенной функцией с областью определенности d . Разложения PE2F.4.48, PE2F.4.53-PE2F.4.55 обобщаются для функции (fd) следующим образом:

$$(fd) = (d1) \& (fd \wedge c) + \sim (d1) \& (fd \wedge \neg c), \quad (4.56)$$

$$(fd) = ((d1) + (fd \wedge \neg c)) \& (\sim (d1) + (fd \wedge c)), \quad (4.57)$$

$$(fd) = (d1) \wp (fd \wedge c) \wp \sim (d1) \wp (fd \wedge \neg c), \quad (4.58)$$

$$(fd) = (d1) \wp (fd \wedge \neg c) \wp \sim (d1) \wp (fd \wedge c). \quad (4.59)$$

Основные частичные операнды, образованные от исходной функции, имеют области определенности, описанные двумя характеристическими функциями $d \wedge c$ и $d \wedge \neg c$. Благодаря использованию булевой конъюнкции \wedge размер областей сокращен по сравнению с размером исходной области d . В разложениях PE2F.4.56-PE2F.4.59 операнд $(d1)$ является полностью определенным и унифицированным во всех вхождениях. В следующих разложениях этот операнд становится частично определенным по области определенности d :

$$(fd) = (cd) \& (fd \wedge c) + \sim (cd) \& (fd \wedge \neg c), \quad (4.60)$$

$$(fd) = ((cd) + (fd \wedge \neg c)) \& (\sim (cd) + (fd \wedge c)), \quad (4.61)$$

$$(fd) = (cd) \wp (fd \wedge c) \wp \sim (cd) \wp (fd \wedge \neg c), \quad (4.62)$$

$$(fd) = (cd) \wp (fd \wedge \neg c) \wp \sim (cd) \wp (fd \wedge c). \quad (4.63)$$

Снижение степени определенности операнда c может повысить эффективность представления продуктов разложений. Особенно это проявляется при построении иерархических разложений частично определенной функции.

4.4. Частичные разложения по операциям исключающее или и эквивалентность

Разрабатываемые в этом параграфе разложения будем называть PE2F, как и в предыдущем параграфе.

Частичные операции исключающее или и эквивалентность имеют свойства, отличающиеся от свойств других восьми базовых двухместных частичных операций. Отличия проявляются в правилах вычисления результирующей области определенности и в невозможности построения разложений, аналогичных разложениям для восьми других базовых двухместных операций.

Наиболее известными и полезными разложениями булевой функции, в основе которых лежит булева операция исключающее или, являются положительное и отрицательное разложения Давио. Закономерен вопрос, можно ли построить в смешанной алгебре разложения, аналогичные разложениям Давио, но базирующиеся на частичной операции исключающее или? Пусть, как и прежде, $(f|1)$ – полностью определенная функция и пусть c – произвольная булева функция, определенная на том же множестве булевых переменных, что и f .

Используя компоненты обобщенного разложения Шеннона PE2F.4.48 и заменяя булевы операции на соответствующие частичные операции в положительном и отрицательном разложениях Давио, записываем:

$$(f|1) = (f \neg c) \odot (c|1) \& ((f|c) \odot (f \neg c)). \quad (4.64)$$

$$(f|1) = (f|c) \odot \sim(c|1) \& ((f|c) \odot (f \neg c)). \quad (4.65)$$

Выполняются ли равенства (4.64) и (4.65) при любых функциях f и c ? Преобразуем правую часть соотношения (4.64) с использованием законов (3.12)-(3.21):

$$\begin{aligned}
& (\neg c) \dot{\vee} (d \vee 1) \& ((f \vee c) \dot{\vee} (\neg c)) = \\
& (\neg c) \dot{\vee} (d \vee 1) \& (f \oplus f \vee c \wedge \neg c) = (\neg c) \dot{\vee} (d \vee 1) \& (0 \vee 0) = \\
& (\neg c) \dot{\vee} (0 \vee \neg c) = (f \oplus 0 \vee \neg c) = (\neg c).
\end{aligned}$$

Для того, чтобы $(\neg c)$ равнялась $(f \vee 1)$, необходимо чтобы выполнялось $c=0$. Аналогично для соотношения (4.65) можно получить необходимость выполнения $c=1$. Как видим, разложения (4.64), (4.65) имеют место для двух очень частных случаев функции c . Однако ситуация в отношении обобщения разложений Давио намного более оптимистична. В последующих главах будет введена операция минимизации для VCR формы частично определенной функции и будет показано, что положительный и отрицательный кофакторы в разложениях Давио могут быть заменены операцией минимизации функции значения по характеристической функции области определенности.

Возможны также другие способы построения разложений функции $(f \vee 1)$ по частичным операциям исключаящее или, конъюнкция и инверсия, снижающие степень определенности операндов. Пусть c – произвольная булева функция. Очевидно, что выполняются следующие разложения:

$$\begin{aligned}
(f \vee 1) &= (f \wedge d \vee 1) \dot{\vee} (f \wedge \neg d \vee 1), \\
(f \wedge d \vee 1) &= (\neg f \vee \neg d) \& (d \vee \neg c), \\
(f \wedge \neg d \vee 1) &= (\neg f \vee \neg \neg c) \& (\neg d \vee c).
\end{aligned}$$

Собирая их воедино, получаем:

$$(f \vee 1) = ((d \vee \neg c) \& (\neg f \vee \neg c)) \dot{\vee} (\neg (d \vee c) \& (\neg f \vee \neg c)). \quad (4.66)$$

По частичным операциям эквивалентность, дизъюнкция и инверсия строим аналогичное разложение:

$$(f \vee 1) = ((d \vee \neg c) + (f \vee \neg c)) \Leftrightarrow (\neg (d \vee \neg c) + (f \vee c)). \quad (4.67)$$

Сравнивая разложения PE2F.4.66, PE2F.4.67 с разложениями PE2F.4.48, PE2F.4.53, приходим к выводу о том, что разложения PE2F.4.48, PE2F.4.53 предпочтительнее, поскольку степень неопределенности операндов в них выше.

4.5. Иерархические частичные разложения

Разложения PEF.4.37-PEF.4.44 а также PE2F.4.56-PE2F.4.63 можно использовать иерархически. На уровне иерархии i каждое из разложений можно применить к каждому новому менее определенному операнду, выбрав новую подходящую функцию c_i . С каждым шагом иерархии или рекурсии операнды становятся все менее определенными, пока функции значения не редуцируются до элементарных многоместных конъюнкций и дизъюнкций, констант 0 и 1 или литералов x_i или $\neg x_i$.

Рассмотрим сначала построение иерархических разложений по одной частичной операции. Выберем частичную конъюнкцию. Пусть на шаге i имеется частично определенная функция $(f_i|d_i)$, и к ней применяется разложение PEF.4.37 по булевой функции c_i :

$$(f_i|d_i) = (f_i|d_i \wedge (f_i \vee c_i)) \& (f_i|d_i \wedge (f_i \vee \neg c_i)). \quad (4.68)$$

Каждый из двух операндов в правой части (4.68) имеет меньшую степень определенности, чем исходная функция. Снова применим разложение PEF.4.37 с новой функцией c_{i+1} , но уже не к функции $(f_i|d_i)$, а к операндам разложения (4.68). Левый операнд:

$$\begin{aligned} (f_i|d_i \wedge (f_i \vee c_i)) = \\ (f_i|d_i \wedge (f_i \vee c_i) \wedge (f_i \vee c_{i+1})) \& (f_i|d_i \wedge (f_i \vee c_i) \wedge (f_i \vee \neg c_{i+1})) = \\ (f_i|d_i \wedge (f_i \vee c_i \wedge c_{i+1})) \& (f_i|d_i \wedge (f_i \vee c_i \wedge \neg c_{i+1})). \end{aligned}$$

Правый операнд:

$$\begin{aligned} (f_i|d_i \wedge (f_i \vee \neg c_i)) = \\ (f_i|d_i \wedge (f_i \vee \neg c_i) \wedge (f_i \vee c_{i+1})) \& (f_i|d_i \wedge (f_i \vee \neg c_i) \wedge (f_i \vee \neg c_{i+1})) = \\ (f_i|d_i \wedge (f_i \vee \neg c_i \wedge c_{i+1})) \& (f_i|d_i \wedge (f_i \vee \neg c_i \wedge \neg c_{i+1})). \end{aligned}$$

Множественное применение разложения PEF.4.37 по частичной конъюнкции дает следующую структуру операнда на иерархическом уровне k :

$$(f_i|d_i \wedge (f_i \vee l_i \wedge \dots \wedge l_k)). \quad (4.69)$$

где l_i - литерал вида c_i или $\neg c_i$. Легко видеть, что при увеличении значения k характеристическая функция области определенности операнда (4.69) при правильном выборе функций $c_i \dots c_k$ стремится

к $d_i \wedge f_i$. Операнд стремится к $(f_i | d_i \wedge f_i) = (1 | d_i \wedge f_i)$. Однако это предельное значение не может быть достигнуто для всех операндов. Достижимым предельным значением является $(D | d_i \wedge (f_i \vee /i \wedge \dots \wedge /k))$, где D – элементарная многоместная дизъюнкция переменных.

Построим теперь иерархическое разложение на базе частичной стрелки Пирса. Используя (4.41) на уровне i иерархии, записываем:

$$(f_i | d_i) = (\neg f_i | d_i \wedge (f_i \vee c_i)) \wp (\neg f_i | d_i \wedge (f_i \vee \neg c_i)). \quad (4.70)$$

Применение разложения PEF.4.41 к левому операнду разложения (4.70) с функцией c_{i+1} дает:

$$\begin{aligned} & (\neg f_i | d_i \wedge (f_i \vee c_i)) = \\ & (\neg \neg f_i | d_i \wedge (f_i \vee c_i) \wedge (\neg f_i \vee c_{i+1})) \wp (\neg \neg f_i | d_i \wedge (f_i \vee c_i) \wedge (\neg f_i \vee \neg c_{i+1})) = \\ & (f_i | d_i \wedge (f_i \wedge c_{i+1} \vee \neg f_i \wedge \neg c_i \wedge c_{i+1})) \wp (f_i | d_i \wedge (f_i \wedge \neg c_{i+1} \vee \neg f_i \wedge c_i \wedge \neg c_{i+1})) = \end{aligned}$$

Правый операнд:

$$\begin{aligned} & (\neg f_i | d_i \wedge (f_i \vee \neg c_i)) = \\ & (\neg \neg f_i | d_i \wedge (f_i \vee \neg c_i) \wedge (\neg f_i \vee c_{i+1})) \wp (\neg \neg f_i | d_i \wedge (f_i \vee \neg c_i) \wedge (\neg f_i \vee \neg c_{i+1})) = \\ & (f_i | d_i \wedge (f_i \wedge c_{i+1} \vee \neg f_i \wedge \neg c_i \wedge c_{i+1})) \wp \\ & (f_i | d_i \wedge (f_i \wedge \neg c_{i+1} \vee \neg f_i \wedge c_i \wedge \neg c_{i+1})). \end{aligned}$$

Анализ характеристических функций областей определенности операндов свидетельствует о том, что в каждую элементарную конъюнкцию характеристической функции входит, по крайней мере, одна из функций c_i, c_{i+1}, \dots и при правильном выборе функций c_i, c_{i+1}, \dots можно стремиться получить на подходящем уровне иерархии пустую область определенности каждого из операндов. На самом деле, предельная область определенности может быть такой, что функция f_i может быть заменена на константы 0 и 1 или литералы X_i и $\neg X_i$.

Рассмотрим очень важный частный случай, когда на соседних уровнях иерархии $c_i = c_{i+1}$. Тогда собирая воедино все части разложения (4.70) и заменяя c_{i+1} на c_i , получаем:

$$\begin{aligned} (f_i | d_i) = & ((f_i | d_i \wedge c_i) \wp (f_i | d_i \wedge (f_i \oplus c_i))) \wp \\ & ((f_i | d_i \wedge (f_i \leftrightarrow c_i)) \wp (f_i | d_i \wedge \neg c_i)). \end{aligned}$$

Применение законов (3.57), (3.58) к двум операндам полученного разложения и замена функции f_i на функции c_i и $\neg c_i$ соответственно дает разложение

$$(f_i|d_i) = ((\neg c_i|d_i \wedge (f_i \oplus c_i)) \wp (f_i|d_i \wedge c_i) \wp) \\ ((c_i|d_i \wedge (f_i \leftrightarrow c_i)) \wp (f_i|d_i \wedge \neg c_i)). \quad (4.71)$$

Не сложно распознать в разложении (4.71) аналог разложения РЕ2F.4.63. Иерархическое разложение на базе частичной операции штрих Шеффера строится аналогично иерархическому разложению на базе частичной операции стрелка Пирса.

Рассмотрим построение иерархических разложений в случае применения чередующихся пар частичных операций дизъюнкции и конъюнкции на каждом уровне иерархии i . Разложение по дизъюнкции и конъюнкции будем выполнять с использованием одной характеристической функции c_i области определенности.

Пусть на шаге i некоторый операнд представлен частично определенной функцией $(f_i|d_i)$. Применим разложение РЕF.4.40:

$$(f_i|d_i) = (f_i|d_i \wedge (\neg f_i \vee c_i)) + (f_i|d_i \wedge (\neg f_i \vee \neg c_i)). \quad (4.72)$$

К первому операнду операции $+$ применим разложение РЕF.4.37:

$$(f_i|d_i \wedge (\neg f_i \vee c_i)) = \\ (f_i|(d_i \wedge (\neg f_i \vee c_i)) \wedge (f_i \vee c_i)) \& (f_i|(d_i \wedge (\neg f_i \vee c_i)) \wedge (f_i \vee \neg c_i)) = \\ (f_i|d_i \wedge c_i) \& (f_i|d_i \wedge ((f_i \wedge c_i) \vee (\neg f_i \wedge \neg c_i))) = \\ (f_i|d_i \wedge c_i) \& (f_i|d_i \wedge (f_i \leftrightarrow c_i)) = \\ (c_i|d_i \wedge (f_i \leftrightarrow c_i)) \& (f_i|d_i \wedge c_i). \quad (4.73)$$

Ко второму операнду операции $+$ также применим разложение РЕF.4.37:

$$(f_i|d_i \wedge (\neg f_i \vee \neg c_i)) = \\ (f_i|(d_i \wedge (\neg f_i \vee \neg c_i)) \wedge (f_i \vee c_i)) \& (f_i|(d_i \wedge (\neg f_i \vee \neg c_i)) \wedge (f_i \vee \neg c_i)) = \\ (f_i|d_i \wedge ((\neg f_i \wedge c_i) \vee (f_i \wedge \neg c_i))) \& (f_i|d_i \wedge \neg c_i) = \\ (f_i|d_i \wedge (f_i \oplus c_i)) \& (f_i|d_i \wedge \neg c_i) = \\ (\neg c_i|d_i \wedge (f_i \oplus c_i)) \& (f_i|d_i \wedge \neg c_i) = \\ \sim (c_i|d_i \wedge (f_i \oplus c_i)) \& (f_i|d_i \wedge \neg c_i). \quad (4.74)$$

Объединяя компоненты (4.72)-(4.74), получаем разложение:

$$(f_i|d_i) = (c_i|d_i \wedge (f_i \leftrightarrow c_i)) \& (f_i|d_i \wedge c_i) + \\ \sim (c_i|d_i \wedge (f_i \oplus c_i)) \& (f_i|d_i \wedge \neg c_i). \quad (4.75)$$

Детальный анализ показывает, что разложение (4.75) есть по существу разложение PE2F.4.60 функции $(f_i|d_i)$ по функции c_i .

Рассмотренные способы построения иерархических разложений не исчерпывают всего многообразия в порядке применения частичных операций на каждом уровне иерархии. В предельном случае задача выбора частичной операции и соответствующего разложения может рассматриваться самостоятельным образом на каждом уровне иерархии для каждого частичного операнда, полученного на предыдущем уровне иерархии.

Рассмотрим теперь построение иерархических разложений на базе разложений PE2F.4.56-PE2F.4.63. Пусть, как и прежде, на шаге i получена частично определенная функция $(f_i|d_i)$. Выберем разложение PE2F.4.60 и применим его к функции $(f_i|d_i)$ по функции c_i :

$$(f_i|d_i) = (c_i|d_i) \& (f_i|d_i \wedge c_i) + \sim (c_i|d_i) \& (f_i|d_i \wedge \neg c_i), \quad (4.76)$$

На шаге $i+1$ применим разложение PE2F.4.60 с функцией c_{i+1} к главным операндам полученного разложения. Левый операнд:

$$(f_i|d_i \wedge c_i) = (c_{i+1}|d_i \wedge c_i) \& (f_i|d_i \wedge c_i \wedge c_{i+1}) + \\ \sim (c_{i+1}|d_i \wedge c_i) \& (f_i|d_i \wedge c_i \wedge \neg c_{i+1}), \quad (4.77)$$

Правый операнд:

$$(f_i|d_i \wedge \neg c_i) = (c_{i+1}|d_i \wedge \neg c_i) \& (f_i|d_i \wedge \neg c_i \wedge c_{i+1}) + \\ \sim (c_{i+1}|d_i \wedge \neg c_i) \& (f_i|d_i \wedge \neg c_i \wedge \neg c_{i+1}), \quad (4.78)$$

Многочисленное применение разложения PE2F.4.60 дает следующую структуру главного операнда на уровне k иерархии:

$$(f_i|d_i \wedge l_i \wedge \dots \wedge l_k). \quad (4.79)$$

где l_i – литерал вида c_i или $\neg c_i$. Легко видеть, что при увеличении значения k характеристическая функция области определенности операнда (4.79) при правильном выборе функций $c_i \dots c_k$ стремится к 0. Аналогичную структуру имеет вспомогательный операнд:

$$(c_k|d_i \wedge l_i \wedge \dots \wedge l_{k-1}). \quad (4.80)$$

Как следствие, иерархическое разложение не базе разложения PE2F.4.60 является сходящимся в том смысле, что с увеличением уровня иерархии степень определенности всех операндов стремится к 0. Эффективность разложения конкретной заданной частично определенной функции существенно зависит от выбираемой последовательности булевых функций $C_1 \dots C_k$.

Пример 4.4. Пусть задана функция $(f|1)$, где $f = x_1 \wedge \neg x_2 \wedge x_3 \vee x_1 \wedge \neg x_2 \wedge x_4 \vee x_1 \wedge \neg x_3 \wedge x_4 \vee x_2 \wedge \neg x_3 \wedge x_4$. Выполним иерархическое разложение функции $(f|1)$, используя соотношение PE2F.4.56, являющееся частным случаем соотношения PE2F.4.60. На первом уровне иерархии выберем $C_1 = x_1 \vee x_2$:

$$\begin{aligned} (f|1) &= (x_1 \vee x_2|1) \& (f|x_1 \vee x_2) + \sim (x_1 \vee x_2|1) \& (f|\neg(x_1 \vee x_2)) = \\ & (x_1 \vee x_2|1) \& (\neg x_2 \wedge x_3 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4 | x_1 \vee x_2) + \\ & \sim (x_1 \vee x_2|1) \& (0|\neg(x_1 \vee x_2)). \end{aligned}$$

На втором уровне иерархии выполним разложение операнда $((\neg x_2 \wedge x_3 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4) | x_1 \vee x_2)$ по функции $C_2 = x_2 \wedge x_3$:

$$\begin{aligned} (\neg x_2 \wedge x_3 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4 | x_1 \vee x_2) &= \\ (x_2 \wedge x_3|1) \& (\neg x_2 \wedge x_3 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4 | (x_1 \vee x_2) \wedge (x_2 \wedge x_3)) + \\ & \sim (x_2 \wedge x_3|1) \& (\neg x_2 \wedge x_3 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4 | (x_1 \vee x_2) \wedge \neg(x_2 \wedge x_3)) = \\ (x_2 \wedge x_3|1) \& (0 | (x_1 \vee x_2) \wedge (x_2 \wedge x_3)) + \\ & \sim (x_2 \wedge x_3|1) \& (x_3 \vee x_4 | (x_1 \vee x_2) \wedge \neg(x_2 \wedge x_3)). \end{aligned}$$

Объединяя разложения на первом и втором уровнях, получаем иерархическое разложение:

$$\begin{aligned} (f|1) &= \\ & (x_1 \vee x_2|1) \& \\ & (x_2 \wedge x_3|1) \& \\ & (0 | (x_1 \vee x_2) \wedge (x_2 \wedge x_3)) + \\ & \sim (x_2 \wedge x_3|1) \& \\ & (x_3 \vee x_4 | (x_1 \vee x_2) \wedge \neg(x_2 \wedge x_3)) + \\ & \sim (x_1 \vee x_2|1) \& \\ & (0 | \neg(x_1 \vee x_2)). \end{aligned}$$

Выделяя в каждом частичном операнде функцию значения и переходя от частичных операций к булевым операциям, выводим

следующее формульное представление функции $f=(x_1 \vee x_2) \wedge \neg(x_2 \wedge x_3) \wedge (x_3 \vee x_4)$.

Иерархические разложения на базе соотношений PE2F.4.57-PE2F.4.59 и PE2F.4.61-PE2F.4.63 строятся аналогичным образом.

4.6. Историческая справка

Разложения частично определенных функций предложены в работах А.А. Прихожего [53, 54, 56, 58-59, 67, 93, 105, 106], опубликованных начиная с 1991 года. Все ранее разработанные и описанные в известной литературе разложения, например [22, 79], предназначены только для полностью определенных булевых функций. В то же время, частичные разложения применимы не только к не полностью определенным, но и к полностью определенным функциям, при этом они разлагают полностью определенные функции на не полностью определенные части.

Идея использования частичных одноместных и двухместных операций для построения разложений функций является новой. В известной литературе применение частичных операций для построения разложений функций не описано. Разложения частично определенных функций по характеристической булевой функции также не имеют аналогов в полностью определенной булевой алгебре. Частным случаем частичных разложений по двум двухместным частичным операциям, примененным к полностью определенной булевой функции, является разложение Шеннона. Идея построения иерархических частичных разложений оригинальна в том, что степень определенности продуктов разложения уменьшается с увеличением уровня иерархии, на котором находятся эти продукты.

Следует заметить, что прямого обобщения разложений Давио в частичной и смешанной алгебрах построить не удалось. Это объясняется тем, что операция исключающее или так же как и операция эквивалентность не обладает достаточными возможностями по оперированию неопределенностью.

Глава 5. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧАСТИЧНО ОПРЕДЕЛЕННЫХ ФУНКЦИЙ

5.1. Совершенная дизъюнктивно-конъюнктивная форма частично определенной функции

Совершенная дизъюнктивно-конъюнктивная форма частично определенной функции (СДКФЧОФ) строится по частичной совершенной дизъюнктивно-конъюнктивной форме ЧСДКФ (2.34)-(2.37), которая в свою очередь построена по полному табличному представлению частичной функции. Схожесть СДКФЧОФ с ЧСДКФ состоит в том, что в обоих случаях используется один набор базовых частичных операций, а именно дизъюнкция, конъюнкция, отрицание. Основное отличие заключается в использовании в СДКФЧОФ пар булевых переменных в VCR форме вместо частичных переменных, используемых в ЧСДКФ. Вследствие этого, в отличие от ЧСДКФ, представляющей частичную функцию, СДКФЧОФ представляет частично определенную функцию такую, что по ней полностью восстанавливается исходная частичная функция. Замечательным свойством получаемой частично определенной функции является возможность быстрого построения не полностью определенной функции, лежащей в основе соответствующего класса Т-эквивалентных частичных функций (см. главу 3). Переход от ЧСДКФ к СДКФЧОФ выполняется на следующих шагах:

1. кодирование всех частичных переменных парами булевых переменных в VCR форме;
2. представление операции # идентификации неопределенности dc парой булевых функций в VCR форме;
3. представление константы dc в виде $(0|0)$;
4. замена всех вхождений частичных переменных и константы dc в ЧСДКФ на их представления в VCR форме и построение формулы в смешанной алгебре;
5. замена всех вхождений операции # соответствующими выражениями в смешанной алгебре;
6. преобразование полученной формулы смешанной алгебры к паре булевых функций в VCR форме, описывающих час-

точно определенную функцию, зависящую от переменных значения и переменных области определенности;

7. преобразование полученной пары булевых функций к другой паре булевых функций, представляющих частично определенную функцию, зависящую только от булевых переменных значения.

Положим, что частичная функция Ω , представленная в ЧСДКФ (2.34), зависит от n частичных переменных y_1, \dots, y_n . Кодировем частичные переменные парами булевых переменных в VCR форме (3.1): $(v_1|d_1), \dots, (v_n|d_n)$, где v_i – булевы переменные значения; d_i – булевы переменные области определенности.

Представим операцию $\#(v_i|d_i)$ *идентификации неопределенности* парой булевых функций $(f_i|h_i) = \#(v_i|d_i)$ в VCR форме, зависящих от переменных v_i, d_i . Для этого построим табл.5.1, используя определение операции $\#$, данное в табл.2.10. В столбце y_i приведены значения частичной переменной, которые кодируются парами булевых значений, приведенных в столбцах v_i и d_i . В столбце $\#y_i$ даны значения операции идентификации dc , зависящие от значений переменной y_i . Столбцы f_i и h_i кодируют значения операции $\#$ в VCR форме. Используя эти столбцы, получаем:

$$\#y_i = \#(v_i|d_i) = (1|\neg d_i). \quad (5.1)$$

Как видим, результат выполнения операции $\#$ зависит от переменной области определенности d_i и не зависит от переменной значения v_i .

Заменяя все вхождения частичных переменных y_i парами булевых переменных $(v_i|d_i)$ и все вхождения операции $\#$ ее представлением (5.1), трансформируем ЧСДКФ, представляющую частичную функцию Ω (2.34)-(2.37), в СДКФЧОФ, представляющую частично определенную функцию ω :

Таблица 5.1

Частичная операция идентификации dc

| y_i | v_i | d_i | $\#y_i$ | f_i | h_i |
|-------|-------|-------|---------|-------|-------|
| 0 | 0 | 1 | dc | 0,1 | 0 |
| 1 | 1 | 1 | dc | 0,1 | 0 |
| dc | 0,1 | 0 | 1 | 1 | 1 |

$$\text{СДКФЧОФ}(\Omega) = \theta_1 + (-\theta_0) \& (0|0), \quad (5.2)$$

где частично определенные функции

$$\theta_1 = \sum_{\Omega(a_1, \dots, a_n)=1} \left(\&_{i=1}^n (v_i | d_i)^{a_i} \right), \quad (5.3)$$

$$\theta_0 = \sum_{\Omega(a_1, \dots, a_n)=0} \left(\&_{i=1}^n (v_i | d_i)^{a_i} \right) \quad (5.4)$$

при

$$(v_i | d_i)^{a_i} = \begin{cases} \sim(v_i | d_i), & \text{если } a_i=0, \\ (v_i | d_i), & \text{если } a_i=1, \\ (1|\sim d_i), & \text{если } a_i=d_i. \end{cases} \quad (5.5)$$

Здесь $a=(a_1, \dots, a_n)$ – набор значений частичных переменных. Заметим, что частично определенные функции θ_1 и θ_0 представлены в дизъюнктивно-конъюнктивной форме.

Преобразование формулы (5.2) к паре булевых функций $(f|h)$, описывающих частично определенную функцию ω , зависящую от переменных значения и переменных области определенности, выполняется с использованием законов (3.6), (3.12), (3.16).

Преобразование частично определенной функции $\omega=(f|h)$ к не полностью определенной функции $\omega^*=(f^*|h^*)$, представляющей класс Т-эквивалентности, которому принадлежит частичная функция Ω , выполняется путем подстановки значения 1 вместо всех вхождений всех переменных областей определенности d_i в функциях f и h , а также путем замены переменных v_i на булевы переменные x_i .

Пример 5.1. Продемонстрируем построение СДКФЧОФ на примере частичной функции Ψ , описанной табл.5.2. В соответствии с соотношениями (5.2)-(5.5) построим сначала формулы смешанной алгебры для функций θ_1 и θ_0 , построенных на множестве конститuent 1 и конститuent 0 соответственно:

$$\theta_1 = \sim(v_1|d_1) \& \sim(v_2|d_2) + \sim(v_1|d_1) \& (v_2|d_2) + \sim(v_1|d_1) \& (1|\sim d_2),$$

$$\theta_0 = (v_1|d_1) \& \sim(v_2|d_2).$$

Таблица 5.2

Пример частичной функции Ψ

| y_1 | y_2 | $\Psi(y_1, y_2)$ |
|-------|-------|------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | dc | 1 |
| 1 | 0 | 0 |
| 1 | 1 | dc |
| 1 | dc | dc |
| dc | 0 | dc |
| dc | 1 | dc |
| dc | dc | dc |

Подставляя выражения для функций θ_1 и θ_0 в формулу (5.2), получаем СДКФЧОФ(ω):

$$\begin{aligned} \text{СДКФЧОФ}(\omega) = & (\sim(v_1|d_1) \& \sim(v_2|d_2) + \\ & \sim(v_1|d_1) \& (v_2|d_2) + \\ & \sim(v_1|d_1) \& (1|\sim d_2)) + \\ & \sim((v_1|d_1) \& \sim(v_2|d_2)) \& (0|0). \end{aligned}$$

Преобразуем полученное выражение к паре булевых функций:

$$\begin{aligned} (f|h) = & (\sim(\sim v_1 \wedge \sim v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee v_2 \wedge d_2) + \\ & (\sim v_1 \wedge v_2 | d_1 \wedge d_2 \vee v_1 \wedge d_1 \vee \sim v_2 \wedge d_2) + \\ & (\sim v_1 | d_1 \wedge \sim d_2 \vee v_1 \wedge d_1) + \\ & (\sim(v_1|d_1) + (v_2|d_2)) \& (0|0) = \\ = & (\sim v_1 | d_1 \wedge d_2 \vee v_1 \wedge d_1) + \\ & (\sim v_1 | d_1 \wedge \sim d_2 \vee v_1 \wedge d_1) + \\ & (\sim v_1 \vee v_2 | d_1 \wedge d_2 \vee \sim v_1 \wedge d_1 \vee v_2 \wedge d_2) \& (0|0) = \\ = & (\sim v_1 | v_1 \wedge d_1 \vee \sim v_1 \wedge d_1 \wedge d_2 \vee \sim v_1 \wedge d_1 \wedge \sim d_2) + \\ & (0 | v_1 \wedge \sim v_2 \wedge d_1 \wedge d_2) = \\ = & (\sim v_1 | d_1) + (0 | v_1 \wedge \sim v_2 \wedge d_1 \wedge d_2) = \\ = & (\sim v_1 | v_1 \wedge \sim v_2 \wedge d_1 \wedge d_2 \vee \sim v_1 \wedge d_1). \end{aligned}$$

Функция $f = \sim v_1$ зависит только от переменной v_1 . Функция $h = v_1 \wedge \sim v_2 \wedge d_1 \wedge d_2 \vee \sim v_1 \wedge d_1$ зависит от всех четырех переменных v_1, v_2, d_1, d_2 .

Преобразуем частично определенную функцию $\omega = (f|h)$ к не полностью определенной функции $\omega' = (f'|h')$. Для этого подставляем

Таблица 5.3
Не полностью определенная функция ω'

| x_1 | x_2 | ω' |
|-------|-------|-----------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | dc |

переменные x_1, x_2 вместо переменных V_1, V_2 и значение 1 вместо переменных d_1, d_2 в функциях f и h :

$$\omega' = (f|h') = (\neg x_1 | x_1 \wedge \neg x_2 \wedge 1 \wedge 1 \vee \neg x_1 \wedge 1) = (\neg x_1 | \neg x_1 \vee \neg x_2).$$

Полученная не полностью определенная функция ω' описывается табл.5.3. Она полностью соответствует частичной функции Ψ , описываемой табл.5.2. \square

Совершенная дизъюнктивно-конъюнктивная форма СДКФЧОФ может быть непосредственно построена по табличному представлению не полностью определенной функции $\psi(x_1, \dots, x_n)$. Аргументы функции ψ принимают значения из множества B и являются полностью определенными, поэтому в СДКФЧОФ (5.2)-(5.5) переменные области определенности d_i могут быть заменены на значение 1, а переменные значения V_i могут быть заменены на переменные X_i . Соотношения (5.2)-(5.5) в этом случае упрощаются:

$$\begin{aligned} \text{СДКФЧОФ}(\psi) = & \sum_{\psi(a_1, \dots, a_n)=1} + \left(\bigotimes_{i=1}^n (X_i^{a_i} | 1) \right) + \\ & \bigotimes_{\psi(a_1, \dots, a_n)=0} \left(+ (0 | X_i^{a_i}) \right), \end{aligned} \quad (5.6)$$

где

$$X_i^{a_i} = \begin{cases} X_i, & \text{если } a_i = 1, \\ \neg X_i, & \text{если } a_i = 0. \end{cases} \quad (5.7)$$

Формула смешанной алгебры (5.6) представлена частичной суммой двух частично определенных слагаемых: дизъюнкцией конъюнкций полностью определенных литералов X_i и $\neg X_i$; конъюнк-

цией дизъюнкций частично определенных констант 0 с областями определенности, описываемыми литералами x_i и $\neg x_i$.

Пример 5.2. Продемонстрируем построение СДКФЧОФ (5.6) на примере не полностью определенной функции ω' из табл.5.3:

$$\begin{aligned} \text{СДКФЧОФ}(\omega') &= ((\neg x_1|1) \& (\neg x_2|1) + (\neg x_1|1) \& (x_2|1)) + \\ &\quad ((0|x_1) + (0|\neg x_2)) = \\ &= (\neg x_1 \wedge \neg x_2|1) + (\neg x_1 \wedge x_2|1) + (0|x_1 \wedge \neg x_2) = \\ &= (\neg x_1|1) + (0|x_1 \wedge \neg x_2) = \\ &= (\neg x_1|x_1 \wedge \neg x_2 \vee \neg x_1) = (\neg x_1|\neg x_1 \vee \neg x_2). \square \end{aligned}$$

5.2. Совершенная конъюнктивно-дизъюнктивная форма частично определенной функции

Если в совершенной дизъюнктивно-конъюнктивной форме частично определенной функции ведущую роль играла частичная операция дизъюнкция, то в *совершенной конъюнктивно-дизъюнктивной форме* частично определенной функции (СКДФЧОФ) ведущую роль играет частичная операция конъюнкция. СКДФЧОФ обладает теми же основными свойствами, что и СДКФЧОФ. Первичными в этой форме являются переменные v_i и d_j , сама форма принимает значения из множества M .

СКДФЧОФ строится по частичной совершенной конъюнктивно-дизъюнктивной форме ЧСКДФ (2.48) и полному табличному представлению частичной функции. При заменах переменных, операций и констант в выражении (2.48) частичной функции Ω , подобных тем, которые выполнены для СДКФЧОФ, получаем:

$$\begin{aligned} \text{СКДФЧОФ}(\Omega) &= \left[\&_{\Omega(a_1, \dots, a_n)=0} \left(\bigoplus_{i=1}^n \sim (v_i | d_i)^{a_i} \right) \right] \& \quad (5.8) \\ &\quad \left[\bigoplus_{\Omega(a_1, \dots, a_n)=1} \left(\&_{i=1}^n (v_i | d_i)^{a_i} \right) + (0|0) \right], \end{aligned}$$

где аргументы операций конъюнкции и дизъюнкции определяются выражением (5.5).

Пример 5.3. Он является продолжением примера 5.1. Продемонстрируем построение СКДФЧОФ на примере частичной функции Ψ , описанной той же табл.5.2. В соответствии с соотношением (5.8) записываем:

$$\begin{aligned} \text{СКДФЧОФ}(\Psi) &= (\sim(v_1|d_1)+(v_2|d_2)) \& \\ & (\sim(v_1|d_1)\&\sim(v_2|d_2)) + \\ & \sim(v_1|d_1)\&(v_2|d_2) + \\ & \sim(v_1|d_1)\&(1|\neg d_2) + (0|0)) = \\ & = (\neg v_1|v_1 \wedge \neg v_2 \wedge d_1 \wedge d_2 \vee \neg v_1 \wedge d_1). \end{aligned}$$

Как видим, выражения для СДКФЧОФ и СКДФЧОФ различны, однако они трансформируются к одной и той же паре булевых функций значения и области определенности. \square

Все построенные выше формы представления частичной и не полностью определенной функции базируются на использовании множества конститuent 1 и множества конститuent 0. Следующая СКДФЧОФ строится по не полностью определенной функции ψ на множестве конститuent 0 и множестве конститuent dc .

$$\begin{aligned} \text{СКДФЧОФ}(\psi) &= \&_{\psi(a_1, \dots, a_n)=0} \left(\bigoplus_{i=1}^n (x_i^{-a_i} | 1) \right) \& \\ & \&_{\psi(a_1, \dots, a_n)=dc} \left(\bigoplus_{i=1}^n (1 | x_i^{-a_i}) \right), \end{aligned} \quad (5.9)$$

где $\neg a_i$ есть булево отрицание a_i . Компоненты выражения (5.9) определены соотношением (5.7).

Пример 5.4. Продемонстрируем построение СКДФЧОФ (5.9) на примере не полностью определенной функции ω' из табл.5.3:

$$\begin{aligned} \text{СКДФЧОФ}(\omega') &= ((\neg x_1|1)+(x_2|1) \& ((1|\neg x_1)+(1|\neg x_2))) = \\ & = (\neg x_1 \vee x_2 | 1) \& (1 | \neg x_1 \vee \neg x_2) = \\ & = (\neg x_1 \vee x_2 | \neg x_1 \vee \neg x_2 \vee x_1 \wedge \neg x_2) = \\ & = (\neg x_1 \vee x_2 | \neg x_1 \vee \neg x_2). \end{aligned}$$

Хотя полученная пара булевых функций и отличается первой составляющей от пары, полученной в примере 5.2, однако она также корректно описывает не полностью определенную функцию из табл.5.3. □

5.3. Дизъюнктивно-конъюнктивная форма частично определенной функции

Построенные выше совершенные формы представления частично определенных функций обладают тем замечательным свойством, что отталкиваясь от них можно вернуться к той частичной функции, на основе которой они построены. Другими словами, построенные на базе этих форм частично определенные функции наследуют все свойства частичных функций, из которых они проистекают.

Если отказаться от требования наследования свойств частичных функций частично определенными функциями, то возможно построение других форм, в ряде случаев более эффективных в применении. Наследование свойств не будет выполняться, если заменить сильную частичную совершенную дизъюнктивно-конъюнктивную форму ЧСДКФ, по которой строится СДКФЧОФ, на одну из трех эквивалентных слабых форм: частичную дизъюнктивно-конъюнктивную форму ЧДКФ, частичную сокращенную дизъюнктивно-конъюнктивную форму ЧСкДКФ или частичную минимальную дизъюнктивно-конъюнктивную форму ЧМДКФ (см. раздел 2.8). Метод построения дизъюнктивно-конъюнктивной формы частично определенной функции ДКФЧОФ один и тот же во всех трех случаях. Он заключается в трансформации соотношений (5.2)-(5.5) к следующему виду:

$$\begin{aligned} \text{ДКФЧОФ}(\Omega) = & \left[\begin{array}{c} + \\ \Omega(a_1, \dots, a_n)=1 \end{array} \left(\begin{array}{c} \& (v_i | d_i)^{a_i} \\ i=1, \dots, n, \\ a_i \neq dc \end{array} \right) \right] + & (5.10) \\ \sim & \left[\begin{array}{c} + \\ \Omega(a_1, \dots, a_n)=0 \end{array} \left(\begin{array}{c} \& (v_i | d_i)^{a_i} \\ i=1, \dots, n, \\ a_i \neq dc \end{array} \right) \right] \& (0 | 0), \end{aligned}$$

где

$$(v_i | d_i)^{a_i} = \begin{cases} (v_i | d_i), & \text{если } a_i = 1, \\ \sim (v_i | d_i), & \text{если } a_i = 0. \end{cases} \quad (5.11)$$

Построение ДКФЧОФ по ЧДКФ, ЧСкДКФ или ЧМДКФ отличается лишь использованием различных исходных таблиц.

Пример 5.5. Пусть исходная частичная функция Δ задана табл.2.13, по которой в примере 2.4 строится ЧСкДКФ. В соответствии с (5.10), (5.11) записываем:

$$\begin{aligned} \text{ДКФЧОФ}(\Delta) = & ((v_1|d_1) \& (v_2|d_2) + \\ & (v_1|d_1) \& (v_3|d_3) + \\ & \sim (v_2|d_2) \& (v_3|d_3) + \\ & \sim (\sim (v_1|d_1) \& (v_2|d_2) \& (v_3|d_3) + \\ & \sim (v_2|d_2) \& \sim (v_3|d_3)) \& (0|0). \end{aligned}$$

Преобразуем полученное выражение к паре булевых функций:

$$\begin{aligned} (f/h) = & (v_1 \wedge v_2 \vee v_1 \wedge v_3 \vee \neg v_2 \wedge v_3 | \\ & \neg v_1 \wedge v_2 \vee v_3 \wedge d_1 \wedge d_2 \wedge d_3 \vee \neg v_2 \wedge \neg v_3 \wedge d_1 \wedge d_2 \wedge d_3 \vee \\ & v_1 \wedge v_2 \wedge d_1 \wedge d_2 \vee v_1 \wedge v_3 \wedge d_1 \wedge d_3 \vee \neg v_2 \wedge v_3 \wedge d_2 \wedge d_3). \end{aligned}$$

Восстанавливаем по частично определенной функции (f/h) частичную функцию Δ' . Она описывается табл.5.4. Сравнение частичной функции Δ' (табл.5.4) с исходной частичной функцией Δ (табл.2.13) показывает, что они отличаются тем, что вместо одного набора $(dc,0,0)$ значений аргументов, на котором определена функция Δ , используется два набора $(0,0,0)$ и $(1,0,0)$, на которых определена функция Δ' . Очевидно, что функции Δ и Δ' принадлежат одному классу Т-эквивалентности, поскольку они сводятся к одной не

Таблица 5.4

Восстановленная по ДКФЧОФ частичная функция Δ'

| № | y_1 | y_2 | y_3 | Функция Δ' |
|---|-------|-------|-------|-------------------|
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 1 | dc | 1 |
| 3 | 1 | dc | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | dc | 0 | 1 | 1 |
| 7 | dc | dc | dc | dc |

Таблица 5.5

Не полностью определенная функция δ

| x_1 | x_2 | x_3 | Функция δ |
|-------|-------|-------|------------------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | dc |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

полностью определенной функции δ , описываемой табл.5.5. Функция δ представляется также в виде

$$\delta = (x_1 \wedge x_2 \vee x_1 \wedge x_3 \vee \neg x_2 \wedge x_3 \mid x_1 \vee \neg x_2 \vee x_3). \square$$

5.4. Конъюнктивно-дизъюнктивная форма частично определенной функции

Построение конъюнктивно-дизъюнктивной формы частично определенной функции КДФЧОФ базируется на трансформации слабой конъюнктивно-дизъюнктивной формы представления частичной функции (2.54) к следующему виду:

$$\text{ДКФЧОФ}(\Omega) = \left[\&_{\Omega(a_1, \dots, a_n)=0} \left(+_{\substack{i=1, \dots, n, \\ a_i \neq dc}} \sim (v_i \mid d_i)^{a_i} \right) \right] \& \quad (5.12)$$

$$\left[+_{\Omega(a_1, \dots, a_n)=1} \left(\&_{\substack{i=1, \dots, n, \\ a_i \neq dc}} (v_i \mid d_i)^{a_i} \right) + (0 \mid 0) \right],$$

где аргументы частичных операций конъюнкция и дизъюнкция определяются выражением (5.11). Как и ДКФЧОФ, КДФЧОФ не обеспечивает полное наследование свойств частичной функции, по которой она строится. Гарантируется, однако, принадлежность всех получаемых частичных функций одному классу Т-эквивалентности.

Таблица 5.6

Восстановленная по КДФЧОФ частичная функция Δ''

| № | y_1 | y_2 | y_3 | Функция Δ'' |
|---|-------|-------|-------|--------------------|
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 1 | dc | 1 |
| 3 | 1 | dc | 1 | 1 |
| 4 | dc | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 |
| 7 | dc | dc | dc | dc |

Пример 5.6. Построим в соответствии с (5.12) КДФЧОФ по частичной функции Δ , заданной табл.2.13:

$$\begin{aligned} \text{КДФЧОФ}(\Delta) = & ((v_1|d_1) + \sim(v_2|d_2) + \sim(v_3|d_3)) \& \\ & (v_2|d_2) + (v_3|d_3)) \& \\ & ((v_1|d_1) \& (v_2|d_2) + \\ & (v_1|d_1) \& (v_3|d_3) + \\ & \sim(v_2|d_2) \& (v_3|d_3) + (0|0)). \end{aligned}$$

Преобразуем построенную формулу к паре булевых функций:

$$\begin{aligned} (f/h) = & (v_1 \wedge v_2 \vee v_1 \wedge v_3 \vee \sim v_2 \wedge v_3 | \\ & \sim v_1 \wedge v_2 \vee v_3 \wedge d_1 \wedge d_2 \wedge d_3 \vee \sim v_2 \wedge v_3 \wedge d_1 \wedge d_2 \wedge d_3 \vee \\ & v_1 \wedge v_2 \wedge d_1 \wedge d_2 \vee v_1 \wedge v_3 \wedge d_1 \wedge d_3 \vee \sim v_2 \wedge \sim v_3 \wedge d_2 \wedge d_3). \end{aligned}$$

Частичная функция Δ'' , восстановленная по частично определенной функции (f/h) , описывается табл.5.6. Легко видеть, что она слегка отличается от частичной функции Δ' из примера 5.5, описываемой табл.5.4. В то же время, функция Δ'' принадлежит к тому же классу Т-эквивалентности, что и функции Δ и Δ' . Этот класс определяется не полностью определенной функцией δ из примера 5.5. \square

5.5. Обобщение формы Риды-Маллера для частично определенной функции

Как следует из главы 3, выразительные возможности частичной операции исключающее или ниже выразительных возможностей других частичных операций. В связи с этим, на базе операции исключающее или не удастся построить по заданной частичной

функции формы представления, аналогичные формам, рассмотренным выше и построенным на базе операций дизъюнкция и конъюнкция.

В то же время мы приведем пример расширенной формы Рида-Маллера ФРМЧОФ, конструируемой по заданной не полностью определенной функции ψ с использованием частичных операций исключающее или (\odot), дизъюнкция (+), конъюнкция (&), отрицание (\sim):

$$\begin{aligned} \text{ФРМЧОФ}(\psi) = & \left(\bigoplus_{\psi(a_1, \dots, a_n)=1} \left(\bigotimes_{i=1}^n (x_i^{a_i} | 1) \right) \right) + \\ & \bigotimes_{\psi(a_1, \dots, a_n)=0} \left(\bigoplus_{i=1}^n (0 | x_i^{a_i}) \right), \end{aligned} \quad (5.13)$$

где $x_i^{a_i}$ определяется соотношением (5.7).

Пример 5.7. Построим ФРМЧОФ (5.13) по не полностью определенной функции ω' из табл.5.3:

$$\begin{aligned} \text{ФРМЧОФ}(\omega') &= ((\neg x_1 | 1) \& (\neg x_2 | 1) \odot (\neg x_1 | 1) \& (x_2 | 1)) + \\ & \quad ((0 | x_1) \odot (0 | \neg x_2)) = \\ &= (\neg x_1 \wedge \neg x_2 | 1) \odot (\neg x_1 \wedge x_2 | 1) + (0 | x_1 \wedge \neg x_2) = \\ &= ((\neg x_1 \wedge \neg x_2) \oplus (\neg x_1 \wedge x_2) | 1) + (0 | x_1 \wedge \neg x_2) = \\ &= (\neg x_1 | 1) + (0 | x_1 \wedge \neg x_2) = \\ &= (\neg x_1 | x_1 \wedge \neg x_2 \vee \neg x_1) = (\neg x_1 | \neg x_1 \vee \neg x_2). \quad \square \end{aligned}$$

5.6. Историческая справка

Формы представления частичных функций предложены в работах С.К. Клини [38 (1952)], М. Дамиани и Д. деМишели [20, 21 (1993)] и др. Табличная полная и интервальная форма представления не полностью определенных булевых функций использована в работах [4, 6, 14, 15, 17-21, 23, 24, 37, 45, 46, 73, 86, 88, 92]. Графовое представление не полностью определенных функций исследо-

вано в работах [1, 13, 14, 23, 24, 32, 33, 39, 41, 53, 54, 79-81]. В работах [53, 93] форма «значение-область определенности» представления не полностью определенной функции положена в основу смешанной алгебры. Совершенная дизъюнктивно-конъюнктивная и совершенная конъюнктивно-дизъюнктивная формы представления частично определенных функций предложены в работах [58, 108]. Описанные в настоящей главе формы представления функций вытекают из смешанной алгебры и являются новыми.

Глава 6. ИСПОЛЬЗОВАНИЕ СМЕШАННОЙ АЛГЕБРЫ ДЛЯ ОПТИМИЗАЦИИ ЛОГИЧЕСКИХ СИСТЕМ

6.1. Задача оптимизации частично определенной логической системы

Рассмотрим логическую систему, описываемую частично определенной булевой функцией φ , представленной в VCR форме парой булевых переменных (функций) v и d : $\varphi = (v|d)$. Функция φ определена табл.3.1. Она принимает значение 0 на наборе (0,1) значений аргументов v и d и принимает значение 1 на наборе (1,1). На двух других наборах (0,0) и (1,0) функция φ принимает одно и то же значение d . Другими словами, если $d=0$, переменная v становится несущественной в частично определенной функции φ . Значит, в этом случае мы можем выбрать произвольное значение переменной v . Приведенные рассуждения остаются справедливыми, если булевы переменные v и d заменить произвольными n местными булевыми функциями $v(x_1, \dots, x_n)$ и $d(x_1, \dots, x_n)$, а частично определенную функцию $\varphi = (v|d)$, зависящую от булевых переменных v и d , заменить на частично определенную функцию $\varphi(x_1, \dots, x_n) = (v(x_1, \dots, x_n)|d(x_1, \dots, x_n))$.

Из приведенных рассуждений следует, что при $d=1$ значение функции v фиксировано, при $d=0$ значение функции v произвольно. Изменив значение $v(x_1, \dots, x_n)$ хотя бы на одном наборе значений переменных x_1, \dots, x_n , на котором $d=0$, получим новую функцию $u(x_1, \dots, x_n)$ такую, что $\varphi = (v|d) = (u|d)$. Легко видеть, когда выполняется равенство $(v|d) = (u|d)$, тогда выполняется равенство $v \wedge d = u \wedge d$.

Каков диапазон булевых функций u , способных заменить булеву функцию v , являющуюся составной частью частично определенной функции φ ? Если 0^{on} – множество конститuent 1 функции u , то этот диапазон описывается соотношением:

$$(v \wedge d)^{on} \subseteq u^{on} \subseteq (w \neg d)^{on}. \quad (6.1)$$

Реальные границы диапазона (6.1) зависят от вида функций v и d . Если $d=0$, то $0^{on} \subseteq u^{on} \subseteq 1^{on}$, или, другими словами, функция u может быть любой булевой функцией, зависящей от n аргументов. Здесь $0^{on} = \emptyset$ – множество конститент 1 булевой функции константа 0; 1^{on} – множество наборов значений векторного аргумента, на ко-

торых функция константа 1 имеет единичное значение, а это есть все булево пространство. Если $d=1$, то $V^{0n} \subseteq U^{0n} \subseteq V^{0n}$, а это означает равенство $U=V$. Если $d=V$, то $V^{0n} \subseteq U^{0n} \subseteq 1^{0n}$, или, другими словами, функция U может быть любой вплоть до константы 1, включающей функцию V по множеству конститuent 1. В любом случае границами диапазона являются функции $V \wedge d$ и $V \neg d$, причем первая функция является нижней границей, вторая функция является верхней границей диапазона. Упростим запись соотношения (6.1) до

$$V \wedge d \leq U \leq V \neg d. \quad (6.2)$$

Свяжем с булевой функцией U некоторую *функцию стоимости* $\chi(U)$ и рассмотрим задачу дискретной оптимизации в виде:

$$\min_{V \wedge d \leq U \leq V \neg d} \chi(U). \quad (6.3)$$

Критерием оптимизации в задаче (6.3) выбран, без нарушения общности, минимум функции стоимости. Критерием оптимизации может быть также максимум функции стоимости, при этом переформулировка задачи (6.3) очевидна. Допустимым множеством или пространством оптимизации в задаче (6.3) является описанный выше диапазон функций U . Физическая или техническая интерпретация функции стоимости χ может быть самой разнообразной. Прежде всего, это размер представления функции U в том или ином базисе, измеряемый, например, числом элементарных операций, входящих в формулу представления U . Если речь идет о технической реализации булевой функции U , то функцией стоимости χ может явиться занимаемая площадь кристалла, время задержки распространения сигнала, потребляемая или рассеиваемая мощность и т.д. Задача оптимизации (6.3) может решаться с учетом ограничений на другие параметры функции U или ее реализации.

Для решения задачи оптимизации (6.3) введем *операцию минимизации* $\min(V \wedge d)$, возвращающую ту функцию U из диапазона $V \wedge d \leq U \leq V \neg d$, для которой $\chi(U)=\min$. Операция $U=\min(V \wedge d)$ должна быть конкретизирована для каждой конкретизации постановки задачи (6.3). В настоящей главе мы приведем несколько примеров конкретизации операции минимизации, включая минимизацию размера описания функций, представленных дизъюнктивной нор-

мальной формой ДНФ, диаграммами двоичных решений ДДР или другими формами.

Операция минимизации обладает также общими свойствами, которые мы изучим в настоящей главе. Изучение общих свойств предполагает, что операция однозначно определена для всех функций $(\mathcal{V}d)$, и это определение дается единым алгоритмом, не зависящим от специфических особенностей конкретной пары булевых функций ν и d .

6.2. Диапазоны булевых функций и операции над диапазонами

Пусть булева функция U принадлежит диапазону $b_U^{low} \leq U \leq b_U^{high}$ и булева функция W принадлежит диапазону $b_W^{low} \leq W \leq b_W^{high}$, где b_U^{low} и b_W^{low} – функции, являющиеся нижними границами диапазонов, b_U^{high} и b_W^{high} – функции, являющиеся верхними границами диапазонов. Диапазон для U называется *непрерывным*, если $(b_U^{low})^{on} \subseteq (b_U^{high})^{on}$ и для каждого множества δ конституент 1 такого, что $(b_U^{low})^{on} \subseteq \delta \subseteq (b_U^{high})^{on}$ найдется функция t из диапазона для U такая, что $(t)^{on} = \delta$. Далее будем полагать, что диапазоны для U и W непрерывные.

Для заданного непрерывного диапазона булевых функций U важно знать, сколько функций он включает. Вычислим множество $\Delta U = (b_U^{high})^{on} \setminus (b_U^{low})^{on}$. Обозначим через $|\Delta U|$ мощность этого множества. Число различных функций, принадлежащих диапазону, равно числу различных подмножеств множества ΔU , а оно, в свою очередь, равно $2^{|\Delta U|}$.

Пример 6.1. Пусть булева функция U зависит от двух булевых переменных x_1, x_2 . Универсальное множество U включает четыре набора значений переменных: $U = \{00, 01, 10, 11\}$. Построим диапазон функций $x_1 \leq U \leq x_2$. Нижней границей диапазона является функция $b_U^{low} = x_1$, верхней границей - $b_U^{high} = x_2$. Множество включения для нижней границы диапазона есть $(x_1)^{on} = \{10, 11\}$, для верхней границы - $(x_2)^{on} = \{01, 11\}$. Очевидно, что множество $\{10, 11\}$ не включается в множество $\{01, 11\}$. Значит, построенный диапазон для U не является непрерывным.

Теперь построим другой диапазон функций $x_1 \wedge x_2 \leq U \leq x_1 \vee x_2$. Множество включения (конституент 1) для нижней границы диапазона есть $(x_1 \wedge x_2)^{on} = \{11\}$, для верхней границы - $(x_1 \vee x_2)^{on} = \{01, 10, 11\}$. Очевидно, что выполняется включение $\{11\} \subseteq \{01, 10, 11\}$. Множество $\Delta U = \{01, 10\}$ состоит из двух элементов. Следовательно, диапазон включает четыре функции, которым соответствуют четыре множества конституент 1: $\delta_1 = \{11\}$, $\delta_2 = \{01, 11\}$, $\delta_3 = \{10, 11\}$, $\delta_4 = \{01, 10, 11\}$. Этими функциями являются $x_1 \wedge x_2$, x_2 , x_1 , $x_1 \vee x_2$. Таким образом, построенный диапазон для U является непрерывным. \square

Если над функциями U и W , принадлежащими заданным диапазонам, выполняются различные булевы операции, то спрашивается, каким диапазонам принадлежат результаты булевых операций и являются ли эти диапазоны непрерывными? Начнем с операции инвертирования: $r = \neg U$. Результат операции инвертирования принадлежит диапазону

$$\neg b_U^{high} \leq \neg U \leq \neg b_U^{low}. \quad (6.4)$$

При выполнении инвертирования U , нижняя и верхняя границы диапазона для U также инвертируются. Более того, инвертированная нижняя граница U становится верхней границей r , а инвертированная верхняя граница U становится нижней границей r . Непрерывность диапазона для r непосредственно следует из непрерывности диапазона для U .

Пример 6.2. Пусть функция U принадлежит непрерывному диапазону $0 \leq U \leq x_1 \vee \neg x_2$, включающему восемь булевых функций. Результат $r = \neg U$ инвертирования функции U принадлежит в соответствии с (6.4) непрерывному диапазону $\neg x_1 \wedge x_2 \leq r \leq 1$, включающему также восемь булевых функций. \square

Если функция r определяется посредством булевой операции конъюнкция в виде $r = U \wedge W$, то эта функция принадлежит диапазону

$$b_U^{low} \wedge b_W^{low} \leq r \leq b_U^{high} \wedge b_W^{high}. \quad (6.5)$$

Нижней границей диапазона для r является конъюнкция нижних границ диапазонов для U и W . Верхняя граница диапазона для r вычисляется как конъюнкция верхних границ диапазонов для U и W .

Диапазон для Γ является непрерывным, поскольку включение $(b_u^{low} \wedge b_w^{low})^{on} \subseteq (b_u^{high} \wedge b_w^{high})^{on}$ следует из двух включений $(b_u^{low})^{on} \subseteq (b_u^{high})^{on}$ и $(b_w^{low})^{on} \subseteq (b_w^{high})^{on}$.

Пример 6.3. Пусть функция U принадлежит непрерывному диапазону $0 \leq U \leq \neg X_1 \vee X_2$, включающему восемь булевых функций, а функция W принадлежит непрерывному диапазону $X_1 \leq W \leq X_1 \vee X_2$, включающему две булевы функции. Булева конъюнкция функций U и W дает результирующую функцию $\Gamma = U \wedge W$, принадлежащую в соответствии с (6.5) непрерывному диапазону $0 \leq \Gamma \leq X_2$, включающему четыре булевы функции. □

Если функция Γ определяется посредством булевой операции дизъюнкция в виде $\Gamma = U \vee W$, то эта функция принадлежит диапазону

$$b_u^{low} \vee b_w^{low} \leq \Gamma \leq b_u^{high} \vee b_w^{high}. \quad (6.6)$$

Как и в случае операции конъюнкции, нижняя граница диапазона для Γ вычисляется через нижние границы диапазонов для U и W , а верхняя граница для Γ вычисляется через верхние границы диапазонов для U и W . Диапазон для Γ является непрерывным, поскольку включение $(b_u^{low} \vee b_w^{low})^{on} \subseteq (b_u^{high} \vee b_w^{high})^{on}$ следует из двух включений $(b_u^{low})^{on} \subseteq (b_u^{high})^{on}$ и $(b_w^{low})^{on} \subseteq (b_w^{high})^{on}$.

Пример 6.4. Пусть, как и в примере 6.3, функция U принадлежит непрерывному диапазону $0 \leq U \leq \neg X_1 \vee X_2$, а функция W принадлежит непрерывному диапазону $X_1 \leq W \leq X_1 \vee X_2$. Булева дизъюнкция функций U и W дает результирующую функцию $\Gamma = U \vee W$, принадлежащую в соответствии с (6.6) непрерывному диапазону $X_1 \leq \Gamma \leq 1$, включающему четыре булевы функции. □

Изучим диапазон для функции $\Gamma = \min(U|W)$. Согласно (6.2), диапазон для Γ записывается в виде $U \wedge W \leq \Gamma \leq U \vee \neg W$. Границы диапазона точно определены, когда функции U и W не варьируются. Когда функции варьируются, границы требуют уточнения. Рассмотрим три случая: U варьируется, W нет; U не варьируется, W варьируется; варьируются обе функции U и W .

Пусть функция U изменяется в диапазоне $b_u^{low} \leq U \leq b_u^{high}$ а функция W фиксирована. Тогда диапазон для Γ есть объединение диапазонов для $\min(U|W)$ при различных значениях U . В множестве всех диапазонов для $\min(U|W)$ крайними являются $b_u^{low} \wedge W \leq \Gamma \leq b_u^{low}$

$\vee \neg W$ и $b_u^{high} \wedge W \leq r \leq b_u^{high} \vee \neg W$. Промежуточный диапазон строится следующим образом. Определим функцию U_i в виде $(U_i)^{on} = (b_u^{low})^{on} \cup U_i$, где $U \subseteq \Delta U$. Множество ΔU есть разность множеств конститuent 1 функций b_u^{high} и b_u^{low} , т.е. $\Delta U = (b_u^{high})^{on} \setminus (b_u^{low})^{on}$. Упорядочим элементы множества ΔU некоторым произвольным образом: $\Delta U = \{\delta_1, \dots, \delta_i, \dots, \delta_k\}$ и определим множество $U = \{\delta_1, \dots, \delta_i\}$. Рассмотрим два соседних диапазона i и $i+1$ для r и убедимся, что они образуют один объединенный непрерывный диапазон. Запишем диапазоны i и $i+1$: $U_i \wedge W \leq r \leq U_i \vee \neg W$ и $U_{i+1} \wedge W \leq r \leq U_{i+1} \vee \neg W$. Объединенный диапазон $U_i \wedge W \leq r \leq U_{i+1} \vee \neg W$. Для его нижней и верхней границ справедливо включение: $(U_i \wedge W)^{on} \subseteq (U_{i+1} \vee \neg W)^{on}$. Более того, объединенный диапазон включает соседние функции U_i и U_{i+1} . Приведенные рассуждения справедливы при любом порядке элементов в множестве ΔU . Значит, результатом объединения всех диапазонов для $\min(U|W)$ при различных значениях U является непрерывный диапазон $b_u^{low} \wedge W \leq r \leq b_u^{high} \vee \neg W$.

Пример 6.5. Пусть функция U принадлежит непрерывному диапазону $X_1 \wedge X_2 \leq U \leq X_1 \vee X_2$, а функция $W = X_1 \oplus X_2$ фиксирована. Тогда булева функция $r = \min(U|W)$ принадлежит диапазону $(X_1 \wedge X_2) \wedge (X_1 \oplus X_2) \leq r \leq (X_1 \vee X_2) \vee \neg (X_1 \oplus X_2)$, который преобразуется к эквивалентному диапазону $0 \leq r \leq 1$. Таким образом, диапазон для r включает все шестнадцать булевых функций, зависящих от переменных X_1, X_2 . □

Пусть теперь функция U фиксирована, а функция W изменяется в диапазоне $b_w^{low} \leq W \leq b_w^{high}$. Тогда диапазон для r есть объединение диапазонов для $\min(U|W)$ при различных значениях W . В множестве всех диапазонов для $\min(U|W)$ крайними являются $U \wedge b_w^{low} \leq r \leq U \vee \neg b_w^{low}$ и $U \wedge b_w^{high} \leq r \leq U \vee \neg b_w^{high}$. Для нижних границ крайних диапазонов выполняется $(U \wedge b_w^{low})^{on} \subseteq (U \wedge b_w^{high})^{on}$. Введем универсальное полное множество W конститuent 1. Очевидно, что $(\neg b_w^{low})^{on} = W \setminus (b_w^{low})^{on}$ и $(\neg b_w^{high})^{on} = W \setminus (b_w^{high})^{on}$. Поскольку справедливо включение $(b_w^{low})^{on} \subseteq (b_w^{high})^{on}$, выполняется включение $(\neg b_w^{high})^{on} \subseteq (\neg b_w^{low})^{on}$. Как следствие, для верхних границ крайних диапазонов выполняется включение $(U \vee \neg b_w^{high})^{on} \subseteq (U \vee \neg b_w^{low})^{on}$. Значит, первый крайний диапазон полностью покрывает второй крайний диапазон. Более того, первый крайний диапазон покрывает все промежуточ-

ные диапазоны. Таким образом, результатом объединения всех диапазонов для $\min(U|W)$ при различных значениях W является непрерывный диапазон $U \wedge b_W^{low} \leq r \leq U \vee \neg b_W^{low}$.

Пример 6.6. Пусть функция $U = X_1 \oplus X_2$ фиксирована, а функция W принадлежит непрерывному диапазону $X_1 \wedge X_2 \leq W \leq X_1 \vee X_2$. Тогда булева функция $r = \min(U|W)$ принадлежит диапазону $(X_1 \oplus X_2) \wedge (X_1 \wedge X_2) \leq r \leq (X_1 \oplus X_2) \vee \neg(X_1 \vee X_2)$, который преобразуется к эквивалентному диапазону $0 \leq r \leq \neg X_1 \vee \neg X_2$. Этот диапазон включает восемь из шестнадцати булевых функций, зависящих от переменных X_1, X_2 . \square

Пусть функция U изменяется в диапазоне $b_U^{low} \leq U \leq b_U^{high}$, а функция W изменяется в диапазоне $b_W^{low} \leq W \leq b_W^{high}$. Тогда диапазон для r есть объединение всех диапазонов для $\min(U|W)$ при различных значениях U и различных значениях W . Если U изменяется, а W нет, то диапазоном для r является $b_U^{low} \wedge W \leq r \leq b_U^{high} \vee \neg W$. Пусть W принимает граничные значения. Тогда крайними диапазонами являются $b_U^{low} \wedge b_W^{low} \leq r \leq b_U^{high} \vee \neg b_W^{low}$ и $b_U^{low} \wedge b_W^{high} \leq r \leq b_U^{high} \vee \neg b_W^{high}$. Легко видеть, что первый граничный диапазон полностью покрывает второй граничный диапазон также как и все промежуточные диапазоны. Следовательно, результатом объединения всех диапазонов для $\min(U|W)$ при различных значениях U и W является непрерывный диапазон

$$b_U^{low} \wedge b_W^{low} \leq r \leq b_U^{high} \vee \neg b_W^{low}. \quad (6.7)$$

Пример 6.7. Пусть функция U принадлежит непрерывному диапазону $X_1 \wedge X_2 \leq U \leq X_1 \vee X_2$, а функция W принадлежит непрерывному диапазону $0 \leq W \leq X_1 \oplus X_2$. Тогда булева функция $r = \min(U|W)$ принадлежит диапазону $(X_1 \wedge X_2) \wedge 0 \leq r \leq (X_1 \vee X_2) \vee \neg(0)$, который преобразуется к эквивалентному диапазону $0 \leq r \leq 1$. Этот диапазон включает все шестнадцать булевых функций, зависящих от двух переменных. \square

Предложенный выше *метод диапазонов функций* и техника выполнения операций над диапазонами применимы к выражениям смешанной алгебры в случае независимых диапазонов, т.е. диапазонов, на которых функции выбираются независимо друг от друга. В случае зависимых диапазонов и согласованного выбора функций при выполнении операций минимизации мы предлагаем *метод ис-*

следования граничных точек, позволяющий построить множество граничных функций для заданного выражения смешанной алгебры. Зная граничные функции, мы можем построить диапазон функций, соответствующий выражению. В общем случае, диапазон, полученный методом диапазонов, отличается от диапазона, полученного методом граничных точек. Более того, второй диапазон всегда не шире (уже) первого диапазона.

Пример 6.8. Пусть задана функция $r = \min(u|w) \wedge \min(w| \min(u|w))$. Спрашивается, каков диапазон функций для r ? Применяя соотношение 6.2 для $s = \min(u|w)$, получаем диапазон $u \wedge w \leq s \leq u \vee \neg w$. Используя соотношение (6.7), для $t = \min(w|s)$, имеем $w \wedge u \wedge w \leq t \leq w \vee \neg(u \wedge w)$ или $w \wedge u \leq t \leq 1$. Выполняя операцию конъюнкции $r = s \wedge t$, получаем диапазон $u \wedge w \wedge w \wedge u \leq s \wedge t \leq (u \vee \neg w) \wedge 1$ или $u \wedge w \leq s \wedge t \leq u \vee \neg w$. Построенный диапазон соответствует, на самом деле, формуле смешанной алгебры $r = s \wedge \min(w|z)$, где $s = \min(u|w)$ и $z = \min(u|w)$. Принципиальным моментом является то, что функции s и z выбираются из диапазонов независимо друг от друга, хотя для обеих функций это один и тот же диапазон.

При другой интерпретации формулы для r два вхождения операции $s = \min(u|w)$ предполагают использование одной и той же функции s для каждого из вхождений. Тогда $r = s \wedge \min(w|s)$. В этом случае метод диапазонов не обеспечивает построение точного диапазона для r . Здесь оказывается применимым методом граничных точек. В определении функции $r = s \wedge t$ используются две операции минимизации $s = \min(u|w)$ и $t = \min(w|s)$, причем операция $s = \min(u|w)$ имеет два вхождения. Для функции s граничными точками являются $u \wedge w$ и $u \vee \neg w$, а для функции t граничными точками являются $w \wedge s$ и $w \vee \neg s$. Всего существует четыре комбинации пар точек для s и t : $s = u \wedge w$ и $t = w \wedge s$; $s = u \vee \neg w$ и $t = w \wedge s$; $s = u \wedge w$ и $t = w \vee \neg s$; $s = u \vee \neg w$ и $t = w \vee \neg s$. Вычислим функцию r для каждой из комбинаций:

1. $r = u \wedge w \wedge w \wedge u \wedge w = u \wedge w$.
2. $r = (u \vee \neg w) \wedge w \wedge (u \vee \neg w) = u \wedge w$.
3. $r = u \wedge w \wedge (w \vee \neg(u \wedge w)) = u \wedge w$.
4. $r = (u \vee \neg w) \wedge (w \vee \neg(u \vee \neg w)) = u \wedge w$.

Таким образом, формула смешанной алгебры $r = \min(u|w) \wedge \min(w|\min(u|w))$ во второй интерпретации описывает диапазон, состоящий только из одной функции, а именно $r = u \wedge w$. □

В заключение параграфа построим все возможные непрерывные диапазоны двухместных булевых функций и выясним, сколько функций находится в каждом из диапазонов. Поскольку существует 16 двухместных булевых функций, представим диапазоны матрицей D размерности 16×16 , показанной на рис.6.1. Строки и столбцы матрицы соответствуют функциям. функции нумеруются $0, \dots, 15$. Элементами матрицы являются символы ' \leq ', ' \geq ', '=' и '-'. Если элемент $d_{ij} = \leq$, то функция i является нижней границей, а функция j верхней границей непрерывного диапазона. Если элемент $d_{ij} = \geq$, то функция i является верхней границей, а функция j нижней границей непрерывного диапазона. Если элемент $d_{ij} = =$, то функции i и j совпадают, следовательно, диапазон является непрерывным и включает ровно одну функцию. Если элемент $d_{ij} = -$, то диапазон не является непрерывным. Поясним алгоритм расстановки символов в матрице D . Возьмем, например, элемент $d_{5,13}$. Двухместная булева функция с номером 5 определяется двоичным вектором 0101, осуществляющим выбор подмножества $\delta_1 = \{01, 11\}$ конститuent 1 из

| | | | | | | | | | | | | | | | | | | | |
|-----|----|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | |
| D = | 0 | = | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | | |
| | 1 | ≥ | = | - | ≤ | - | ≤ | - | ≤ | - | ≤ | - | ≤ | - | ≤ | - | ≤ | | |
| | 2 | ≥ | - | = | - | - | - | - | - | - | - | - | - | - | - | - | - | | |
| | 3 | ≥ | ≥ | - | = | - | - | - | - | - | - | - | - | ≤ | - | - | - | ≤ | |
| | 4 | ≥ | - | - | - | = | - | ≤ | ≤ | - | - | - | - | - | ≤ | - | ≤ | ≤ | |
| | 5 | ≥ | ≥ | - | - | - | = | - | - | ≤ | - | - | - | - | - | ≤ | - | ≤ | |
| | 6 | ≥ | - | ≥ | - | ≥ | - | = | ≤ | - | - | - | - | - | - | - | ≤ | ≤ | |
| | 7 | ≥ | - | ≥ | ≥ | ≥ | ≥ | ≥ | = | - | - | - | - | - | - | - | - | ≤ | |
| | 8 | ≥ | - | - | - | - | - | - | - | = | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | ≤ | |
| | 9 | ≥ | ≥ | - | - | - | - | - | - | - | ≥ | = | - | ≤ | - | ≤ | - | ≤ | |
| | 10 | ≥ | - | ≥ | - | - | - | - | - | - | ≥ | - | = | - | - | - | ≤ | ≤ | |
| | 11 | ≥ | ≥ | - | ≥ | - | - | - | - | - | ≥ | ≥ | - | = | - | - | - | ≤ | |
| | 12 | ≥ | - | - | - | ≥ | - | - | - | ≥ | - | - | - | - | = | ≤ | - | ≤ | |
| | 13 | ≥ | ≥ | - | - | - | ≥ | - | - | ≥ | ≥ | - | - | - | - | = | - | ≤ | |
| | 14 | ≥ | - | ≥ | - | ≥ | - | ≥ | - | ≥ | - | ≥ | - | ≥ | - | ≥ | - | = | ≤ |
| | 15 | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | ≥ | = | |

Рисунок 6.1 – Непрерывные диапазоны двухместных булевых функций

| | | | | | | | | | | | | | | | | |
|-------|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 1 | 2 | 2 | 4 | 2 | 4 | 4 | 8 | 2 | 4 | 4 | 8 | 4 | 8 | 8 | 16 |
| 1 | 2 | 1 | - | 2 | - | 2 | - | 4 | - | 2 | - | 4 | - | 4 | - | 8 |
| 2 | 2 | - | 1 | - | - | - | 2 | 4 | - | - | 2 | - | - | - | 4 | 8 |
| 3 | 4 | 2 | - | 1 | - | - | - | 2 | - | - | - | 2 | - | - | - | 4 |
| 4 | 2 | - | - | - | 1 | - | 2 | 4 | - | - | - | - | 2 | - | 4 | 8 |
| 5 | 4 | 2 | - | - | - | 1 | - | 2 | - | - | - | - | - | 2 | - | 4 |
| 6 | 4 | - | 2 | - | 2 | - | 1 | 2 | - | - | - | - | - | - | 2 | 4 |
| Q = 7 | 8 | 4 | 4 | 2 | 4 | 2 | 2 | 1 | - | - | - | - | - | - | - | 2 |
| 8 | 2 | - | - | - | - | - | - | - | 1 | 2 | 2 | 4 | 2 | 4 | 4 | 8 |
| 9 | 4 | 2 | - | - | - | - | - | - | 2 | 1 | - | 2 | - | 2 | - | 4 |
| 10 | 4 | - | 2 | - | - | - | - | - | 2 | - | 1 | - | - | - | 2 | 4 |
| 11 | 8 | 4 | - | 2 | - | - | - | - | 4 | 2 | - | 1 | - | - | - | 2 |
| 12 | 4 | - | - | - | 2 | - | - | - | 2 | - | - | - | 1 | 2 | 2 | 4 |
| 13 | 8 | 4 | - | - | - | 2 | - | - | 4 | 2 | - | - | 2 | 1 | - | 2 |
| 14 | 8 | - | 4 | - | 4 | - | 2 | - | 4 | - | 2 | - | 2 | - | 1 | 2 |
| 15 | 16 | 8 | 8 | 4 | 8 | 4 | 4 | 2 | 8 | 4 | 4 | 2 | 4 | 2 | 2 | 1 |

Рисунок 6.2 – Число булевых функций в непрерывных диапазонах

универсального множества $U = \{00, 01, 10, 11\}$. Двухместная булева функция с номером 13 определяется двоичным вектором 1101, осуществляющим выбор подмножества $\delta_2 = \{00, 01, 11\}$ из множества U . Очевидно, выполняется включение $\delta_1 \subseteq \delta_2$. Функции с номерами 5 и 13 являются нижней и верхней границей непрерывного диапазона, следовательно, $\alpha_{5,13} = '\leq'$. Возьмем теперь элемент $\alpha_{3,6}$. Функциям с номерами 3 и 6 соответствуют двоичные векторы 0011 и 0110, которые не дают включение одного подмножества конститuent 1 в другое подмножество. Следовательно, функции не являются крайними точками какого-либо непрерывного диапазона, отсюда $\alpha_{3,6} = '\neq'$.

Матрица D является симметричной относительно главной диагонали. Подсчитывая общее количество символов ' \leq ', ' \geq ' и '=' в матрице, находим, что 136 диапазонов из 256 являются непрерывными, а 120 диапазонов таковыми не являются. Диапазоны, у которых нижняя и верхняя границы совпадают, находятся на главной диагонали матрицы. Диапазоны, не являющиеся непрерывными, расположены внутри матрицы. Число булевых функций в каждом непрерывном диапазоне описывается матрицей Q , показанной на рис.6.2. В правом верхнем и левом нижнем углах матрицы расположены диапазоны, включающие максимальное число функций 16.

При движении от этих углов к главной диагонали матрицы Q число функций в диапазонах сокращается до 1. Следует заметить, что сокращение не всегда является монотонным.

Аналогичные матрицы D и Q размерностями 256×256 , 65536×65536 , ..., $2^{2^n} \times 2^{2^n}$ могут быть построены для трехместных, четырехместных и n -местных булевых функций. Размеры диапазонов функций при этом увеличиваются. Так максимальный размер диапазона булевых функций, зависящих от n аргументов, равен числу функций $\nu = 2^{2^n}$. С увеличением n размер пространства оптимизации в задаче (6.3) увеличивается стремительно, возможности по минимизации значения функции стоимости $\chi(u)$ возрастают.

6.3. Свойства операции минимизации

Рассмотрим частично определенную функцию $(\nu|d)$ и операцию минимизации $\min(\nu|d)$. Возьмем логическое отрицание (инверсию) функции $\sim(\nu|d) = (\neg\nu|d)$. Минимизируем инверсию с помощью операции $\min(\neg\nu|d)$. Зададимся вопросом, могут ли быть равными минимум инверсии функции и инверсия минимума функции? Другими словами, выполняется ли равенство:

$$\neg \min(\nu|d) = \min(\neg\nu|d). \quad (6.8)$$

Проанализируем соотношения (6.8) с двух позиций. Во-первых, совпадают ли диапазоны изменения булевых функций левой и правой частей равенства? Во-вторых, совпадает ли инверсия результата минимизации булевой функции ν с результатом минимизации инверсии булевой функции $\neg\nu$?

Ответим сначала на первый вопрос. Согласно (6.2), операция $u = \min(\nu|d)$ осуществляет поиск минимальной функции u в диапазоне $\nu \wedge d \leq u \leq \nu \neg d$. Инверсия операции $\neg u = \neg \min(\nu|d)$ инвертирует диапазон до $\neg \nu \wedge d \leq \neg u \leq \neg \nu \neg d$. Операция $w = \min(\neg\nu|d)$ осуществляет поиск минимальной функции w в диапазоне $\neg \nu \wedge d \leq w \leq \neg \nu \neg d$. Как видим, инвертированный результат операции минимизации функции и результат операции минимизации инвертированной функции принадлежат одному и тому же диапазону функций.

Ответ на второй вопрос требует более детального рассмотрения и конкретизации операции минимизации. Рассматривая вопрос о равенстве двух функций, одна из которых является инверсией третьей функции, и равенстве параметров функций, необходимо, прежде всего, учесть форму представления каждой из функций, а также процедуру выполнения операции инвертирования. Рассмотрим две конкретизации, в которых функцией стоимости является размер представления булевой функции.

В первой конкретизации воспользуемся двумя формами представления одновременно: дизъюнктивной нормальной формой ДНФ и конъюнктивной нормальной формой КНФ. Функцией стоимости $\chi(U)$ будет число элементарных конъюнкций в ДНФ (элементарных дизъюнкций в КНФ) или суммарное число литералов в ДНФ (КНФ). Суммарное число двухместных операций дизъюнкции и конъюнкции в ДНФ или КНФ равно суммарному числу литералов минус единица. Рассмотрим все множество допустимых форм представления функций V, U, W и их инверсий: ДНФ(V), КНФ(V), ДНФ($\neg V$), КНФ($\neg V$), ДНФ(U), КНФ(U), ДНФ($\neg U$), КНФ($\neg U$), ДНФ(W), КНФ(W). С целью проверки равенства (6.8) и с учетом особенностей реализации операции минимизации потребуем, чтобы следующие пары функций были представлены в одной форме: V и U , $\neg V$ и W , $\neg U$ и W . Тогда следующие сочетания форм являются допустимыми:

1. ДНФ(V), ДНФ(U), ДНФ($\neg U$), ДНФ($\neg V$), ДНФ(W);
2. ДНФ(V), ДНФ(U), КНФ($\neg U$), КНФ($\neg V$), КНФ(W);
3. КНФ(V), КНФ(U), ДНФ($\neg U$), ДНФ($\neg V$), ДНФ(W);
4. КНФ(V), КНФ(U), КНФ($\neg U$), КНФ($\neg V$), КНФ(W).

Проанализируем сочетания 1-4 с позиций функции стоимости. Очевидно, что в сочетаниях 1, 2 $\chi(\text{ДНФ}(V)) \geq \chi(\text{ДНФ}(U))$, а в сочетаниях 3, 4 $\chi(\text{КНФ}(V)) \geq \chi(\text{КНФ}(U))$, поскольку U есть минимизированная V . Аналогично, в сочетаниях 1, 3 $\chi(\text{ДНФ}(\neg V)) \geq \chi(\text{ДНФ}(W))$, а в сочетаниях 2, 4 $\chi(\text{КНФ}(\neg V)) \geq \chi(\text{КНФ}(W))$. В сочетании 2 $\chi(\text{ДНФ}(U)) = \chi(\text{КНФ}(\neg U))$, а в сочетании 3 $\chi(\text{КНФ}(U)) = \chi(\text{ДНФ}(\neg U))$, поскольку здесь $\neg U$ получается из U применением законов де Моргана, сохраняющих число элементарных конъюнкций (дизъюнкций) или число литералов. В то же время, в сочетании 1 $\chi(\text{ДНФ}(\neg U)) \neq \chi(\text{ДНФ}(U))$, а в

сочетании 4 $\chi(\text{КНФ}(\neg u)) \neq \chi(\text{КНФ}(u))$, поскольку здесь $\neg u$ получается из u раскрытием скобок и упрощением выражений, в результате чего $\neg u$ не является минимизированной. Кроме того, в сочетании 2 $\chi(\text{КНФ}(\neg v)) = \chi(\text{ДНФ}(v))$, а в сочетании 3 $\chi(\text{ДНФ}(\neg v)) = \chi(\text{КНФ}(v))$ опять же в силу применения законов де Моргана. В то же время, в сочетании 1 $\chi(\text{ДНФ}(\neg v)) \neq \chi(\text{ДНФ}(v))$, а в сочетании 4 $\chi(\text{КНФ}(\neg v)) \neq \chi(\text{КНФ}(v))$. Таким образом, из приведенных соотношений следует, что равенство (6.8) выполняется для сочетаний форм 2 и 3, а для сочетаний 1 и 4 равенство не выполняется. В дальнейшем для реализации операции минимизации будем применять сочетание форм 2 и 3, а сочетания 2 и 4 применять не будем. Благодаря этому, равенство (6.8) можно рассматривать как закон.

Во второй конкретизации воспользуемся одной из двух форм представления булевой функции: упорядоченной диаграммой двоичных решений УДДР и сокращенной упорядоченной диаграммой двоичных решений СУДДР. Функцией стоимости $\chi(\text{ДДР}(u))$ будет число нетерминальных вершин в диаграмме двоичных решений. Поскольку операция инвертирования заключается лишь в переименовании терминальной вершины 0 в терминальную вершину 1 и переименовании вершины 1 в вершину 0, то эта операция сохраняет значение функции стоимости $\chi(\text{ДДР}(u)) = \chi(\text{ДДР}(\neg u))$ для диаграмм всех типов. Следовательно, равенство (6.8) выполняется отдельно для УДДР и отдельно для СУДДР. Оно становится законом для диаграмм двоичных решений.

Изучим свойства операции минимизации, связанные с двухместными булевыми операциями. Рассмотрим две частично определенные функции $(v_1|d)$ и $(v_2|d)$ с различными функциями значения v_1 , v_2 и одинаковой функцией области определенности d . Применим к ним операцию частичной конъюнкции: $(v_1|d) \& (v_2|d) = (v_1 \wedge v_2|d)$. Исследуем условия выполнения следующих соотношений:

$$\min(v_1|d) \wedge \min(v_2|d) = \min(v_1 \wedge v_2|d), \quad (6.9)$$

$$\min(v_1 \wedge \min(v_2|d)|d) = \min(v_1 \wedge v_2|d), \quad (6.10)$$

$$\min(\min(v_1|d) \wedge \min(v_2|d)|d) = \min(v_1 \wedge v_2|d), \quad (6.11)$$

Проверим совпадение диапазонов изменения булевых функций, стоящих в левой и правой частях равенств. Результат операции

$u = \min(v_1|d)$ лежит в диапазоне $v_1 \wedge d \leq u \leq v_1 \vee \neg d$, операции $w = \min(v_2|d)$ – в диапазоне $v_2 \wedge d \leq w \leq v_2 \vee \neg d$ и операции $z = \min(v_1 \wedge v_2|d)$ – в диапазоне $v_1 \wedge v_2 \wedge d \leq z \leq v_1 \wedge v_2 \vee \neg d$. Конъюнкция диапазонов для u и w дает диапазон $v_1 \wedge d \wedge v_2 \wedge d \leq u \wedge w \leq (v_1 \vee \neg d) \wedge (v_2 \vee \neg d)$, преобразуемый к диапазону $v_1 \wedge v_2 \wedge d \leq u \wedge w \leq v_1 \wedge v_2 \vee \neg d$. Таким образом, диапазоны для $u \wedge w$ и z совпадают, что доказывает с этой точки зрения равенство (6.9).

С целью доказательства равенства (6.10) построим диапазон для результата суперпозиции двух операций минимизации $s = \min(v_1 \wedge \min(v_2|d)|d)$. Им является диапазон $(v_1 \wedge (v_2 \wedge d)) \wedge d \leq s \leq v_1 \wedge (v_2 \vee \neg d) \vee \neg d$, преобразуемый к диапазону $v_1 \wedge v_2 \wedge d \leq s \leq v_1 \wedge v_2 \vee \neg d$. Видим, что диапазоны для s и z совпадают, а это доказывает с этой точки зрения равенство (6.10).

С целью доказательства равенства (6.11) построим диапазон для результата суперпозиции трех операций минимизации $t = \min(\min(v_1|d) \wedge \min(v_2|d)|d)$. Им является диапазон $(v_1 \wedge d) \wedge (v_2 \wedge d) \wedge d \leq t \leq (v_1 \vee \neg d) \wedge (v_2 \vee \neg d) \vee \neg d$, преобразуемый к диапазону $v_1 \wedge v_2 \wedge d \leq t \leq v_1 \wedge v_2 \vee \neg d$. Видим, диапазоны для t и z совпадают, что доказывает равенство (6.11).

Совпадают ли результаты левой и правой частей каждого из равенств (6.9)-(6.11) с точки зрения различных конкретизаций задачи оптимизации (6.3)? Пусть булевы функции v_1, v_2, d представлены в ДНФ/КНФ формах, а функцией стоимости $\chi(u)$ является число элементарных конъюнкций (дизъюнкций) или общее число литералов. Тогда в общем случае результат левой части соотношения (6.9) не равен результату правой части соотношения, так как конъюнкция результатов раздельной минимизации двух булевых функций не лучше в общем случае результата совместной минимизации конъюнкции тех же функций. Значит, равенство (6.9) не выполняется для ДНФ/КНФ. В то же время для ДНФ/КНФ выполняются равенства (6.10), (6.11). Пусть теперь булевы функции v_1, v_2, d представлены в виде УДДР. Учитывая свойства алгоритмов реализации операций минимизации и операции конъюнкции над УДДР, заключаем, что два равенства (6.10)-(6.11) в этом случае выполняются, а равенство (6.9) может не выполняться.

Применим к частично определенным функциям $(v_1|d)$ и $(v_2|d)$ операцию частичной дизъюнкции: $(v_1|d)+(v_2|d)=(v_1\vee v_2|d)$. Исследуем условия выполнения следующих соотношений:

$$\min(v_1|d)\vee\min(v_2|d) = \min(v_1\vee v_2|d), \quad (6.12)$$

$$\min(v_1\vee\min(v_2|d)|d) = \min(v_1\vee v_2|d), \quad (6.13)$$

$$\min(\min(v_1|d)\vee\min(v_2|d)|d) = \min(v_1\vee v_2|d), \quad (6.14)$$

Диапазоны для u и w остаются прежними. Результат операции $z=\min(v_1\vee v_2|d)$ лежит в диапазоне $(v_1\vee v_2)\wedge d \leq z \leq v_1\vee v_2\vee \neg d$. Дизъюнкция диапазонов для u и w дает диапазон $v_1\wedge d\vee v_2\wedge d \leq u\wedge w \leq (v_1\vee \neg d)\vee(v_2\vee \neg d)$, преобразуемый к диапазону $(v_1\vee v_2)\wedge d \leq u\wedge w \leq v_1\vee v_2\vee \neg d$. Таким образом, диапазоны для $u\wedge w$ и z совпадают, что доказывает с этой точки зрения равенство (6.12). Результат суперпозиции двух операций минимизации $s=\min(v_1\vee\min(v_2|d)|d)$ лежит в диапазоне $(v_1\vee v_2\wedge d)\wedge d \leq s \leq v_1\vee(v_2\vee \neg d)\vee \neg d$, преобразуем к диапазону $(v_1\vee v_2)\wedge d \leq s \leq v_1\vee v_2\vee \neg d$. Видим, что диапазоны для s и z совпадают, а это доказывает равенство (6.13). Результат суперпозиции трех операций минимизации $t=\min(\min(v_1|d)\vee\min(v_2|d)|d)$ лежит в диапазоне $(v_1\wedge d)\vee(v_2\wedge d) \leq t \leq (v_1\vee \neg d)\vee(v_2\vee \neg d)\vee \neg d$, преобразуем к диапазону $(v_1\vee v_2)\wedge d \leq t \leq v_1\wedge v_2\vee \neg d$. Видим, диапазоны для t и z совпадают, что доказывает равенство (6.14).

Анализ условий идентичности результатов левой и правой частей каждого из равенств (6.12)-(6.14) для операции дизъюнкции совпадает с выполненным выше анализом для операции конъюнкции.

Рассмотрим далее две частично определенные функции $(\psi|d_1)$ и $(\psi|d_2)$ с одной функцией значения ψ и с различными функциями области определенности d_1, d_2 . Применим к ним операцию частичной конъюнкции: $(\psi|d_1)\&(\psi|d_2)=(\psi|d_1\wedge d_2\vee \neg v\wedge(d_1\vee d_2))$. Исследуем условия выполнения следующих равенств:

$$\min(\psi|d_1)\wedge\min(\psi|d_2) = \min(\psi|d_1\wedge d_2\vee \neg v\wedge(d_1\vee d_2)), \quad (6.15)$$

$$\min(\min(\psi|d_1)|d_2) = \min(\psi|d_1\wedge d_2). \quad (6.16)$$

Результат операции $u=\min(\psi|d_1)$ лежит в диапазоне $v\wedge d_1 \leq u \leq v\wedge \neg d_1$, операции $w=\min(\psi|d_2)$ – в диапазоне $e \ v\wedge d_2 \leq w \leq v\wedge \neg d_2$ и операции $z=\min(\psi|d_1\wedge d_2\vee \neg v\wedge(d_1\vee d_2))$ – в диапазоне $v\wedge d_1\wedge d_2\vee v\wedge \neg v\wedge$

$(d_1 \vee d_2) \leq z \leq w \neg (d_1 \wedge d_2 \vee \neg \vee \wedge (d_1 \vee d_2))$, преобразуем к диапазону $\vee \wedge d_1 \wedge d_2 \leq z \leq w \neg d_1 \wedge \neg d_2$. Конъюнкция диапазонов для u и w дает диапазон $\vee \wedge d_1 \wedge \vee \wedge d_2 \leq u \wedge w \leq (w \neg d_1) \wedge (w \neg d_2)$, преобразуемый к диапазону $\vee \wedge d_1 \wedge d_2 \leq u \wedge w \leq w \neg d_1 \wedge \neg d_2$. Диапазоны для $u \wedge w$ и z совпадают, равенство (6.15) выполнимо. Результат суперпозиции двух операций минимизации $s = \min(\min(\vee d_1) | d_2)$ лежит в диапазоне $\vee \wedge d_1 \wedge d_2 \leq s \leq w \neg d_1 \vee \neg d_2$, а операции $t = \min(\vee d_1 \wedge d_2)$ – в диапазоне $\vee \wedge d_1 \wedge d_2 \leq t \leq w \neg d_1 \vee \neg d_2$. Диапазоны для s и t совпадают, что доказывает выполнимость равенства (6.16).

Соотношения (6.9)-(6.16) могут быть модифицированы для других двухместных булевых операций, включая штрих Шеффера, стрелку Пирса, прямую и обратную импликации. Характер выводов по модифицированным соотношениям остается прежним.

6.4. *Оптимальное разложение поведения системы по булевой операции*

Разрабатываемые в этом параграфе разложения будем называть МЕО от английского Minimizing Expansion on Operation.

Пусть полностью определенная булева функция f представлена в виде $f = g \diamond h$, где g и h – некоторые булевы функции; \diamond - двухместная булева операция. Частичные разложения РЕО.4.10-РЕО.4.17 позволяют представить функцию ($f|1$) через частичные двухместные операции с уменьшенной степенью определенности операндов. Операция минимизации частично определенной функции позволяет выгодно использовать уменьшенную степень определенности операндов посредством следующих разложений:

$$g \wedge h = \min(g|h) \wedge \min(h|g \vee \neg h), \quad (6.17)$$

$$\neg(g \rightarrow h) = \neg(\min(g|\neg h) \rightarrow \min(h|g \vee h)), \quad (6.18)$$

$$\neg(g \leftarrow h) = \neg(\min(g|h) \leftarrow \min(h|\neg g \vee \neg h)), \quad (6.19)$$

$$g \vee h = \min(g|\neg h) \vee \min(h|\neg g \vee h), \quad (6.20)$$

$$g \downarrow h = \min(g|\neg h) \downarrow \min(h|\neg g \vee h), \quad (6.21)$$

$$g \leftarrow h = \min(g|h) \leftarrow \min(h|\neg g \vee \neg h), \quad (6.22)$$

$$g \rightarrow h = \min(g|\neg h) \rightarrow \min(h|g \vee h), \quad (6.23)$$

$$g|h = \min(g|h) | \min(h|g \vee \neg h). \quad (6.24)$$

Минимизирующие разложения (6.17)-(6.24), сокращенное название МЕО.6.17-МЕО.6.24, строятся непосредственно по разложениям РЕО.4.10-РЕО.4.17 посредством применения операции минимизации ко всем операндам и замены двухместной частичной операции на соответствующую двухместную булеву операцию. Хотя корректность разложений МЕО.6.17-МЕО.6.24 вытекает из корректности разложений РЕО.4.10-РЕО.4.17, мы проведем дополнительное доказательство полученных равенств. Возьмем, например, разложение МЕО.6.17. Результат операции $u = \min(g|h)$ лежит в диапазоне $g \wedge h \leq u \leq g \vee \neg h$, операции $w = \min(h|g \vee \neg h)$ – в диапазоне $g \wedge h \leq w \leq h \vee \neg(g \vee \neg h)$, преобразуемом к диапазону $g \wedge h \leq w \leq h$. Конъюнкция диапазонов для u и w дает диапазон $g \wedge h \leq u \wedge w \leq (g \vee \neg h) \wedge h$, преобразуемый к диапазону $g \wedge h \leq u \wedge w \leq g \wedge h$. Очевидно, что диапазон для $u \wedge w$ включает только одну функцию $g \wedge h$, что и доказывает разложение МЕО.6.17.

Сила разложений МЕО.6.17-МЕО.6.24 состоит в том, что при заданной структуре булевой функции они минимизируют части этой структуры по тем или иным критериям и сохраняют при этом функцию всей структуры.

Пример 6.9. Исходная булева функция f , зависящая от четырех переменных x_1, x_2, x_3, x_4 и представленная картой Карно, дана на рис.6.3. Алгебраическое представление функции f в минимальной дизъюнктивной нормальной форме (МДНФ), построенное по шести кубам карты Карно:

$$f = \neg x_1 \wedge x_2 \wedge x_4 \vee x_2 \wedge \neg x_3 \wedge x_4 \vee \neg x_1 \wedge \neg x_2 \wedge \neg x_4 \vee \\ \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \vee x_1 \wedge x_2 \wedge x_3 \wedge \neg x_4 \vee x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4.$$

Функцией стоимости $\chi(f)$ выберем число двухместных булевых операций конъюнкция и дизъюнкция в представлении функции f . Функция стоимости $\chi(\text{МДНФ}(f)) = 19$.

Воспользуемся разложением МЕО.6.17 по операции конъюнкция. Прежде всего, выберем функции g и h такие, что $f = g \wedge h$. Функции g и h выбираются в диапазоне функций от f до 1. Проанализируем два операнда $(g|h)$ и $(h|g \vee \neg h)$ в разложении МЕО.6.17. Если выбрать функцию g близкой к константе 1, то второй операнд будет почти полностью определенным и, поэтому, мало минимизи-

руемым. Точно также, если выбрать функцию h близкой к константе 1, то первый операнд будет мало минимизируемым. Далее, если выбрать функцию h близкой к функции f , то второй операнд будет также мало минимизируемым. Отсюда вытекает следующая стратегия: разумно выбрать $g=f$ и выбрать h между функцией f и константой 1 таким образом, чтобы одна из переменных в первом операнде стала несущественной. Для функции f , показанной на рис.6.3, выбираем функции g и h , показанные на рис.6.4. Функция g

f)

| | | | | | |
|-------|-------|---|---|---|---------|
| | x_1 | | | | |
| | 0 | 1 | 0 | 1 | |
| x_3 | 1 | 0 | 1 | 0 | } x_4 |
| | 0 | 1 | 1 | 0 | |
| | 1 | 0 | 0 | 1 | |
| | x_2 | | | | |

Рисунок 6.3 – Исходная булева функция f , представленная картой Карно

g)

| | | | | | |
|-------|-------|---|---|---|---------|
| | x_1 | | | | |
| | 0 | 1 | 0 | 1 | |
| x_3 | 1 | 0 | 1 | 0 | } x_4 |
| | 0 | 1 | 1 | 0 | |
| | 1 | 0 | 0 | 1 | |
| | x_2 | | | | |

 \wedge

| | | | | | |
|-------|-------|---|---|---|---------|
| | x_1 | | | | |
| | 0 | 1 | 1 | 1 | |
| x_3 | 1 | 0 | 1 | 1 | } x_4 |
| | 0 | 1 | 1 | 1 | |
| | 1 | 0 | 1 | 1 | |
| | x_2 | | | | |

Рисунок 6.4 – Булевы функции g и h , представленные картами Карно

(g/h)

| | | | | | |
|-------|-------|----|---|---|---------|
| | x_1 | | | | |
| | dc | 1 | 0 | 1 | |
| x_3 | 1 | dc | 1 | 0 | } x_4 |
| | dc | 1 | 1 | 0 | |
| | 1 | dc | 0 | 1 | |
| | x_2 | | | | |

 \wedge

| | | | | | |
|-------|-------|---|----|----|---------|
| | x_1 | | | | |
| | 0 | 1 | dc | 1 | |
| x_3 | 1 | 0 | 1 | dc | } x_4 |
| | 0 | 1 | 1 | dc | |
| | 1 | 0 | dc | 1 | |
| | x_2 | | | | |

Рисунок 6.5 – Частично определенные функции (g/h) и $(h/g \vee h)$

полностью совпадает с функцией f , в функции h четыре значения 0 функции f стали значениями 1. Тогда частично определенные функции $(g|h)$ и $(h|g\nu\bar{h})$, вычисленные по булевым функциям g и h , показаны на рис.6.5. Учитывая возможность замены dc произвольно на 0 или 1, МДНФ функции $min(g|h)$ строим по трем кубам карты Карно:

$$min(g|h) = x_1 \vee x_2 \wedge x_4 \vee \neg x_2 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(min(g|h)))=4$. Аналогично, МДНФ функции $min(h|g\nu\bar{h})$ строим по пяти кубам карты Карно:

$$min(h|g\nu\bar{h}) = \neg x_3 \vee x_1 \wedge \neg x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge x_4 \vee x_1 \wedge x_2 \wedge \neg x_4 \vee \neg x_1 \wedge \neg x_2 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(min(h|g\nu\bar{h})))=12$. Собирая во едино продукты разложения, получаем $\chi(min(g|h) \wedge min(h|g\nu\bar{h}))=4+1+12=17$. Таким образом, полученное представление функции f на две операции компактнее исходного представления в МДНФ. \square

Для булевых операций исключаящее или и эквивалентность, разложения ПЕО.4.18-ПЕО.4.19 трансформируются посредством применения операций дизъюнкции, конъюнкции, инверсии и минимизации в разложения:

$$g \oplus h = min(g|\bar{h}) \wedge min(\bar{h}|g) \vee min(\neg g|h) \wedge min(h|\neg g), \quad (6.25)$$

$$g \leftrightarrow h = min(g|h) \wedge min(h|g) \wedge min(\neg g|\bar{h}) \wedge min(\bar{h}|\neg g). \quad (6.26)$$

С учетом разложений ПЕО.4.21-ПЕО.4.28 разложения МЕО.6.17-МЕО.6.24 легко обобщаются на случай частично определенной функции $(f|d) = (g \circ h|d)$:

$$min(g \wedge h|d) = min(g|d \wedge h) \wedge min(h|d \wedge (g\nu\bar{h})), \quad (6.27)$$

$$min(\neg(g \rightarrow h)|d) = \neg(min(g|d \wedge \bar{h}) \rightarrow min(h|d \wedge (g\nu h))), \quad (6.28)$$

$$min(\neg(g \leftarrow h)|d) = \neg(min(g|d \wedge h) \leftarrow min(h|d \wedge (\neg g\nu\bar{h}))), \quad (6.29)$$

$$min(g \vee h|d) = min(g|d \wedge \bar{h}) \vee min(h|d \wedge (\neg g\nu h)), \quad (6.30)$$

$$min(g \downarrow h|d) = min(g|d \wedge \bar{h}) \downarrow min(h|d \wedge (\neg g\nu h)), \quad (6.31)$$

$$min(g \leftarrow h|d) = min(g|d \wedge h) \leftarrow min(h|d \wedge (\neg g\nu\bar{h})), \quad (6.32)$$

$$min(g \rightarrow h|d) = min(g|d \wedge \bar{h}) \rightarrow min(h|d \wedge (g\nu h)), \quad (6.33)$$

$$min(g|h|d) = min(g|d \wedge h) / min(h|d \wedge (g\nu\bar{h})). \quad (6.34)$$

В отличие от разложений МЕО.6.17-МЕО.6.24, левая и правая части каждого из разложений МЕО.6.27-МЕО.6.34 представляют не одну функцию, а диапазоны функций, которые являются идентичными. В этом смысле разложения МЕО.6.27-МЕО.6.34 корректны. С точки зрения выбора конкретных функций из диапазонов, операция минимизации левой части дает, в общем случае, функцию, отличную от функции, генерируемой двумя операциями минимизации правой части. Преимущество левой части состоит в том, что совместная минимизация конъюнкции двух булевых функций имеет универсальный характер по сравнению с отдельной минимизацией этих функций. Преимуществом правой части является повышенная степень неопределенности операндов.

Разложения МЕО.6.27-МЕО.6.34 обеспечивают построение иерархических частичных разложений с уменьшением степени определенности операндов по мере увеличения уровня иерархии.

Все предложенные в настоящем параграфе разложения используют операцию минимизации на одном уровне вложенности и не используют суперпозицию операций минимизации, т.е. вхождение одних операций внутрь других. Следующие разложения доказывают, что использование суперпозиции операций минимизации увеличивает силу разложений, дополнительно снижая степень определенности операндов:

$$g \wedge h = \min(g|h) \wedge \min(h|\min(g|h)), \quad (6.35)$$

$$\neg(g \rightarrow h) = \neg(\min(g|\neg h) \rightarrow \min(h|\min(g|\neg h))), \quad (6.36)$$

$$\neg(g \leftarrow h) = \neg(\min(g|h) \leftarrow \min(h|\neg \min(g|h))), \quad (6.37)$$

$$g \vee h = \min(g|\neg h) \vee \min(h|\neg \min(g|\neg h)), \quad (6.38)$$

$$g \downarrow h = \min(g|\neg h) \downarrow \min(h|\neg \min(g|\neg h)), \quad (6.39)$$

$$g \leftarrow h = \min(g|h) \leftarrow \min(h|\neg \min(g|h)), \quad (6.40)$$

$$g \rightarrow h = \min(g|\neg h) \rightarrow \min(h|\min(g|\neg h)), \quad (6.41)$$

$$g|h = \min(g|h) | \min(h|\min(g|h)). \quad (6.42)$$

Каждое из разложений МЕО.6.35-МЕО.6.42 содержит двойное вхождение одной и той же операции минимизации. Для двойного вхождения имеет место согласованный выбор одной результирующей функции из двух диапазонов. Следовательно, для доказательства разложений применим метод граничных точек. Разложение

МЕО.6.35 для булевой операции конъюнкция доказано в примере 6.5. Другие разложения доказываются аналогичным образом.

Разложения МЕО.6.35-МЕО.6.42 не обеспечивают построение иерархических частичных разложений, однако они обеспечивают построение иерархических разложений булевой функции f посредством подходящего выбора функций g и h и применения локальных частичных разложений на каждом уровне иерархии. Одна из методик выбора функций g и h заключается в следующем. Возьмем, к примеру, операцию конъюнкция. С учетом функции стоимости $\chi(g)$ и критерия оптимизации задачи (6.3) выбираем функцию g такую, что $f^n \subseteq g^n \subseteq 1^n$. В качестве функции h выбираем сначала функцию f . С учетом свойств множества конституент 1 функции g заключаем $g \wedge f = f$. Считаем, что функция g минимизирована по f , тогда разложение МЕО.6.35 приобретает вид $f = g \wedge \min(\uparrow g)$. Окончательно, в качестве функции h выбираем $h = \min(\uparrow g)$. Если функцией стоимости $\chi(g)$ является размер формы представления функции g в некотором базисе, то важнейшим дополнительным критерием оптимизации является уменьшение числа существенных переменных в продуктах разложения.

Пример 6.10. Продолжим рассмотрение функции f , описанной в примере 6.9. Для построения разложения функции f по булевой операции конъюнкция применим формулу (6.35). Построим функцию g (рис.6.6), чье множество конституент 1 покрывает множество конституент 1 функции f . Для этого поменяем значение 0 на значение 1 функции f на четырех наборах значений переменных, показанных темной заливкой на карте Карно. Наборы выбраны так, чтобы переменная x_1 стала несущественной в функции g . МДНФ функции g строим по трем кубам на карте Карно:

$$g = x_3 \vee x_2 \wedge x_4 \vee \neg x_2 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(g))=4$. Заметим, что при выборе наборов можно ставить целью сделать несущественной переменную x_2 , x_3 , x_4 или даже пару переменных. Любопытно, что для данной функции f переменные x_2 , x_4 можно сделать несущественными лишь превратив функцию g в константу 1. Построив булеву функцию g , далее строим частично определенную функцию $(\uparrow g)$. На рис.6.6 функция $(\uparrow g)$ представлена картой Карно, четыре клетки которой

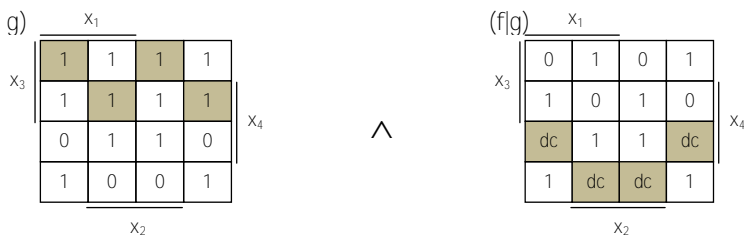


Рисунок 6.6 – Функции g и $(f|g)$, представленные картами Карно

заполнены значением dc , а в остальных клетках значения совпадают со значениями функции f . Учитывая возможность замены dc на 0 или 1, строим функцию $h = \min(f|g)$ в МДНФ на пяти кубах:

$$h = \neg x_3 \vee x_1 \wedge \neg x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge x_4 \vee x_1 \wedge x_2 \wedge \neg x_4 \vee \neg x_1 \wedge \neg x_2 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(h))=12$. Собирая воедино продукты разложения, получаем $\chi(g \wedge h) = 4 + 1 + 12 = 17$. Сравнивая функцию стоимости исходного и результирующего представления функции f , делаем вывод о том, результирующее представление компактнее на две операции. \square

Разложения МЕО.6.35–МЕО.6.42 очевидным образом обобщаются на случай частично определенной функции $(f|d) = (g \bowtie h|d)$:

$$\min(g \wedge h|d) = \min(g|d \wedge h) \wedge \min(h|d \wedge \min(g|d \wedge h)), \quad (6.43)$$

$$\min(\neg(g \rightarrow h)|d) = \neg(\min(g|d \wedge \neg h) \rightarrow \min(h|d \wedge \min(g|d \wedge \neg h))), \quad (6.44)$$

$$\min(\neg(g \leftarrow h)|d) = \neg(\min(g|d \wedge h) \leftarrow \min(h|d \wedge \min(g|d \wedge h))), \quad (6.45)$$

$$\min(g \vee h|d) = \min(g|d \wedge \neg h) \vee \min(h|d \wedge \neg \min(g|d \wedge \neg h)), \quad (6.46)$$

$$\min(g \downarrow h|d) = \min(g|d \wedge \neg h) \downarrow \min(h|d \wedge \neg \min(g|d \wedge \neg h)), \quad (6.47)$$

$$\min(g \leftarrow h|d) = \min(g|d \wedge h) \leftarrow \min(h|d \wedge \neg \min(g|d \wedge h)), \quad (6.48)$$

$$\min(g \rightarrow h|d) = \min(g|d \wedge \neg h) \rightarrow \min(h|d \wedge \min(g|d \wedge \neg h)), \quad (6.49)$$

$$\min(g|h|d) = \min(g|d \wedge h) | \min(h|d \wedge \min(g|d \wedge h)). \quad (6.50)$$

Докажем сформулированные разложения методом исследования граничных точек на примере разложения МЕО.6.46. В определении функции $r = s \vee t$, соответствующей правой части равенства (6.46), используются две операции минимизации $s = \min(g|d \wedge \neg h)$ и $t = \min(h|d \wedge \neg s)$, причем операция $s = \min(g|d \wedge \neg h)$ имеет два вхожде-

ния. Для функции s граничными точками являются $g \wedge d \wedge \neg h$ и $g \vee \neg(d \wedge \neg h)$, а для функции t граничными точками являются $h \wedge d \wedge \neg s$ и $h \vee \neg(d \wedge \neg s)$. Вычислим функцию r на четырех комбинациях граничных точек s и t :

1. $s = g \wedge d \wedge \neg h, t = h \wedge d \wedge \neg s,$
 $r = (g \wedge d \wedge \neg h) \vee h \wedge d \wedge \neg (g \wedge d \wedge \neg h) = (g \vee h) \wedge d.$
2. $s = g \vee \neg(d \wedge \neg h), t = h \wedge d \wedge \neg s,$
 $r = (g \vee \neg(d \wedge \neg h)) \vee h \wedge d \wedge \neg (g \vee \neg(d \wedge \neg h)) = g \vee h \vee \neg d.$
3. $s = g \wedge d \wedge \neg h, t = h \vee \neg(d \wedge \neg s),$
 $r = (g \wedge d \wedge \neg h) \vee h \vee \neg(d \wedge \neg (g \wedge d \wedge \neg h)) = g \vee h \vee \neg d.$
4. $s = g \vee \neg(d \wedge \neg h), t = h \vee \neg(d \wedge \neg s),$
 $r = (g \vee \neg(d \wedge \neg h)) \vee h \vee \neg(d \wedge \neg (g \vee \neg(d \wedge \neg h))) = g \vee h \vee \neg d.$

Таким образом, вычисленные граничные точки свидетельствует о том, что функция r принадлежит диапазону $(g \vee h) \wedge d \leq r \leq g \vee h \vee \neg d$. Результат операции $u = \min(g \vee h | d)$, стоящей в левой части равенства (6.46), лежит в диапазоне $(g \vee h) \wedge d \leq u \leq g \vee h \vee \neg d$. Диапазоны для r и u идентичны, что и доказывает корректность разложения МЕО.6.46.

Разложения МЕО.6.27-МЕО.6.34 являются базой для построения иерархических частичных разложений, уменьшающих степень определенности операндов по мере увеличения уровня иерархии. Реализация разложений МЕО.6.17-МЕО.6.50 может быть выполнена для любых форм представления функций g, h, d .

6.5. *Оптимальное разложение поведения системы по характеристической функции*

Разрабатываемые в этом параграфе разложения будем называть МЕФ от английского Minimizing Expansion on Function.

Пусть дана полностью определенная булева функция f , структура которой априори не известна. Как структурировать функцию f посредством построения разложений, использующих операцию минимизации совместно с булевыми операциями? Введем в рассмотрение произвольную булеву функцию c , определенную на том же множестве булевых переменных, на каком определена функция f . Частичные разложения РЕФ.4.29-РЕФ.4.36 позволяют представить

функцию ($f|1$) через функцию c и частичные двуместные операции с уменьшенной степенью определенности операндов. Операция минимизации выгодно использует уменьшенную степень определенности операндов посредством следующих разложений:

$$f = \min(f \vee c) \wedge \min(f \vee \neg c). \quad (6.51)$$

$$f = \neg(\min(f \vee c) \rightarrow \min(\neg f \vee \neg c)), \quad (6.52)$$

$$f = \neg(\min(\neg f \vee c) \leftarrow \min(f \vee \neg c)), \quad (6.53)$$

$$f = \min(f \neg \vee c) \vee \min(f \neg \vee \neg c), \quad (6.54)$$

$$f = \min(\neg f \vee c) \downarrow \min(\neg f \vee \neg c), \quad (6.55)$$

$$f = \min(f \neg \vee c) \leftarrow \min(\neg f \neg \vee \neg c), \quad (6.56)$$

$$f = \min(\neg f \neg \vee c) \rightarrow \min(f \neg \vee \neg c), \quad (6.57)$$

$$f = \min(\neg f \neg \vee c) / \min(\neg f \neg \vee \neg c). \quad (6.58)$$

Докажем разложения MEF.6.51-MEF.6.58 на примере разложения MEF.6.57. Результат операции $u = \min(\neg f \neg \vee c)$ принадлежит диапазону $\neg f \wedge (\neg \vee c) \leq u \leq \neg \vee \neg (\neg \vee c)$, преобразуемому к диапазону $\neg f \leq u \leq \neg \vee \neg c$. Результат операции $w = \min(f \neg \vee \neg c)$ принадлежит диапазону $f \wedge (\neg \vee \neg c) \leq w \leq \vee \neg (\neg \vee \neg c)$, преобразуемому к диапазону $f \wedge \neg c \leq w \leq f$. Правую часть равенства MEF.6.57 запишем в виде $r = u \rightarrow w = \neg u \vee w$. Диапазон для $\neg u$ есть инвертированный диапазон для u : $f \wedge c \leq \neg u \leq f$. Выполнение операции дизъюнкции над диапазонами для $\neg u$ и w дает диапазон $(f \wedge c) \vee (f \wedge \neg c) \leq r \leq (f) \vee (f)$ или итоговый диапазон $f \leq r \leq f$. Он включает одну функцию f , что и доказывает разложение MEF.6.57.

Пример 6.11. Продолжим рассмотрение булевой функции f , описанной в примере 6.10 и представленной картой Карно на рис.6.3. Выполним разложение функции f по булевой операции конъюнкции в соответствии с соотношением MEF.6.51. Построим функцию c (рис.6.7), чье множество конституент 1 не пересекается с множеством конституент 1 функции f . Для этого, значения клеток левой части карты Карно, показанной темной заливкой, примем инверсными значениям функции f , а значения клеток правой части карты примем равными 0. Значения функции c выбраны так, чтобы переменная x_3 стала несущественной в одном из продуктов разложения. Имея функцию c , строим продукты разложения MEF.6.51, показанные на рис.6.8. Первым продуктом является частично

с)

| | | | | | |
|-------|-------|---|---|---|-------|
| | x_1 | | | | |
| | 1 | 0 | 0 | 0 | |
| x_3 | 0 | 1 | 0 | 0 | x_4 |
| | 1 | 0 | 0 | 0 | |
| | 0 | 1 | 0 | 0 | |
| | 0 | 1 | 0 | 0 | |
| | x_2 | | | | |

Рисунок 6.7 – Булева функция c

| | | | | | |
|--------------|-------|---|----|----|-------|
| $(f \vee c)$ | x_1 | | | | |
| | 0 | 1 | dc | 1 | |
| x_3 | 1 | 0 | 1 | dc | x_4 |
| | 0 | 1 | 1 | dc | |
| | 1 | 0 | dc | 1 | |
| | 1 | 0 | dc | 1 | |
| | x_2 | | | | |

∧

| | | | | | |
|-------------------|-------|----|---|---|-------|
| $(f \vee \neg c)$ | x_1 | | | | |
| | dc | 1 | 0 | 1 | |
| x_3 | 1 | dc | 1 | 0 | x_4 |
| | dc | 1 | 1 | 0 | |
| | 1 | dc | 0 | 1 | |
| | 1 | dc | 0 | 1 | |
| | x_2 | | | | |

Рисунок 6.8 – Частично определенные функции $(f \vee c)$ и $(f \vee \neg c)$

определенная функция $(f \vee c)$. Значения клеток левой части карты Карно, представляющей $(f \vee c)$, совпадают со значениями функции f . Четыре клетки правой части карты получили значения dc . Минимизируем функцию $g = \min(f \vee c)$ в классе МДНФ с учетом того, что значение dc может быть произвольным образом заменено на 0 или 1. В результате, МДНФ функции g строится по пяти кубам карты Карно:

$$g = \neg x_1 \vee x_2 \wedge x_3 \wedge \neg x_4 \vee x_2 \wedge \neg x_3 \wedge x_4 \vee \neg x_2 \wedge x_3 \wedge x_4 \vee \neg x_2 \wedge \neg x_3 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(g)) = 12$.

Вторым продуктом разложения является частично определенная функция $(f \vee \neg c)$. Значения клеток правой части карты Карно, представляющей $(f \vee \neg c)$, совпадают со значениями функции f . Четыре клетки левой части получают значение dc . Минимизируем функцию $h = \min(f \vee \neg c)$ в классе МДНФ на трех кубах карты Карно:

$$h = x_1 \vee x_2 \wedge x_4 \vee \neg x_2 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(h))=4$. Собирая воедино продукты разложения, получаем $\chi(g \wedge h)=12+1+4=17$. Результирующее представление функции f компактнее на две операции по сравнению с исходным представлением в МДНФ. □

С учетом разложений PEF.4.37-PEF.4.44 разложения MEF.6.51-MEF.6.58 легко обобщаются на случай частично определенной функции ($f|d$):

$$\min(f|d) = \min(f|d \wedge (\bar{f} \vee c)) \wedge \min(f|d \wedge (\bar{f} \vee \bar{c})). \quad (6.59)$$

$$\min(f|d) = \neg(\min(f|d \wedge (\bar{f} \vee c)) \rightarrow \min(\neg f|d \wedge (\bar{f} \vee \bar{c}))), \quad (6.60)$$

$$\min(f|d) = \neg(\min(\neg f|d \wedge (\bar{f} \vee c)) \leftarrow \min(f|d \wedge (\bar{f} \vee \bar{c}))), \quad (6.61)$$

$$\min(f|d) = \min(f|d \wedge (\neg \bar{f} \vee c)) \vee \min(f|d \wedge (\neg \bar{f} \vee \bar{c})), \quad (6.62)$$

$$\min(f|d) = \min(\neg f|d \wedge (\bar{f} \vee c)) \downarrow \min(\neg f|d \wedge (\bar{f} \vee \bar{c})), \quad (6.63)$$

$$\min(f|d) = \min(f|d \wedge (\neg \bar{f} \vee c)) \leftarrow \min(\neg f|d \wedge (\neg \bar{f} \vee \bar{c})), \quad (6.64)$$

$$\min(f|d) = \min(\neg f|d \wedge (\neg \bar{f} \vee c)) \rightarrow \min(f|d \wedge (\neg \bar{f} \vee \bar{c})), \quad (6.65)$$

$$\min(f|d) = \min(\neg f|d \wedge (\neg \bar{f} \vee c)) / \min(\neg f|d \wedge (\neg \bar{f} \vee \bar{c})). \quad (6.66)$$

Разложения MEF.6.59-MEF.6.66 доказываются методом диапазонов. Эффективность разложений MEF.6.51-MEF.6.66 в сильной степени зависит от выбора функции c . Крайние значения $c=0$ и $c=1$ вовсе не снижают уровень определенности операндов, так как в этом случае операнды редуцируются к полностью определенным функциям 0 , 1 , f и $\neg f$ в разложениях MEF.6.51-MEF.6.58 и редуцируются к операции минимизации частично определенных функций $\min(0|d)$, $\min(1|d)$, $\min(f|d)$, $\min(\neg f|d)$ в разложениях MEF.6.59-MEF.6.66. Для всех разложений наблюдается одна закономерность: чем выше степень неопределенности одного операнда, тем ниже степень неопределенности другого операнда. Значит, искусство минимизации заключается в выборе «золотой середины» для функции c .

6.6. Оптимальное разложение поведения системы по двум булевым операциям

Разрабатываемые в этом параграфе разложения будем называть ME2F от английского Minimizing Expansion on Two Operations and a Function.

Пусть, как и прежде, f – заданная исходная булева функция, и c – произвольно выбираемая нами булева функция, определенная на том же множестве булевых переменных, что и f . Использование операции минимизации позволяет преобразовать частичное разложение PE2F.4.48, построенное на базе двух двухместных частичных операций дизъюнкции и конъюнкции, в разложение, построенное на базе двух двухместных булевых операций дизъюнкции и конъюнкции:

$$f = c \wedge \min(f|c) \vee \neg c \wedge \min(f|\neg c). \quad (6.67)$$

В силу принципиальной важности разложения, докажем его методом диапазонов. Результат операции $u = \min(f|c)$ принадлежит диапазону $f \wedge c \leq u \leq f \vee \neg c$, а результат операции $w = \min(f|\neg c)$ принадлежит диапазону $f \wedge \neg c \leq w \leq f \vee c$. Конъюнкция $s = c \wedge u$ дает диапазон $f \wedge c \leq s \leq f \wedge c$, а конъюнкция $t = \neg c \wedge w$ дает диапазон $f \wedge \neg c \leq t \leq f \wedge \neg c$. Применяя дизъюнкцию $r = s \vee t$, получаем диапазон $(f \wedge c) \vee (f \wedge \neg c) \leq r \leq (f \wedge c) \vee (f \wedge \neg c)$, преобразуемый к диапазону $f \leq r \leq f$. Результирующий диапазон для r состоит из одной функции f , что и доказывает разложение ME2F.6.67.

При $c = x_i$ разложение ME2F.6.67 становится разложением Шеннона (1.1). Благодаря этому мы можем сказать, что разложение ME2F.6.67 обобщает разложение Шеннона. Для частично определенной функции $(f|d)$ разложение ME2F.6.67 трансформируется к виду

$$\min(f|d) = c \wedge \min(f|d \wedge c) \vee \neg c \wedge \min(f|d \wedge \neg c). \quad (6.68)$$

Здесь левая и правая части равенства представляют не одну конкретную функцию как в ME2F.6.67, но обеспечивают выбор одной из функций в диапазоне $f \wedge d \leq r \leq f \vee \neg d$.

В разложении ME2F.6.68 операнд c является полностью определенным. Следующее разложение позволяет уменьшить область определенности операнда c :

$$\begin{aligned} \min(f|d) = & \min(c|d) \wedge \min(f|d \wedge \min(c|d)) \vee \\ & \neg \min(c|d) \wedge \min(f|d \wedge \neg \min(c|d)). \end{aligned} \quad (6.69)$$

Замена функции c на функцию $\min(c|d)$ является правомерной в разложении ME2F.6.68 постольку, поскольку c является произ-

вольной булевой функцией. В разложениях ME2F.6.67, ME2F.6.68 операции дизъюнкции и конъюнкции могут меняться местами. Это приводит к построению разложений

$$f = (c \vee \min(f \neg c)) \wedge (\neg c \vee \min(f c)), \quad (6.70)$$

$$\min(f d) = (c \vee \min(f d \wedge \neg c)) \wedge (\neg c \vee \min(f d \wedge c)). \quad (6.71)$$

Вместо операций дизъюнкции и конъюнкции может дважды применяться операция штрих Шеффера и дважды применяться операция стрелка Пирса:

$$f = (d \min(f c)) / (\neg c / \min(f \neg c)), \quad (6.72)$$

$$f = (c \downarrow \min(f c)) \downarrow (\neg c \downarrow \min(f \neg c)), \quad (6.73)$$

$$\min(f d) = (c / \min(f d \wedge c)) / (\neg c / \min(f d \wedge \neg c)). \quad (6.74)$$

$$\min(f d) = (c \downarrow \min(f d \wedge c)) \downarrow (\neg c \downarrow \min(f d \wedge \neg c)). \quad (6.75)$$

Пример 6.12. Снова рассмотрим булеву функцию f , описанную в примерах 6.10 и 6.11 и представленную картой Карно на рис.6.3. Выполним разложение функции f по двум двухместным булевым операциям в соответствии с соотношением ME2F.6.67. В качестве функции c выберем отдельную переменную: $c = x_2$. Функция представлена картой Карно на рис. 6.9. Имея функцию c , строим продукты разложения ME2F.6.67, показанные на рис.6.10. Первым продуктом является частично определенная функция $(f c)$. Значения клеток двух средних столбцов карты Карно, представляющей $(f c)$, совпадают со значениями функции f . Клетки левого крайнего и правого крайнего столбцов карты получили значения dc .

Минимизируем функцию $g = \min(f c)$ в классе МДНФ с учетом того, что значение dc может быть произвольным образом заменено на 0 или 1. В результате, МДНФ функции g строится по трем кубам карты Карно:

$$g = \neg x_1 \wedge x_4 \vee \neg x_3 \wedge x_4 \vee x_1 \wedge x_3 \wedge \neg x_4.$$

Функция стоимости $\chi(\text{МДНФ}(g))=6$. Вторым продуктом разложения является частично определенная функция $(f \neg c)$. Значения клеток левого и правого столбцов карты Карно, представляющей $(f \neg c)$, совпадают со значениями функции f . Клетки средних столбцов получают значение dc . Минимизируем функцию $h = \min(f \neg c)$ в

$$C=X_2$$

| | | | | | |
|-------|---|-------|---|---|---|
| | | x_1 | | | |
| | | 0 | 1 | 1 | 0 |
| x_3 | 0 | 0 | 1 | 1 | 0 |
| | 1 | 0 | 1 | 1 | 0 |
| | 2 | 0 | 1 | 1 | 0 |
| | 3 | 0 | 1 | 1 | 0 |
| | | x_2 | | | |

Рисунок 6.9 – Булева функция $C=X_2$

| | | | | | |
|---------|---|-------|---|---|----|
| $(f C)$ | | x_1 | | | |
| | | dc | 1 | 0 | dc |
| x_3 | 0 | dc | 0 | 1 | dc |
| | 1 | dc | 1 | 1 | dc |
| | 2 | dc | 1 | 1 | dc |
| | 3 | dc | 0 | 0 | dc |
| | | x_2 | | | |

| | | | | | |
|--------------|---|-------|----|----|---|
| $(f \neg C)$ | | x_1 | | | |
| | | 0 | dc | dc | 1 |
| x_3 | 0 | 1 | dc | dc | 0 |
| | 1 | 0 | dc | dc | 0 |
| | 2 | 1 | dc | dc | 1 |
| | 3 | 0 | dc | dc | 1 |
| | | x_2 | | | |

Рисунок 6.10 – Частично определенные функции $(f|C)$ и $(f|\neg C)$ для $C=X_2$

классе МДНФ по трем кубам карты Карно:

$$h = x_1 \wedge \neg x_4 \vee \neg x_3 \wedge \neg x_4 \vee x_1 \wedge x_3 \wedge x_4.$$

Функция стоимости $\chi(\text{МДНФ}(h))=6$. Собирая воедино продукты разложения, получаем функцию

$$f = x_2 \wedge (\neg x_1 \wedge x_4 \vee \neg x_3 \wedge x_4 \vee x_1 \wedge x_3 \wedge \neg x_4) \vee \neg x_2 \wedge (x_1 \wedge \neg x_4 \vee \neg x_3 \wedge \neg x_4 \vee x_1 \wedge x_3 \wedge x_4),$$

для которой функция сложности $\chi(\hat{f})=6+3+6=15$. Результирующее представление функции f компактнее на четыре операции по сравнению с исходным представлением в МДНФ. □

Пример 6.13. Он является продолжением примера 6.12. Снова применим к функции f разложение ME2F.6.67, но в качестве C выберем функцию $C=X_2 \oplus X_4$. Функция представлена картой Карно на рис. 6.11. Продукты разложения ME2F.6.67 для новой функции C

$$C = X_2 \oplus X_4$$

| | | | | | |
|-------|-------|---|---|---|-------|
| | X_1 | | | | |
| | 0 | 1 | 1 | 0 | |
| X_3 | 1 | 0 | 0 | 1 | X_4 |
| | 1 | 0 | 0 | 1 | |
| | 0 | 1 | 1 | 0 | |
| | X_2 | | | | |

Рисунок 6.11 – Булева функция $C = X_2 \oplus X_4$

$(f|C)$

| | | | | | |
|-------|-------|----|----|----|-------|
| | X_1 | | | | |
| | dc | 1 | 0 | dc | |
| X_3 | 1 | dc | dc | 0 | X_4 |
| | 0 | dc | dc | 0 | |
| | dc | 0 | 0 | dc | |
| | X_2 | | | | |

$(f|\neg C)$

| | | | | | |
|-------|-------|----|----|----|-------|
| | X_1 | | | | |
| | 0 | dc | dc | 1 | |
| X_3 | dc | 0 | 1 | dc | X_4 |
| | dc | 1 | 1 | dc | |
| | 1 | dc | dc | 1 | |
| | X_2 | | | | |

Рисунок 6.12 – Частично определенные функции $(f|C)$ и $(f|\neg C)$ для $C = X_2 \oplus X_4$

показаны на рис.6.12. В первом продукте $(f|C)$ значения четырех угловых и четырех центральных клеток карты Карно получили значение dc , значения остальных клеток совпадают со значениями функции f . Минимизируем функцию $g = \min(f|C)$ в классе МДНФ на одном кубе:

$$g = X_1 \wedge X_3.$$

Функция стоимости $\chi(\text{МДНФ}(g)) = 1$. Во втором продукте $(f|\neg C)$ значения четырех угловых и четырех центральных клеток карты Карно совпадают со значениями функции f , остальные клетки получают значение dc . Минимизируем функцию $h = \min(f|\neg C)$ в классе МДНФ на двух кубах:

$$h = \neg X_1 \vee \neg X_3.$$

Функция стоимости $\chi(\text{МДНФ}(h))=1$. Собирая воедино продукты разложения, получаем систему из двух функций:

$$\begin{aligned} c &= x_2 \oplus x_4, \\ f &= c \wedge (x_1 \wedge x_3) \vee \neg c \wedge (\neg x_1 \vee \neg x_3), \end{aligned}$$

для которых функция стоимости $\chi(\hat{f})=1+1+3+1=6$ в числе операций \vee, \wedge, \oplus и $\chi(\hat{f})=8$ в числе операций \vee, \wedge , поскольку функция стоимости операции \oplus равна 3 при представлении ее операциями \vee, \wedge . Таким образом, полученное представление функции f компактнее, по крайней мере, на тринадцать операций по сравнению с исходным представлением в МДНФ. Данный пример показывает также, насколько важным в разложении ME2F.6.67 является выбор функции c . Так, выбор $c=x_2 \oplus x_4$ вместо $c=x_2$ позволил сократить функцию сложности $\chi(\hat{f})$ с 15 до 6, т.е. более чем в два раза. \square

Для разложений ME2F.6.67-ME2F.6.75 также как и для ранее рассмотренных разложений действует принцип сообщающихся сосудов при выборе функции c . Выбор «золотой середины» для функции c является эффективной стратегией оптимизации и в этом случае.

6.7. Оптимальное разложение поведения системы по операциям исключающее или и эквивалентность

Разрабатываемые в этом параграфе разложения будем называть ME2F, как и в предыдущем параграфе.

В параграфе 4.4 изучена возможность обобщения законов Давио в смешанной алгебре. На базе частичных операций исключающее или, конъюнкция и инверсия не представляется возможным построить в смешанной алгебре разложения, аналогичные разложениям Давио, построенным в булевой алгебре. Введение в разложения PE2F.4.64, PE2F.4.65 операции минимизации меняет ситуацию и делает разложения корректными для любой функции c :

$$f = \min(f \neg c) \oplus c \wedge (\min(f c) \oplus \min(f \neg c)), \quad (6.76)$$

$$f = \min(f c) \oplus \neg c \wedge (\min(f c) \oplus \min(f \neg c)), \quad (6.77)$$

Поскольку операция минимизации $s = \min(f \neg c)$ входит дважды в разложение ME2F.6.76, а операция минимизации $t = \min(f c)$ входит

дважды в разложение ME2F.6.77, эти разложения доказываются методом граничных точек. Докажем разложение ME2F.6.76. Для функции s граничными точками являются $f_{\wedge \neg c}$ и $f_{\vee c}$, а для функции t граничными точками являются $f_{\wedge c}$ и $f_{\vee \neg c}$. Вычислим функцию $r = s \oplus c \wedge (t \oplus s)$ на четырех комбинациях граничных точек s и t :

1. $s = f_{\wedge \neg c}, t = f_{\wedge c},$
 $r = f_{\wedge \neg c} \oplus c \wedge (f_{\wedge c} \oplus f_{\wedge \neg c}) = f.$
2. $s = f_{\vee c}, t = f_{\wedge c},$
 $r = f_{\vee c} \oplus c \wedge (f_{\wedge c} \oplus f_{\vee c}) = f.$
3. $s = f_{\wedge \neg c}, t = f_{\vee \neg c},$
 $r = f_{\wedge \neg c} \oplus c \wedge (f_{\vee \neg c} \oplus f_{\wedge \neg c}) = f.$
4. $s = f_{\vee c}, t = f_{\vee \neg c},$
 $r = f_{\vee c} \oplus c \wedge (f_{\vee \neg c} \oplus f_{\vee c}) = f.$

Таким образом, вычисленные граничные точки свидетельствует о том, что функция r принадлежит диапазону, включающему одну функцию f . Равенство ME2F.6.76 доказано. Мы говорим, что оно обобщает положительное разложение Давио по той причине, что замена в нем функции c на переменную x_i дает известное в литературе положительное разложение Давио булевой функции f по переменной x_i . Аналогично доказывается равенство ME2F.6.77. Его частным случаем является отрицательное разложение Давио по переменной x_i .

Пример 6.14. Он является продолжением примера 6.12, в котором для заданной функции f выбрана функция $c = x_2$. Булевы функции $\min(f|_c)$ и $\min(f|_{\neg c})$, полученные в примере 6.12 в алгебраической форме, представлены картами Карно на рис.6.13. Выполняя операцию исключающее или, получаем $\min(f|_c) \oplus \min(f|_{\neg c}) = 1$ (рис.6.13). Тогда представление f на основе положительного разложения Давио:

$$f = \min(f|_{\neg c}) \oplus c \wedge 1 = (x_1 \wedge \neg x_4 \vee \neg x_3 \wedge \neg x_4 \vee x_1 \wedge x_3 \wedge x_4) \oplus x_2.$$

Функция стоимости $\chi(\wedge) = 7$ в числе операций \vee, \wedge, \oplus и $\chi(\wedge) = 9$ в числе операций \vee, \wedge . Представление f на основе отрицательного разложения Давио:

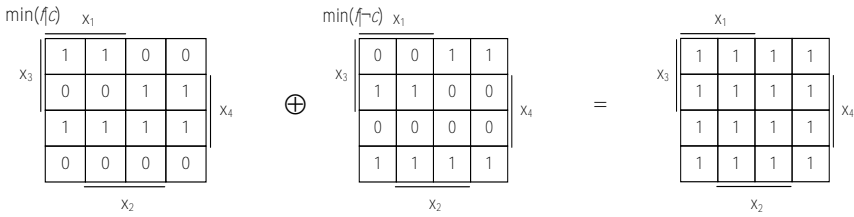


Рисунок 6.13 – Выполнение \oplus над функциями $\min(f|c)$ и $\min(f|\neg c)$ при $c=x_2$

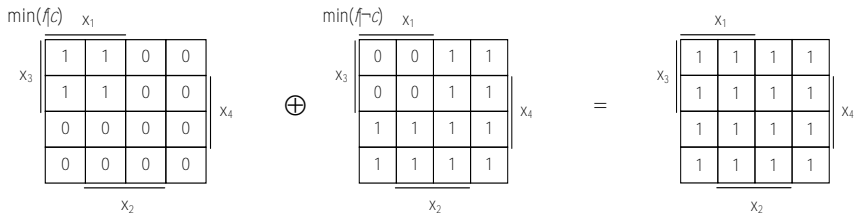


Рисунок 6.14 – Выполнение \oplus над функциями $\min(f|c)$ и $\min(f|\neg c)$ при $c=x_2 \oplus x_4$

$$f = \min(f|c) \oplus \neg c \wedge 1 = (\neg x_1 \wedge x_4 \vee \neg x_3 \wedge x_4 \vee x_1 \wedge x_3 \wedge \neg x_4) \oplus \neg x_2,$$

Функция стоимости $\chi(f)=7$ в числе операций \vee , \wedge , \oplus и $\chi(f)=9$ в числе операций \vee , \wedge . \square

Пример 6.15. Он является продолжением примеров 6.13, 6.14. Выберем теперь $c=x_2 \oplus x_4$. Булевы функции $\min(f|c)$ и $\min(f|\neg c)$, полученные в примере 6.13 в алгебраической форме, представлены картами Карно на рис.6.14. Выполняя операцию исключающее или, получаем $\min(f|c) \oplus \min(f|\neg c)=1$ (рис.6.14). Тогда представление f на основе положительного разложения Давио:

$$f = \min(f|\neg c) \oplus c \wedge 1 = (\neg x_1 \vee \neg x_3) \oplus (x_2 \oplus x_4),$$

Функция стоимости $\chi(f)=3$ в числе операций \vee , \wedge , \oplus и $\chi(f)=7$ в числе операций \vee , \wedge . Представление f на основе отрицательного разложения Давио:

$$f = \min(f|c) \oplus \neg c \wedge 1 = (x_1 \wedge x_3) \oplus \neg (x_2 \oplus x_4),$$

Таблица 6.1

Сравнение различных типов разложений булевой функции f

| № | Тип разложения | Операции \vee, \wedge, \oplus | | Операции \vee, \wedge | |
|---|------------------------------|---------------------------------|-----|-------------------------|-----|
| | | $\chi(f)$ | % | $\chi(f)$ | % |
| 1 | МДНФ | 19 | 100 | 19 | 100 |
| 2 | МЕО.6.17 | 17 | 89 | 17 | 89 |
| 3 | МЕО.6.35 | 17 | 89 | 17 | 89 |
| 4 | МЕФ.6.51 | 17 | 89 | 17 | 89 |
| 5 | Шеннон 1.1 $C=X_2$ | 15 | 79 | 15 | 79 |
| 6 | МЕ2Ф.6.67 $C=X_2 \oplus X_4$ | 6 | 32 | 8 | 42 |
| 7 | Давио (1.2) $C=X_2$ | 7 | 37 | 9 | 47 |
| 8 | МЕ2Ф.6.76 $C=X_2 \oplus X_4$ | 3 | 16 | 7 | 37 |

Снова функция стоимости $\chi(f)=3$ в числе операций \vee, \wedge, \oplus и $\chi(f)=7$ в числе операций \vee, \wedge .

Таким образом, выбор $C=X_2 \oplus X_4$ вместо $C=X_2$ позволил сократить $\chi(f)$ с 7 до 3 в числе операций \vee, \wedge, \oplus и сократить $\chi(f)$ с 9 до 7 в числе операций \vee, \wedge .

Собирая результаты, полученные в примерах 6.9-6.15, строим табл.6.1. Даже такой простой пример убедительно доказывает, что предложенные разложения, построенные на законах смешанной алгебры, позволяют существенно (на 84%) улучшать параметры представлений булевых функций и их реализаций. Интересен тот факт, что известные разложения Шеннона и Давио заметно уступают разложениям, построенным по законам смешанной алгебры. \square

Изучим другие законы с участием операций исключающее или и эквивалентность.

Посредством замены операции дизъюнкция операцией исключающее или в разложении (6.67) получаем разложение:

$$f = c \wedge \min(f, c) \oplus \neg c \wedge \min(f, \neg c). \quad (6.78)$$

Докажем разложение МЕ2Ф.6.78 методом граничных точек. Для функции $s = \min(f, c)$ граничными точками являются $f \wedge c$ и $f \vee \neg c$, а для функции $t = \min(f, \neg c)$ граничными точками являются $f \wedge \neg c$ и $f \vee c$. Вычислим функцию $r = c \wedge s \oplus \neg c \wedge t$ на четырех комбинациях граничных точек s и t :

2. $s = f \wedge c, t = f \wedge \neg c,$
 $r = c \wedge f \wedge c \oplus \neg c \wedge f \wedge \neg c = f.$
3. $s = f \vee \neg c, t = f \wedge \neg c,$
 $r = c \wedge (f \vee \neg c) \oplus \neg c \wedge f \wedge \neg c = f.$
4. $s = f \wedge c, t = f \vee c,$
 $r = c \wedge f \wedge c \oplus \neg c \wedge (f \vee c) = f.$
5. $s = f \vee \neg c, t = f \vee c,$
 $r = c \wedge (f \vee \neg c) \oplus \neg c \wedge (f \vee c) = f.$

Вычисленные граничные точки показывают, что диапазон функций r включает одну функцию f . Разложение ME2F.6.78 доказано.

Перепишем законы MEO.6.25, MEO.6.26 с учетом замены g на f и замены h на c .

$$\begin{aligned} f \oplus c &= \min(f \neg c) \wedge \min(\neg c f) \vee \min(\neg f c) \wedge \min(c \neg f), \\ f \leftrightarrow c &= \min(f c) \wedge \min(c f) \wedge \min(\neg f \neg c) \wedge \min(\neg c \neg f). \end{aligned}$$

Если принять $\min(c \neg f) = c$ и $\min(\neg c f) = \neg c$, то разложение MEO.6.25 сводится к разложению

$$f \oplus c = c \wedge \min(\neg f c) \vee \neg c \wedge \min(f \neg c),$$

которое с учетом закона (6.4) для булевой операции инвертирования может быть записано в виде:

$$f \oplus c = c \wedge \neg \min(f c) \vee \neg c \wedge \min(f \neg c). \quad (6.79)$$

Если принять $\min(c f) = c$ и $\min(\neg c \neg f) = \neg c$, то разложение MEO.6.26 сводится к разложению

$$f \leftrightarrow c = c \wedge \min(f c) \vee \neg c \wedge \min(\neg f \neg c),$$

которое с учетом закона (6.4) для булевой операции инвертирования может быть записано в виде:

$$f \leftrightarrow c = c \wedge \min(f c) \vee \neg c \wedge \neg \min(f \neg c). \quad (6.80)$$

Любопытен тот факт, что соотношения ME2F.6.79, ME2F.6.80 являющиеся преобразованиями функции f с помощью функции c и операций исключаящее или и эквивалентность, очень близки по своей структуре разложению ME2F.6.67.

Для частично определенной функции (fd) разложение (6.78) трансформируется к виду

$$\min(fd) = c \wedge \min(fd \wedge c) \oplus \neg c \wedge \min(fd \wedge \neg c), \quad (6.81)$$

а разложения ME2F.6.76, ME2F.6.77 трансформируются к виду

$$\min(fd) = \min(fd \wedge \neg c) \oplus c \wedge f_{\oplus}, \quad (6.82)$$

$$\min(fd) = \min(fd \wedge c) \oplus \neg c \wedge f_{\oplus}. \quad (6.83)$$

где

$$f_{\oplus} = \min(fd \wedge c) \oplus \min(fd \wedge \neg c). \quad (6.84)$$

Функция f_{\oplus} вычисляется выполнением операции исключающее или после выполнения двух операций минимизации. Она показывает разницу между функциями $u = \min(fd \wedge c)$ и $w = \min(fd \wedge \neg c)$. Функции U и W называются *ортогональными*, если на любом наборе a значений переменных функция U принимает значение 1 тогда и только тогда, когда функция W принимает на этом наборе значение 0. Если функции U и W являются ортогональными, то $f_{\oplus} = 1$.

6.8. Оптимизация иерархических разложений логических проектов

Задача оптимизации (6.3) частично определенной булевой функции (fd) сформулирована в расчете на представление функции в VCR форме. Как следствие, допустимое множество задачи (6.3) описывается диапазоном булевых функций (6.2). При разработке разложений функций МЕО, МЕФ, МЕ2F допустимое множество расширено до произвольного вычисляемого непрерывного диапазона булевых функций. Вновь расширим формулировку задачи оптимизации (6.3), но теперь в расчете на иерархическое представление и построение функций. Если функция представлена формулой или выражением в смешанной алгебре, то графически это выражение можно представить деревом разбора. Нетерминальным вершинам дерева соответствуют чередующиеся частично определенные функции и операции над функциями, терминальным вершинам соответствуют константы и переменные, представляющие функции значения в VCR формах.

Известно, что одну частично определенную функцию можно представить большим множеством разнообразных выражений. Если собрать воедино деревья всех этих выражений, то получим *дерево поиска* оптимального представления функции. Нетерминальными вершинами дерева поиска являются чередующиеся И и ИЛИ вершины. Вершинам И соответствуют операции, а исходящие дуги направлены в функции, являющиеся операндами этих операций. Число исходящих дуг определяется местностью операции. Вершинам ИЛИ соответствуют частично определенные функции, а исходящие дуги соответствуют различным разложениям этих функций. Число исходящих дуг определяется числом альтернативных разложений. Меткой вершины И является значение функции стоимости операции $\chi(o)$. Метками вершины ИЛИ являются функция значения ν , характеристическая функция области определенности d , функция стоимости $\chi(\nu)$. Дуги, исходящие из вершины И и входящие в вершину ИЛИ, не метятся. Дуги, исходящие из вершины ИЛИ и входящие в вершину И, метятся именами разложений i , возможно, компонентами разложений, каковым, например, является функция c . Меткой этих дуг может быть также функция стоимости $\chi(c)$. Терминальным вершинам (листьям) дерева поиска соответствуют функции $(\nu|d)$, в которых функция ν представлена константой 0 или 1, литералом x_i и $\neg x_i$. Чтобы получить такие листья, необходим подходящий выбор порядка применения разложений. Для того, чтобы из дерева поиска получить конкретное дерево выражения, представляющего заданную функцию, необходимо выбрать в дереве поиска всех потомков каждой вершины И и выбрать ровно одного потомка каждой вершины ИЛИ.

Число ветвлений в вершине ИЛИ дерева поиска, которой соответствует функция f , определяется двумя факторами:

1. разнообразием разложений функции f и разнообразием двухместных операций, по которым строятся разложения;
2. разнообразием компонентов (функций c , g и h), по которым строятся разложения функции f .

Если число двухместных операций ограничено десятью, число типов разложений ограничено десятками, то число вариантов функций c , g и h является огромным для каждой вершины дерева поиска. Следовательно, дерево поиска имеет чрезвычайно большие

размеры. Явный обход такого дерева поиска методом перебора просто невозможен. Необходима разработка эффективных динамических методов оптимизации, ограничивающих перебор вариантов за счет выбора предпочтительных направлений обхода.

Ограничимся далее при построении дерева T тремя разложениями: ME2F.6.68, ME2F.6.78 и ME2F.6.79. Исследования показывают, эти разложения дают лучшие результаты и являются технологически предпочтительными. В таком дереве вершине ИЛИ соответствуют частично определенная функция (ψd) и функция стоимости $\chi(\psi)$. Вершине И соответствует одна из трех трехместных операций: \mathfrak{R} , \mathfrak{S} , \mathfrak{T} . Операция \mathfrak{R} представляет разложение ME2F.6.68, являющееся суперпозицией операций \vee , \wedge . Операции \mathfrak{S} и \mathfrak{T} представляют разложения ME2F.6.78 и ME2F.6.79, являющиеся суперпозициями операций \oplus , \wedge . Каждая вершина И имеет три исходящие дуги и три дочерние вершины, причем дочерними вершинами могут быть ИЛИ вершины и листья. Дочерними вершинами вершины \mathfrak{R} являются c , $\min(\psi d \wedge c)$ и $\min(\psi d \wedge \neg c)$. Дочерними вершинами вершины \mathfrak{S} являются c , $\min(\psi d \wedge \neg c)$ и f_{\oplus} . Дочерними вершинами \mathfrak{T} вершины являются c , $\min(\psi d \wedge c)$ и f_{\oplus} .

Возможно несколько процедур расчета минимального значения функции стоимости $\chi(T)$ для всего дерева поиска T . Каждая процедура обусловлена параметром, ассоциируемым с функцией стоимости. Если дерево T построено для заданной частично определенной функции $(f d)$ и функция стоимости $\chi(T)$ описывает размер представления булевой функции f в числе двухместных операций, то процедура заключается в следующем.

Функцию стоимости $\chi(t)=\chi(\psi)$ ассоциируем с каждой вершиной t дерева T . Если t – терминальная вершина, то $\chi(t)=0$. Если t – вершина И, и вершины t_1 , t_2 , t_3 являются дочерними для t , то $\chi(t)=\chi(d)+\chi(t_1)+\chi(t_2)+\chi(t_3)$, где $\chi(d)$ – стоимость операции \mathfrak{R} , \mathfrak{S} или \mathfrak{T} . Значение функции стоимости $\chi(\mathfrak{R})=3$ в числе операций \vee , \wedge . Значение функции стоимости $\chi(\mathfrak{S})=2$ в числе операций \oplus , \wedge и $\chi(\mathfrak{S})=4$ в числе операций \vee , \wedge . Значение функции стоимости $\chi(\mathfrak{T})=2$ в числе операций \oplus , \wedge и $\chi(\mathfrak{T})=4$ в числе операций \vee , \wedge . Если t – вершина ИЛИ и вершины t_1, \dots, t_k являются дочерними для t , то $\chi(t)=\min(\chi(t_1)+\dots+\chi(t_k))$. Заметим, что описанная процедура не учитывает

многократное появление одинаковых подфункций в дереве T при вычислении $\chi(T)$.

Если функция стоимости $\chi(T)$ оценивает время вычисления булевой функции f в числе двухместных операций (временную задержку распространения сигнала в реализации функции), то процедура вычисления $\chi(T)$ отличается от предыдущей процедуры формулой расчета $\chi(t)$ для И вершины t : $\chi(t) = \chi(o) + \max(\chi(t_1), \chi(t_2), \chi(t_3))$. Если функция стоимости $\chi(T)$ описывает мощность, потребляемую реализацией булевой функции f , то процедура становится намного сложнее.

Пример 6.16. Построим дерево поиска представления функции f , показанной на рис.6.3, ограничившись разложениями ME2F.6.67, ME2F.6.78, ME2F.6.79 и функцией $c = x_i \oplus x_{i+1}$ при $i=1,2,3$. Остаточные функции от функции f :

$$V_1 = f(x_1 = \neg x_2) = x_2 \wedge x_4 \vee \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \vee x_3 \wedge x_4,$$

$$V_2 = f(x_1 = x_2) = x_2 \wedge \neg x_3 \wedge x_4 \vee \neg x_2 \wedge \neg x_4 \vee x_3 \wedge \neg x_4,$$

$$V_3 = V_1 \oplus V_2 = x_3,$$

$$V_4 = V_1(x_2 = \neg x_3) = x_4,$$

$$V_5 = V_1(x_2 = x_3) = x_3 \wedge x_4 \vee \neg x_3 \wedge \neg x_4 = \neg(x_3 \oplus x_4),$$

$$V_6 = V_4 \oplus V_5 = \neg x_3,$$

$$V_7 = V_2(x_2 = \neg x_3) = \neg x_3 \wedge x_4 \vee x_3 \wedge \neg x_4 = x_3 \oplus x_4,$$

$$V_8 = V_2(x_2 = x_3) = \neg x_4,$$

$$V_9 = V_7 \oplus V_8 = \neg x_3.$$

Используя остаточные функции $V_1 - V_9$, строим дерево поиска, показанное на рис.6.15. По сравнению с общими правилами построения дерева поиска данное дерево имеет следующие особенности:

- нетерминальные ИЛИ вершины показаны заливкой, И вершины показаны узором, для листьев заливка и узор отсутствуют;
- вершины ИЛИ помечены функциями значения V и значениями функции стоимости $\chi(V)$, однако они не помечены характеристическими функциями области определенности d ;

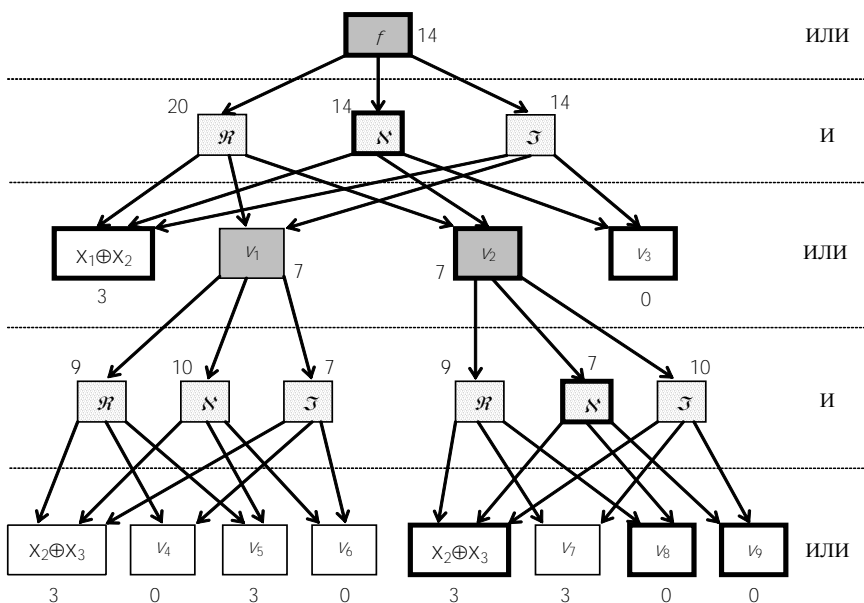


Рисунок 6.15 – Дерево поиска представления функции f

- наряду с константами и переменными, метками листьев дерева являются двухместные операции \oplus с их операндами;
- функции стоимости всех вершин выражены в числе двухместных операций \vee, \wedge .

Поиск минимального значения функции стоимости $\chi(\mathcal{T})$ всего дерева выполнялся в процессе обхода дерева снизу вверх с использованием операций суммирования для И вершин и операций взятия минимума для ИЛИ вершин. Минимальное значение $\chi(\mathcal{T}) = 14$. В процессе поиска найдено поддерево, дающее минимальное значение и состоящее из И и ИЛИ вершин, обведенных жирной линией. Поддерево сформировано на двойном применении разложения ME2F.6.78. Ему соответствует выражение

$$f = (\neg x_4 \oplus ((x_2 \oplus x_3) \wedge \neg x_3)) \oplus (x_1 \oplus x_2) \wedge x_3,$$

включающее шесть операций \wedge и \oplus . Следует отметить, что, согласно табл.6.1, полученное представление не является минимальным. Причина кроется в демонстрационном характере примера. Если расширить спектр функций c включением $c=\chi_i$ и разрешить произвольный порядок применения функций, то можно получить представление с меньшим значением функции стоимости. \square

Для сокращения обхода дерева поиска используются *нижняя* $\chi^{low}(t)$ и *верхняя* $\chi^{high}(t)$ оценки минимального значения функции стоимости вершин. Чем ближе значения оценок к фактическим минимальным значениям, тем существеннее сокращается обход. Одна из стратегий сокращения обхода сверху вниз с возвратами базируется на методе ветвей и границ. Пусть в текущий момент обхода мы находимся в вершине t и пусть $\chi^{high}(T)$ – наименьшая верхняя оценка минимального значения $\chi^{min}(T)$, найденная к текущему моменту обхода дерева T . Начальное значение $\chi^{high}(T)=\infty$.

Находясь в вершине t , нижнюю оценку стоимости всего дерева T разобьем на две части: $\chi_t^{low}(T)=\chi^{low}(t)+\chi_t^{high}(T')$, где $\chi^{low}(t)$ – нижняя оценка стоимости поддерева, начинающегося в вершине-корне t , $\chi_t^{high}(T')$ – верхняя оценка стоимости остальной части дерева T . Оценка $\chi_t^{high}(T')$ передается в вершину t при переходе в нее из родительской вершины. По мере продвижения обхода дерева, оценка $\chi_t^{high}(T')$ сходится сверху к минимальному значению $\chi_t^{min}(T')$ стоимости этой части дерева. Значение $\chi^{low}(t)$ оценивается при переходе в t из родительской вершины или при возврате в t из одной из дочерних вершин следующим образом.

Пусть t есть И вершина, которая имеет три дочерние вершины t_1, t_2, t_3 , и пусть переход в t осуществлен из родительской вершины ИЛИ. В этом случае минимальное значение $\chi^{min}(t_i)$ ни для одной из вершин $i=1,2,3$ еще не известно. Находим оценки $\chi^{low}(t_i)$ и $\chi^{high}(t_i)$ для каждой вершины i . Тогда нижняя оценка стоимости вершины t определяется как $\chi^{low}(t) = \chi(O_t) + \chi^{low}(t_1) + \chi^{low}(t_2) + \chi^{low}(t_3)$. Если $\chi^{low}(t) + \chi_t^{high}(T') \geq \chi^{high}(T)$, то поддерево с корнем в вершине t отсекается, и выполняется возврат в родительскую вершину ИЛИ. В противном случае выполняется переход из t в дочернюю вершину t_1 , в которой строится функция c , вычисляется минимальная стоимость $\chi^{min}(t_1)$ и выполняется возврат в вершину t . При переходе из t в t_1 передается верхняя оценка $\chi_{t_1}^{high}(T \cup t) = \chi_t^{high}(T') + \chi(O_{t_1}) + \chi^{high}(t_2) + \chi^{high}(t_3)$. Следую-

шим шагом является выбор одной из дочерних вершин t_2 или t_3 для перехода из t . Порядок обхода вершин может повлиять на число отсекаемых поддеревьев. Сначала нужно перейти к вершине, которая даст наиболее вероятное отсечение поддеревьев. Для этого вычисляются разности:

$$\begin{aligned}\Delta_2 &= \chi(O_i) + \chi^{\min}(t_1) + \chi^{\text{high}}(t_2) + \chi^{\text{low}}(t_3) + \chi^{\text{low}}(T) - \chi^{\text{high}}(T), \\ \Delta_3 &= \chi(O_i) + \chi^{\min}(t_1) + \chi^{\text{low}}(t_2) + \chi^{\text{high}}(t_3) + \chi^{\text{low}}(T) - \chi^{\text{high}}(T), \\ \Delta_{23} &= \chi(O_i) + \chi^{\min}(t_1) + \chi^{\text{high}}(t_2) + \chi^{\text{high}}(t_3) + \chi^{\text{low}}(T) - \chi^{\text{high}}(T),\end{aligned}$$

Если выполняется $\Delta_2 \geq 0$ $\Delta_3 < 0$, то сначала необходимо обойти вершину t_2 затем при необходимости t_3 . Если выполняется $\Delta_3 \geq 0$ и $\Delta_2 < 0$, то сначала необходимо обойти вершину t_3 затем при необходимости t_2 . Если $\Delta_{23} < 0$, то отсечение поддеревьев отсутствует при любом порядке. Если $\Delta_{23} \geq 0$, но $\Delta_2 < 0$ и $\Delta_3 < 0$, то предпочтительнее обход t_2 затем t_3 при $\Delta_2 \geq \Delta_3$, и обход t_3 затем t_2 в противном случае. При переходе из вершины t в вершину t_2 до обхода вершины t_3 передается верхняя оценка $\chi_{t_2}^{\text{high}}(T \cup t) = \chi_t^{\text{high}}(T) + \chi(O_i) + \chi^{\min}(t_1) + \chi^{\text{high}}(t_3)$. При переходе из t в t_2 после обхода t_3 передается верхняя оценка $\chi_{t_2}^{\text{high}}(T \cup t) = \chi_t^{\text{high}}(T) + \chi(O_i) + \chi^{\min}(t_1) + \chi^{\min}(t_3)$. Если произошло отсечение t_2 или t_3 , выполняется возврат в вершину t и затем в родительскую вершину ИЛИ без обновления верхней оценки $\chi^{\text{high}}(T)$. Если отсечений не было, верхняя оценка обновляется: $\chi^{\text{high}}(T) = \chi_t^{\text{high}}(T) + \chi(O_i) + \chi^{\min}(t_1) + \chi^{\text{high}}(t_2) + \chi^{\text{high}}(t_3)$.

Пусть t есть ИЛИ вершина, которая имеет дочерние вершины t_1, \dots, t_k , и пусть переход в t осуществлен из родительской вершины И. Минимальное значение $\chi^{\min}(t_i)$ не известно ни для одной вершины $i=1, \dots, k$. Находим оценки $\chi^{\text{low}}(t_i)$ и $\chi^{\text{high}}(t_i)$ для каждой вершины t_i . Упорядочиваем вершины по не убыванию нижней оценки $\chi^{\text{low}}(t_i)$. Находим в упорядоченном списке первую вершину l такую, что $\chi^{\text{low}}(t_l) + \chi^{\text{high}}(T) \geq \chi^{\text{high}}(T)$. Вершины $j=1, \dots, l-1$ сохраняем в списке, а вершины l, \dots, k вместе с соответствующими поддеревьями отсекаем. Обход дочерних вершин t_j выполняем по порядку, начиная с первой. При переходе из t в t_j передается верхняя оценка $\chi_t^{\text{high}}(T)$. Пусть теперь выполняется возврат из дочерней вершины t_j в родительскую вершину t . Если при обходе t_j произошло обновление верхней оценки $\chi^{\text{high}}(T)$, список дочерних вершин проверяется на сокращение.

Выполняется обход оставшихся вершин в порядке следования слева направо.

Возможности оптимизации частично определенных булевых функций и систем функций могут возрасти, если разложения по восьми двухместным булевым функциям используются одновременно. Тогда при построении иерархических разложений, важнейшим элементом стратегии оптимизации является выбор одного из восьми разложений для каждого частичного операнда на каждом уровне иерархии. Выбор осуществляется в соответствии с функцией стоимости $\chi(U)$. Возможно построение эвристических безвозвратных стратегий обхода дерева иерархического разложения. Возможно также построение стратегий обхода с возвратами, в частности, на базе *метода ветвей и границ*.

6.9. *Несущественные переменные и ортогональные остаточные функции*

Переменная x_i является *несущественной* в функции f , если $f_{x_i=0} = f_{x_i=1}$. В этом случае функция f равна остаточной функции: $f = x_i \wedge f_{x_i=0} \vee \neg x_i \wedge f_{x_i=0} = f_{x_i=0}$. При построении МДНФ функции f несущественность переменной x_i проявляется в том, что она не входит в выражение МДНФ. Существенная переменная всегда входит в МДНФ. Если x_i является существенной в f и f выражается через кофакторы $f_{x_i=0}$ и $f_{x_i=1}$ в соответствии с разложением Шеннона или Давио, то в каждом из кофакторов x_i становится несущественной. В этом состоит сила разложений Шеннона и Давио. На каждом шаге построения иерархического разложения эти разложения делают несущественной одну из переменных. Следовательно, если все n переменных являются существенными, глубина иерархии равна n .

С другой стороны, всегда ли можно для заданной функции f построить функции для операндов разложений (6.17)-(6.24) такие, что хотя бы в одной из них найдется, по крайней мере, одна несущественная переменная? Если учесть тот факт, что последовательное применение операций дизъюнкции и конъюнкции дает разложение Шеннона, делающее несущественной одну переменную в обеих остаточных функциях, заключаем, что применение одной операции может сделать несущественной, по крайней мере, одну

переменную в одной остаточной функции. Стратегия получения несущественной переменной x_i заключается в построении функций-компонент g разложения таким образом, что две остаточные функции $g_{x_i=0}$ и $g_{x_i=1}$ становятся идентичными. Все сказанное относится также к разложениям (6.35)-(6.42), а также к разложениям (6.51)-(6.58).

Наряду с поиском условий построения идентичных остаточных функций, важнейшим механизмом оптимизации представления функций на базе операции исключающее или является поиск условий построения ортогональных остаточных функций. Для заданной функции f и некоторой функции c остаточные функции $\min(f|c)$ и $\min(f|\neg c)$ называются *ортогональными*, если $\min(f|c) = \neg \min(f|\neg c)$. Подставляя $\neg \min(f|\neg c)$ вместо $\min(f|c)$ в разложение PE2F.6.67, получаем: $f = c \wedge \neg \min(f|\neg c) \vee \neg c \wedge \min(f|\neg c) = c \oplus \min(f|\neg c)$. Тот же результат получаем из разложения PE2F.6.76 $f = \min(f|\neg c) \oplus c \wedge (\neg \min(f|\neg c) \oplus \min(f|\neg c)) = c \oplus \min(f|\neg c)$, а также из разложения PE2F.6.77 $f = \neg \min(f|\neg c) \oplus \neg c \wedge (\neg \min(f|\neg c) \oplus \min(f|\neg c)) = c \oplus \min(f|\neg c)$. Таким образом, мы вывели еще одно очень важное разложение:

$$f = c \oplus \min(f|\neg c), \quad \text{если } \min(f|c) = \neg \min(f|\neg c). \quad (6.85)$$

Назовем его MEXF.6.85. Подставляя в правой части равенства (6.78) $\neg \min(f|\neg c)$ вместо $\min(f|c)$, получаем прямое вычисление остаточной функции $\min(f|\neg c) = f \oplus c$. Успешное применение разложения MEXF.6.85 зависит от метода одновременного построения и совместной минимизации двух булевых функций c и $\min(f|\neg c)$. Пусть обе функции представлены в классе МДНФ. Как построить функцию c , приводящую к ортогональности остаточных функций?

Возьмем предельный случай $c=f$. Тогда остаточные функции $\min(f|f)=1$ и $\min(f|\neg f)=0$. Они ортогональны, однако выгоды от такой функции c нет. Для построения других функций c воспользуемся тактикой добавления конститuent 1 к множеству f^n и исключения конститuent 1 из этого множества.

Обозначим $u = \min(f|c)$ и $w = \min(f|\neg c)$. Пусть к множеству f^n добавляется набор $a \notin f^n$. Тогда $c^{an} = f^n \cup \{a\}$ и $c(a)=1$. Как следствие $u(a)=0$ и $(f|\neg a)=dc$. Для ортогональности u и w на наборе a принима-

ем $W(a)=1$. Функции U и W уже не являются константами 1 и 0 соответственно.

Пусть теперь из множества f^{on} удаляется набор $a \in f^{on}$. Тогда $c^{on} = f^{on} \setminus \{a\}$ и $c(a)=0$. Как следствие $(f \uparrow c) = dc$ и $W(a)=0$. Для ортогональности U и W на наборе a принимаем $U(a)=1$.

Последовательная реконструкция функции c должна быть направлена на формирование меньшего числа кубов булева пространства с большим размером и меньшим числом существенных переменных в функциях c и $min(f \uparrow c)$. Надо искать слегка не завершенные кубы и завершать их в процессе конструирования функции c .

Пример 6.17. На рис.6.16 дана булева функция f , МДНФ которой строится на шести кубах карты Карно:

$$f = X_1 \wedge X_2 \vee X_1 \wedge X_4 \vee X_2 \wedge \neg X_3 \vee X_2 \wedge \neg X_4 \vee \neg X_2 \wedge X_4 \vee \neg X_3 \wedge X_4.$$

Функция стоимости $\chi(f)=11$ в числе операций \vee, \wedge . Воспользуемся разложением МЭХФ.6.85. Видим, что не завершенными являются два куба на восьми клетках каждый. Функцию c получаем из функции f изменением значения 0 на значение 1 в одной клетке, показанной заливкой на рис.6.16. Строим остаточные булевы функции $min(f \uparrow c)$ и $min(f \downarrow c)$, удовлетворяющие условию ортогональности и представленные картами Карно на рис.6.17. Клетки с инвертированными значениями показаны заливкой. В клетках без заливки значения остаточных функций совпадают со значениями функции f . Строим МДНФ функции c .

$$c = X_2 \vee X_4.$$

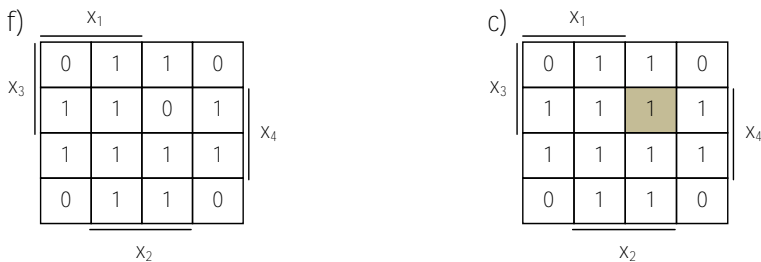


Рисунок 6.16 – Булевы функции f и C

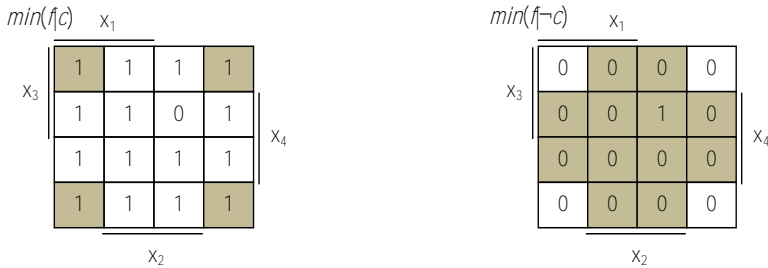


Рисунок 6.17 – Ортогональные остаточные функции $\min(f|c)$ и $\min(f|\neg c)$

Строим МДНФ остаточной функции:

$$\min(f|\neg c) = \neg x_1 \wedge x_2 \wedge x_3 \wedge x_4.$$

Согласно разложению МЭХФ.6.85 найденное представление функции f имеет вид:

$$f = (x_2 \vee x_4) \oplus (\neg x_1 \wedge x_2 \wedge x_3 \wedge x_4).$$

Функция стоимости $\chi(\hat{f})=5$ в числе операций \vee, \wedge, \oplus и $\chi(\hat{f})=7$ в числе операций \vee, \wedge, \square

6.10. Историческая справка

Задача оптимизации полностью и частично определенных логических функций и систем решалась в работах [4, 6, 7, 11, 14, 17-19, 22-24, 29, 31-34, 43, 45, 46, 72, 79, 86-88, 92] и многих других. Критериями оптимизации явились максимум быстродействия, минимум сложности или стоимости реализации, минимум энергопотребления. Разработаны методы двухуровневой и многоуровневой оптимизации. Для двухуровневой минимизации используется метод Квайна-Мак-Класки и его многочисленные модификации. Для проектов большого размера применяются различного рода эвристики. Для многоуровневой минимизации используется метод факторизации, метод двоичных диаграмм решений (графовый метод), метод обозреваемости неопределенности. Для оптимизации систем, опи-

сываемых полиномиальными представлениями, используются табличный метод, метод решения логических уравнений и др.

В работах А.А. Прихожего [53, 54, 56, 58, 59, 67, 97, 101-106] всесторонне исследованы свойства операции минимизации частично определенной функции, для которой допустимое множество описывается диапазоном булевых функций. Основываясь на этих свойствах, предложен метод оптимизации, построенный на разбиении полностью или частично определенной системы на менее определенные части с помощью частичных операций. Меньшая определенность дает большие возможности в поиске оптимальных решений. С одной стороны, число альтернативных вариантов стремительно увеличивается, что приводит к расширению пространства оптимизации. С другой стороны, в расширенном пространстве возрастают возможности найти оптимальный проект с лучшими значениями оптимизируемых параметров.

В настоящей главе книги дано систематическое изложение метода разбиения системы на менее определенные части с помощью широкого спектра предложенных разложений частично определенных функций. Благодаря многочисленным вариантам разложений функций образуется огромное пространство поиска оптимальных решений уже на двух уровнях иерархии. При использовании многоуровневой оптимизации пространство поиска стремительно расширяется.

Глава 7. ПОСТРОЕНИЕ ДИАГРАММ РЕШЕНИЙ ПО РАЗЛОЖЕНИЯМ ФУНКЦИЙ В БУЛЕВОЙ И СМЕШАННОЙ АЛГЕБРАХ

7.1. Построение диаграмм решений по разложениям функций в булевой алгебре

Основу построения диаграмм решений посредством булевой алгебры составляют три разложения полностью определенной булевой функции: разложение Шеннона, положительное разложение Давио, отрицательное разложение Давио. Применяются две формы разложения Шеннона: с использованием операции дизъюнкция \vee и с использованием операции исключающее или \oplus . Известно много типов диаграмм решений для представления булевых, псевдобулевых и многозначных функций, однако все они построены в виде ациклического ориентированного графа, проход по вершинам которого от корня к терминальным вершинам позволяет определить значение функции. В данной работе вводятся новые типы диаграмм, расширяющие возможности существующих типов. Ограничимся рассмотрением диаграмм решений, представляющих полностью и частично определенные булевы функции.

Диаграммы решений. Диаграммой решений ДР над множеством булевых переменных $X = \{x_1, \dots, x_n\}$ и множеством терминальных вершин T называется связанный, ориентированный, ациклический граф $G = (V, E)$ с одним корнем $root(G) \in V$ и следующими свойствами:

1. вершина $v \in V$ есть либо нетерминальная, либо терминальная вершина;
2. каждая нетерминальная вершина v метится переменной из X , определяемой отображением $index(v)$, и имеет ровно две вершины-последователя, обозначаемые $low(v)$ и $high(v)$;
3. каждая терминальная вершина v метится значением $value(v) \in T$ и не имеет вершин-последователей.

Дуга между вершинами v и $low(v)$ обозначается $e_{low(v)}$, дуга между вершинами v и $high(v)$ обозначается $e_{high(v)}$. На графическом изображении ДР примем, что левая дуга, исходящая из вершины v , направлена в вершину $low(v)$, правая дуга направлена в вершину

$high(v)$. В качестве множества терминальных вершин обычно выбирается $T=\{0, 1\}$. Размер диаграммы решений G равен числу нетерминальных вершин в графе и обозначается $size(G)$. Глубина диаграммы определяется как максимальная длина пути из корня в терминальную вершину и обозначается $depth(G)$. Множество вершин, помеченных переменной x_i , называется уровнем $level(x_i)$.

Диаграмма решений G называется *полной* (ПДР), если каждая переменная входит ровно один раз в каждый путь из корня в терминальную вершину на графе G .

Диаграмма решений G называется *свободной* (СвДР), если каждая переменная входит не более одного раза в каждый путь из корня в терминальную вершину на графе G .

Диаграмма решений G называется *упорядоченной* (УДР), если она является свободной, и переменные входят в одном и том же порядке в каждый путь из корня в терминальную вершину на графе G .

Пример 7.1. Пример полной упорядоченной диаграммы решений ДР приведен на рис.7.1. Диаграмма построена на упорядоченном множестве переменных $X=\{x_1, x_2, x_3\}$ и множестве терминальных вершин $T=\{0, 1\}$. □

Двоичные диаграммы решений. Диаграмма решений ДР, семантика каждой нетерминальной вершины которой определяется разложением Шеннона (1.1), называется двоичной диаграммой решений (ДДР). Каждой вершине ДДР соответствует некоторая булева функция. Если ДР определена на множестве переменных X и терминальном множестве $T=\{0, 1\}$, то вершине v соответствует булева функция f_v , определяемая следующим образом:

1. если v есть терминальная вершина и $value(v)=1$ ($value(v)=0$), то $f_v=1$ ($f_v=0$);

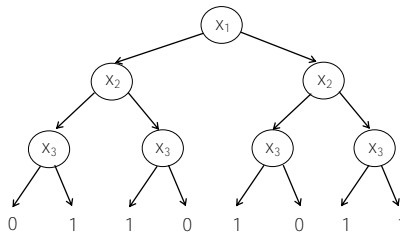


Рисунок 7.1 – Пример полной ДР

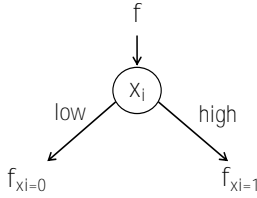


Рисунок 7.2 – Базовый фрагмент ДДР

2. если v есть нетерминальная вершина и $index(v) = x_i$, то

$$f_v = \neg x_i \wedge f_{low(v)} \vee x_i \wedge f_{high(v)}.$$

Базовый фрагмент ДДР показан на рис.7.2. Всей ДДР соответствует булева функция $f_{root(G)}$. Полная диаграмма двоичных решений (ПДДР), свободная диаграмма двоичных решений (СвДДР), упорядоченная диаграмма двоичных решений (УДДР) определяются аналогично ПДР, СвДР и УДР. Рассмотрение УДДР приводит к необходимости рассмотрения и решения проблемы нахождения *оптимального порядка переменных*. Порядок переменных описывается отображением $\pi: \{1, \dots, n\} \rightarrow X$, где $\pi(i)$ есть переменная, находящаяся на i -ой позиции. В зависимости от порядка переменных размер УДДР может изменяться от линейного до экспоненциального.

По ДДР строится алгебраическое выражение для функции $f_{root(G)}$ двумя способами. Первый способ заключается во введении промежуточных булевых переменных для нетерминальных вершин ДДР и записи формулы разложения Шеннона для каждой из нетерминальных вершин. В результате получаем многоуровневое булево выражение для функции $f_{root(G)}$. Второй способ состоит в построении ДНФ или КНФ функции при прохождении от корня к листьям диаграммы. При одном прохождении от корня к листу, помеченному 1, строится одна элементарная конъюнкция ДНФ, а при одном прохождении от корня к листу, помеченному 0, строится одна элементарная дизъюнкция КНФ.

Пример 7.2 (продолжение примера 7.1). Пример свободной упорядоченной ДДР дан на рис.7.3. По диаграмме стрится следующее многоуровневое булево выражение для функции f .

$$U_1 = \neg x_2 \wedge x_3 \vee x_2 \wedge \neg x_3,$$

$$U_2 = x_2 \vee \neg x_3,$$

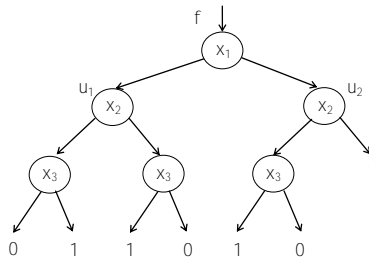


Рисунок 7.3 – Пример свободной упорядоченной ДДР

$$f = x_1 \wedge u_1 \vee \neg x_1 \wedge u_2.$$

ДНФ функции f строится по четырем путям от корня к листьям графа диаграммы, помеченным 1:

$$\text{ДНФ}(f) = \neg x_1 \wedge \neg x_2 \wedge x_3 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee x_1 \wedge \neg x_3 \vee x_1 \wedge x_2.$$

КНФ функции f строится по трем путям от корня к листьям графа диаграммы, помеченным 0:

$$\text{КНФ}(f) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3). \quad \square$$

Положительные функциональные диаграммы решений. Диаграмма решений ДР, семантика каждой нетерминальной вершины которой определяется положительным разложением Давио (1.2), называется положительной функциональной диаграммой решений (пФДР). Нетерминальной вершине v пФДР соответствует булева функция f_v , определяемая следующим образом:

- если $\text{index}(v) = x_i$, то $f_v = f_{\text{low}(v)} \oplus x_i \wedge f_{\text{high}(v)}$.

Функция, соответствующая терминальной вершине пФДР, определяется также как и в ДДР. Базовый фрагмент пФДР показан на рис.7.4. По пФДР строится многоуровневое алгебраическое выражение функции $f_{\text{root}(G)}$ с использованием положительного разложения Давио и операций \oplus и \wedge . При прохождении от корня к листьям диаграммы строится форма Рида-Маллера положительной полярности пФРМ. Одно прохождение от корня к листу порождает одну элементарную конъюнкцию пФРМ.

Пример 7.3 (продолжение примеров 7.1-7.2). Пример свобод-

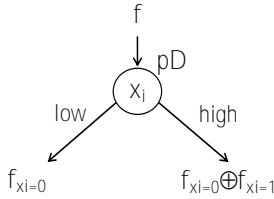


Рисунок 7.4 – Базовый фрагмент пФДР

ной упорядоченной пФДР дан на рис.7.5. По диаграмме строится следующее многоуровневое булево выражение для функции f :

$$\begin{aligned}
 U_1 &= X_2 \oplus X_3, \\
 U_2 &= 1 \oplus X_2 \wedge (1 \oplus X_3), \\
 f &= U_1 \oplus X_1 \wedge U_2.
 \end{aligned}$$

Выражение пФРМ функции f строится по путям от корня к листьям графа диаграммы, помеченным 1:

$$\text{пФРМ}(f) = x_1 \oplus x_2 \oplus x_3 \oplus x_1 \wedge x_2 \oplus x_1 \wedge x_2 \wedge x_3.$$

В выражении все вхождения переменных являются не инвертированными. Число элементарных конъюнкций соответствует числу путей от корня до 1. □

Отрицательные функциональные диаграммы решений. Диаграмма решений ДР, семантика каждой нетерминальной вершины которой определяется отрицательным разложением Давио (1.3), называется отрицательной функциональной диаграммой решений (оФДР). Нетерминальной вершине \vee оФДР соответствует булева функция f_v , определяемая следующим образом:

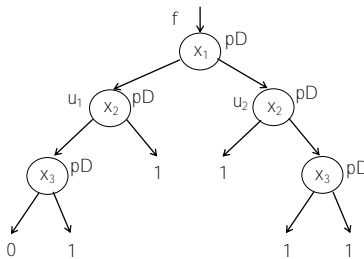


Рисунок 7.5 – Пример свободной упорядоченной пФДР

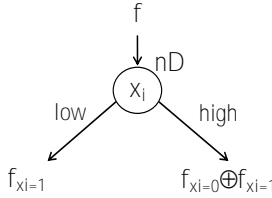


Рисунок 7.6 – Базовый фрагмент оФДР

- если $index(v) = x_i$, то $f_v = f_{low(v)} \oplus \neg x_i \wedge f_{high(v)}$.

Базовый фрагмент оФДР показан на рис.7.6. По оФДР строится многоуровневое алгебраическое выражение функции $f_{root(G)}$ с использованием отрицательного разложения Давио и операций \oplus , \wedge и \neg . При прохождении от корня к листьям диаграммы строится форма Рида-Маллера отрицательной полярности оФРМ. Одно прохождение от корня к листу порождает одну элементарную конъюнкцию оФРМ.

Пример 7.4 (продолжение примеров 7.1-7.3). Пример свободной упорядоченной оФДР дан на рис.7.7. По диаграмме строится следующее многоуровневое булево выражение для функции f .

$$\begin{aligned}
 u_1 &= 1 \oplus \neg x_2 \wedge (1 \oplus \neg x_3), \\
 u_2 &= 1 \oplus \neg x_3 \oplus \neg x_2 \wedge \neg x_3, \\
 f &= u_1 \oplus \neg x_1 \wedge u_2.
 \end{aligned}$$

Выражение оФРМ функции f строится по пяти путям от корня к листьям графа диаграммы, помеченным 1:

$$\text{oФРМ}(f) = 1 \oplus \neg x_1 \oplus \neg x_2 \oplus \neg x_1 \wedge \neg x_3 \oplus \neg x_2 \wedge \neg x_3 \oplus \neg x_1 \wedge \neg x_2 \wedge \neg x_3. \quad \square$$

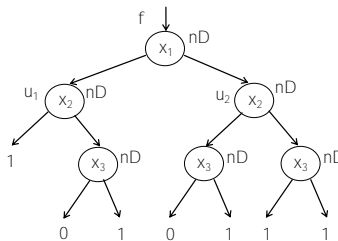


Рисунок 7.7 – Пример свободной упорядоченной оФДР

Список типов декомпозиции. Обобщением трех типов диаграмм решений является ДР, в которой нетерминальной вершине ставится в соответствие одно из трех разложений булевой функции: разложение Шеннона (S) (1.4); положительное разложение Давио (pD) (1.2), отрицательное разложение Давио (nD) (1.3). Указанные разложения ассоциируются с булевыми переменными x_1, \dots, x_n посредством списка типов декомпозиции (СТД) $d = (d_1, \dots, d_n)$, где $d_i \in \{S, pD, nD\}$. Таким образом, всем нетерминальным вершинам, помеченным одной и той же переменной, соответствует один тип декомпозиции. Такого рода диаграммы называются функциональными диаграммами решений Кронекер (КФДР). По КФДР строится многоуровневое алгебраическое выражение функции $f_{root(G)}$ с использованием разложений Шеннона и Давио, а также операций \wedge , \oplus и \neg . По этому типу диаграмм строится выражение Кронекер (форма Рида-Маллера *смешанной полярности*), которая по числу элементарных конъюнкций может быть предпочтительнее пФРМ и оФРМ.

Пример 7.5 (продолжение примеров 7.1-7.4). Пример свободной упорядоченной КФДР дан на рис.7.8. Для переменных $X = (x_1, x_2, x_3)$ выбран список типов декомпозиции $d = (nD, S, pD)$. Многоуровневое булево выражение для функции f , представленной диаграммой:

$$u_1 = ((1 \oplus x_3) \wedge \neg x_2) \oplus (1 \wedge x_2),$$

$$u_2 = (1 \wedge \neg x_2) \oplus (x_3 \wedge x_2),$$

$$f = u_1 \oplus \neg x_1 \wedge u_2.$$

ФРМ смешанной полярности общего вида функции f строится следующим образом:

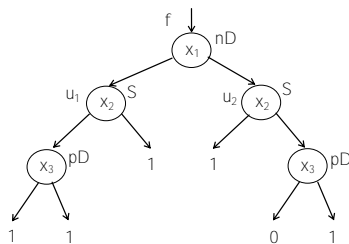


Рисунок 7.8 – Пример свободной упорядоченной КФДР

$$\Phi PM(\hat{f}) = 1 \oplus \neg x_1 \wedge \neg x_2 \oplus \neg x_2 \wedge x_3 \oplus \neg x_1 \wedge x_2 \wedge x_3.$$

Переменная x_1 имеет отрицательную полярность, поскольку ей соответствует разложение типа nD . Переменная x_3 имеет положительную полярность, поскольку ей соответствует разложение типа pD . Полярность переменной x_2 не фиксирована, поскольку ей соответствует разложение типа S . \square

Сокращенные диаграммы решений. Для заданного порядка π переменных и заданного списка d типов декомпозиции существует ряд диаграмм решений, представляющих одну и ту же булеву функцию f . *Сокращенная диаграмма решений* (СДР) строится из исходной ДР применением трех правил редуцирования.

Правило I. Если для двух нетерминальных вершин $v, u \in V$ выполняется $index(v) = index(u)$, $low(v) = low(u)$ и $high(v) = high(u)$, то вершина u исключается из графа, а все входящие в нее дуги переориентируются на вершину v .

Правило S. Если для вершины $v \in V$ выполняется $low(v) = high(v) = u$, то вершина v исключается из графа, а все входящие в нее дуги переориентируются на вершину u .

Правило D. Если для вершины $v \in V$ вершина $high(v) = u$ есть терминальная вершина, для которой $value(u) = 0$, то вершина v исключается из графа, а все входящие в нее дуги переориентируются на вершину $low(v)$.

Правила редуцирования I, S, D иллюстрируются на рис.7.9-7.11. Заливкой показана удаляемая вершина. Применение правила I ко всем трем типам вершин S, pD, nD, правила S к вершинам типа S и правила D к вершинам типа pD и nD не изменяет описываемую диаграммой функцию. Получаемая в результате редуцирования сокращенная диаграмма является уникальной и единственной. Применение правил I и S к УДДР дает сокращенную упорядоченную двоичную диаграмму решений (СУДДР). Применение правил I и D к упорядоченным пФДР и оФДР дает сокращенные упорядоченные функциональные диаграммы решений СУпФДР и СУоФДР соответственно. На сегодняшний день СУДДР является наиболее широко используемой формой представления систем булевых функций при решении задач моделирования, верификации, синтеза логических схем. Сила СУДДР заключается в ее компактности, что обуславливает высокое быстродействие алгоритмов манипулирования.

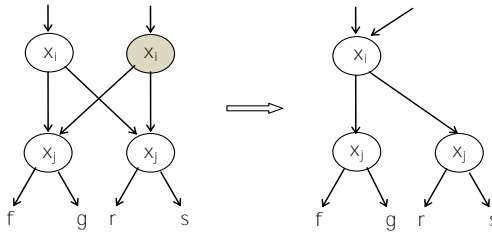


Рисунок 7.9 – Правило редуцирования I

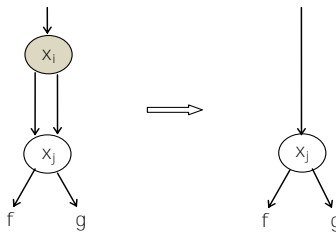


Рисунок 7.10 – Правило редуцирования S

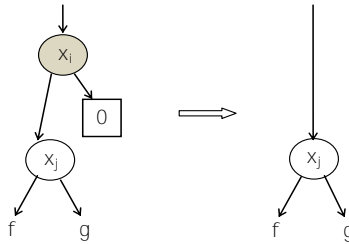


Рисунок 7.11 – Правило редуцирования D

Пример 7.6. (продолжение примеров 7.2, 7.4). Применение правил I и S к свободной упорядоченной ДДР, изображенной на рис 7.3, дает СУДДР, представленную на рис.7.12. Применение правил I и D к свободной упорядоченной оФДР, изображенной на рис 7.7, дает СУоФДР, представленную на рис.7.13. □

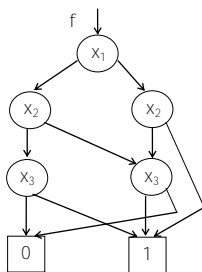


Рисунок 7.12 – Пример сокращенной упорядоченной СУДДР

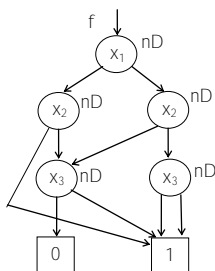


Рисунок 7.13 – Пример сокращенной упорядоченной СУоФДР

Диаграммы решений с инвертированными дугами. Если в интерпретированной диаграмме решений (ДДР, пФДР, оФДР, КФДР) некоторой вершине v соответствует булева функция f , а другой вершине u соответствует булева функция $\neg f$, то вершина u может быть удалена из диаграммы, а все входящие в нее дуги заменены инвертированными дугами, переориентированными в вершину v . Инвертированные дуги будем изображать пунктирными линиями. Поскольку все дуги, входящие в терминальную вершину 1, могут быть заменены инвертированными дугами и переориентированы в вершину 0, множество терминальных вершин может быть сокращено до $T=\{0\}$. При появлении инвертированных дуг общее число вершин в диаграмме также сокращается. Сокращенные названия диаграмм с инвертированными дугами начинаются с буквы И.

Пример 7.7. (продолжение примера 7.6). Введем инвертированные дуги в СУДДР, изображенную на рис 7.12. Соответствующая ИСУДДР представлена на рис.7.14. Благодаря введению инвертированных дуг число вершин сократилось на 2, с 7 до 5. \square

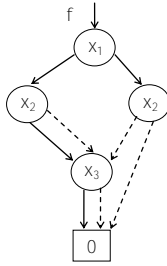


Рисунок 7.14 – Пример ИСУДДР

7.2. Построение диаграмм решений по разложениям функций в смешанной алгебре

В настоящем параграфе введем основные типы *if* диаграмм решений. В следующих параграфах изучим диаграммы основных типов более детально, рассмотрим важнейшие частные случаи диаграмм.

If диаграммы решений. *If* диаграммой решений *lf*ДР над множеством булевых переменных $X = \{x_1, \dots, x_n\}$ и множеством значений-меток $T = \{0, 1\}$ называется связанный, ориентированный, ациклический граф $G = (V, E)$ с одним корнем $root(G) \in V$ и следующими свойствами:

1. вершина $v \in V$ есть либо нетерминальная, либо терминальная вершина;
2. каждая нетерминальная вершина v именуется *ifd* и имеет ровно три вершины-последователя, обозначаемые $if(v)$, $high(v)$, $low(v)$;
3. каждая терминальная вершина v метится меткой $label(v)$, которой может быть переменная $x_i \in X$, инверсия переменной $\neg x_i$, значение $t \in T$; терминальная вершина не имеет вершин-последователей.

Дуга между вершинами v и $if(v)$ обозначается $e_{if(v)}$; дуга между вершинами v и $high(v)$ обозначается $e_{high(v)}$; дуга между вершинами v и $low(v)$ обозначается $e_{low(v)}$. На графическом изображении *lf*ДР примем, что левая дуга, исходящая из вершины v , направлена в вершину $if(v)$, средняя дуга направлена в вершину $high(v)$, правая

дуга направлена в вершину $low(v)$. Размер диаграммы решений $size(G)$ равен суммарному числу вершин в графе. Число нетерминальных вершин обозначим через $size_n(G)$, число терминальных вершин – через $size_t(G)$. Глубина диаграммы $depth(G)$ равна максимальной длине пути из корня в терминальную вершину. В длине пути учитываются только нетерминальные вершины.

Пример 7.8. Для построения *if* диаграмм решений в данном и последующих примерах воспользуемся функцией, определенной в примере 7.9. На рис.7.15 приведен пример *if* диаграммы решений *lf*ДР, построенной на семи нетерминальных вершинах *ifd* и пятнадцати терминальных вершинах. Глубина диаграммы равна трем. □

Двоичные if диаграммы решений. Диаграмма решений *lf*ДР, семантика каждой нетерминальной вершины которой определяется разложением ME2F.6.67, называется двоичной *if* диаграммой решений (Д*lf*ДР). Каждой вершине Д*lf*ДР соответствует некоторая булева функция. Если *lf*ДР определена на множестве переменных X и множестве значений $T=\{0, 1\}$, то вершине v соответствует булева функция f_v , определяемая следующим образом:

1. если v есть терминальная вершина и
 если $label(v)=x_i$, то $f_v=x_i$,
 если $label(v)=\neg x_i$, то $f_v=\neg x_i$,
 если $label(v)=0$, то $f_v=0$,
 если $label(v)=1$, то $f_v=1$,
2. если v есть нетерминальная вершина, то
 $f_v = f_{if(v)} \wedge f_{high(v)} \vee \neg f_{if(v)} \wedge f_{low(v)}$.

Базовый фрагмент Д*lf*ДР показан на рис.7.16. Всей Д*lf*ДР

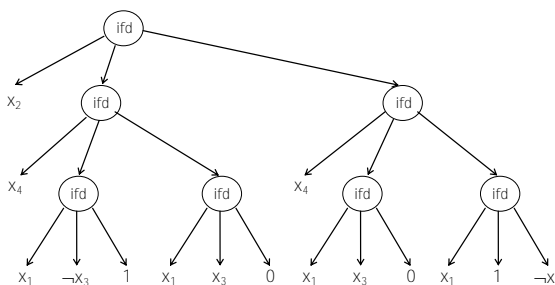


Рисунок 7.15 – Пример *lf*ДР

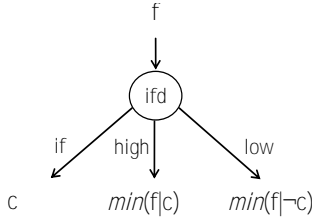


Рисунок 7.16 – Базовый фрагмент Д|fДР

соответствует булева функция $f_{root(G)}$. По Д|fДР строится алгебраическое многоуровневое выражение для функции $f_{root(G)}$ в виде суперпозиции функций \vee , \wedge , \neg с использованием дополнительных локальных булевых переменных. Это выражение можно представить как систему ДНФ или систему КНФ.

Пример 7.9. (продолжение примера 7.8). На рис.7.17 изображена Д|fДР, представляющая булеву функцию f , определенную картой Карно на рис. 6.3. Многоуровневое булево выражение для функции f :

$$\begin{aligned}
 U_1 &= X_2 \wedge \neg X_4 \vee \neg X_2 \wedge X_4, \\
 U_2 &= X_1 \wedge X_2, \\
 U_3 &= \neg X_1 \vee \neg X_3, \\
 f &= U_1 \wedge U_2 \vee \neg U_1 \wedge U_3.
 \end{aligned}$$

Следует отметить, если |fДР из рис.7.15 проинтерпретировать как Д|fДР, то описываемая ею функция совпадает с функцией, описываемой Д|fДР из рис.7.17, однако размер и глубина последней существенно меньше. □

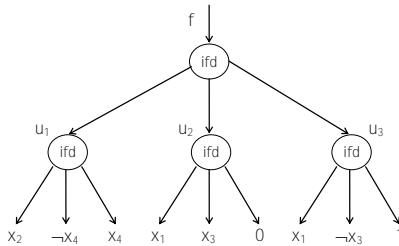


Рисунок 7.17 – Пример Д|fДР

Положительные функциональные if диаграммы решений.

Диаграмма решений IfДР, семантика каждой нетерминальной вершины которой определяется разложением МЕ2F.6.76, называется положительной функциональной if диаграммой решений (пФIfДР). Нетерминальной вершине v пФIfДР соответствует булева функция f_v , определяемая следующим образом: $f_v = f_{high(v)} \oplus f_{if(v)} \wedge f_{low(v)}$. Функция, соответствующая терминальной вершине пФIfДР, определяется так же, как и в ДIfДР. Базовый фрагмент пФIfДР показан на рис.7.18. Каждая нетерминальная вершина пФIfДР метится меткой pf . По пФIfДР строится алгебраическое многоуровневое выражение в виде суперпозиции функций \oplus, \wedge с использованием дополнительных локальных булевых переменных. Это выражение может быть трансформировано к пФРМ.

Пример 7.10. (продолжение примера 7.9). На рис.7.19 изображена пФIfДР, представляющая булеву функцию f . Многоуровневое булево выражение для функции f .

$$u_1 = x_4 \oplus (x_2 \wedge 1),$$

$$u_2 = 1 \oplus (x_1 \wedge x_3),$$

$$f = u_2 \oplus u_1 \wedge 1.$$

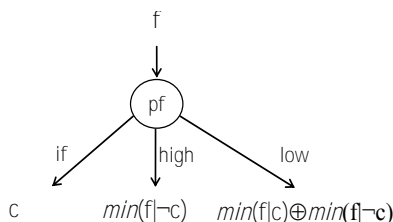


Рисунок 7.18 – Базовый фрагмент пФIfДР

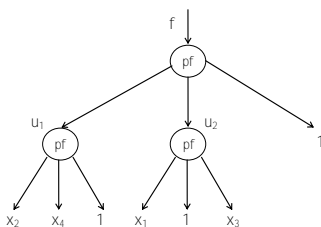


Рисунок 7.19 – Пример пФIfДР

Подставляя выражения для U_1 и U_2 в выражение для f , получаем пФРМ для функции f .

$$\text{пФРМ}(f) = 1 \oplus x_1 \wedge x_3 \oplus x_2 \oplus x_4. \quad \square$$

Отрицательные функциональные if диаграммы решений.
 Диаграмма решений IfDR , семантика каждой нетерминальной вершины которой определяется разложением ME2F.6.77, называется отрицательной функциональной *if* диаграммой решений (оФIfDR). Нетерминальной вершине v оФIfDR соответствует булева функция f_v , определяемая следующим образом: $f_v = f_{\text{high}(v)} \oplus \neg f_{\text{if}(v)} \wedge f_{\text{low}(v)}$. Функция, соответствующая терминальной вершине оФIfDR, определяется также как и в DIfDR . Базовый фрагмент оФIfDR показан на рис.7.20. Каждая нетерминальная вершина оФIfDR метится меткой *nf*. По оФIfDR строится алгебраическое многоуровневое выражение в виде суперпозиции функций \oplus, \wedge, \neg с использованием дополнительных локальных булевых переменных. Это выражение может быть трансформировано к оФРМ.

Пример 7.11. (продолжение примеров 7.9, 7.10). На рис.7.21 изображена оФIfDR, представляющая булеву функцию f .

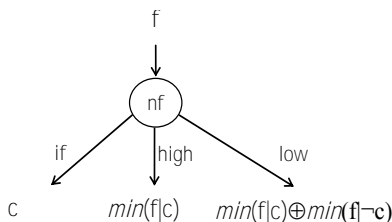


Рисунок 7.20 – Базовый фрагмент оФIfDR

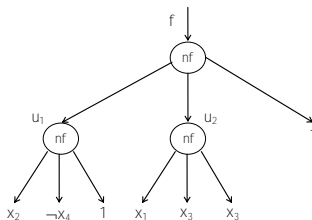


Рисунок 7.21 – Пример оФIfDR

Многоуровневое булево выражение для функции f :

$$\begin{aligned} u_1 &= \neg x_4 \oplus \neg x_2 \wedge 1, \\ u_2 &= (1 \oplus \neg x_3) \oplus \neg x_1 \wedge (1 \oplus \neg x_3), \\ f &= u_2 \oplus \neg u_1 \wedge 1. \end{aligned}$$

Подставляя выражения для u_1 и u_2 в выражение для f , получаем оФРМ для функции f :

$$\text{оФРМ}(f) = \neg x_1 \oplus \neg x_2 \oplus \neg x_3 \oplus \neg x_4 \oplus \neg x_1 \wedge \neg x_3. \quad \square$$

Комбинированные if диаграммы решений. Диаграмма решений IfДР , семантика каждой нетерминальной вершины которой определяется одним из трех разложений ME2F.6.78, ME2F.6.76 и ME2F.6.77, называется комбинированной *if* диаграммой решений (KIfДР). Возможность использования разложений трех типов в одной диаграмме усиливает ее изобразительные возможности и повышает компактность представления функций. Расширение KIfДР по сравнению с КФДР состоит не только в том, что разложения ME2F.6.67, ME2F.6.76 и ME2F.6.77 обобщают разложения Шеннона и Давио, но и в том, что KIfДР не устанавливает никакого порядка применения разложений. Заметим, что КФДР использует список типов декомпозиций, ассоциируемый со списком переменных, и применяет типы разложений по уровням, ассоциируемым с переменными. Выбор для текущей вершины v предпочтительного типа декомпозиции зависит от вида функции f , соответствующей этой вершине, и возможности эффективного построения функции c . Параметры остаточных функций $\min(f|c)$, $\min(f|\neg c)$ и $\min(f|c) \oplus \min(f|\neg c)$ имеют решающее влияние на выбор типа декомпозиции. По KIfДР строится алгебраическое многоуровневое выражение в виде суперпозиции функций \oplus , \wedge , \neg с использованием дополнительных локальных булевых переменных. Диаграмме KIfДР соответствует форма Рида-Маллера самого *общего вида*.

Пример 7.12. (продолжение примеров 7.9-7.11). На рис.7.22 изображена KIfДР , представляющая все ту же булеву функцию f . Многоуровневое булево выражение для функции f :

$$\begin{aligned} u_1 &= x_4 \oplus (x_2 \wedge 1), \\ u_2 &= x_1 \wedge x_3 \oplus \neg x_1 \wedge 0, \\ f &= u_2 \oplus \neg u_1 \wedge 1. \end{aligned}$$

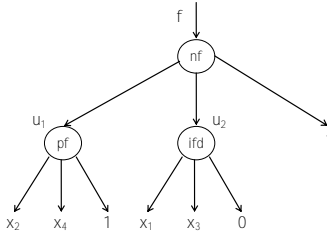


Рисунок 7.22 – Пример KIfDR

Подставляя выражения для u_1 и u_2 в выражение для f , получаем ФРМ для функции f :

$$\text{ФРМ}(f) = x_1 \wedge x_3 \oplus x_2 \oplus \neg x_4.$$

Число элементарных конъюнкций в этой форме является минимальным. \square

7.3. Двоичные if диаграммы решений

Изучим свойства и рассмотрим типы двоичных if диаграмм решений. Следует заметить, что в ранних известных работах был введен оператор *if-then-else*, который использовался в качестве технического средства реализации процедур и пакетов программ для манипулирования двоичными диаграммами решений. В то же время для этого оператора полностью отсутствовала семантическая база в виде разложений не полностью определенных функций.

Сокращенные двоичные if диаграммы решений. Для одной и той же функции существует ряд двоичных if диаграмм решений, отличающихся размером и глубиной. Сокращенная двоичная if диаграмма решений (СДIfDR) не содержит двух изоморфных поддиаграмм. СДIfDR строится из исходной DIfDR применением правил редуцирования. СДIfDR расширяет множество альтернативных представлений функции по сравнению с СУДDR.

Правило I_{if} . Если для двух нетерминальных вершин $v, u \in V$ выполняется $if(v) = if(u)$, $high(v) = high(u)$ и $low(v) = low(u)$, то вершина u исключается из графа, а все входящие в нее дуги переориентируются на вершину v . Правило I_{if} иллюстрируется рис.7.23.

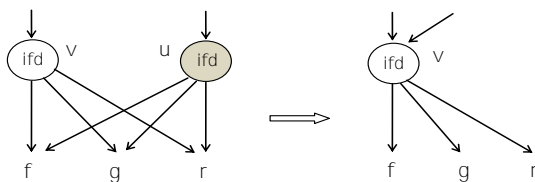


Рисунок 7.23 – Правило редуцирования I_{if}

Правило S_{if} . Если для вершины $v \in V$ выполняется $high(v) = low(v) = g$, то вершина v и подграф, начинающийся в вершине $if(v)$, исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину g . Правило S_{if} иллюстрируется рис.7.24.

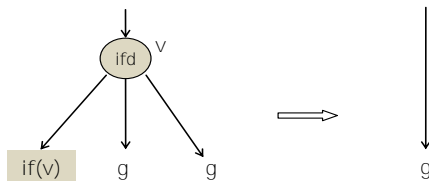


Рисунок 7.24 – Правило редуцирования S_{if}

Правило $High_1$. Если для вершины $v \in V$ выполняется $if(v) = 1$, то вершина v , вершина $if(v)$ и подграф, начинающийся в вершине $h = low(v)$ исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину $g = high(v)$. Правило $High_1$ иллюстрируется рис.7.25.

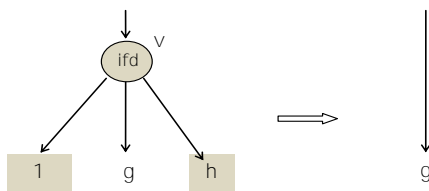


Рисунок 7.25 – Правило редуцирования $High_1$

Правило LOW_0 . Если для вершины $v \in V$ выполняется $if(v) = 0$, то вершина v , вершина $if(v)$ и подграф, начинающийся в вершине $g = high(v)$ исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину $h = low(v)$. Правило LOW_0 иллюстрируется рис.7.26.

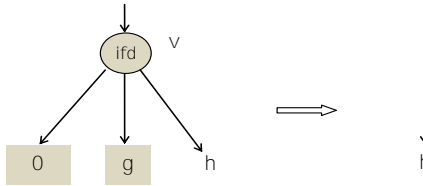


Рисунок 7.26 – Правило редуцирования Low_0

Правило lf_{10} . Если для вершины $v \in V$ выполняется $high(v)=1$ и $low(v)=0$, то вершина v , вершина $high(v)$ и вершина $low(v)$ исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину $c=if(v)$. Правило lf_{10} иллюстрируется рис.7.27.

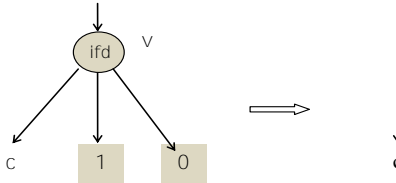


Рисунок 7.27 – Правило редуцирования lf_{10}

Правило lf_{01} . Если для вершины $v \in V$ выполняется $high(v)=0$ и $low(v)=1$, то вершина v , вершина $high(v)$ и вершина $low(v)$ исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину $c=\neg if(v)$. Правило lf_{01} иллюстрируется рис.7.28. Если вершина c терминальная, то операция \neg инвертирования выполняется над литералом, если нетерминальная, операция выполняется над поддиаграммой.

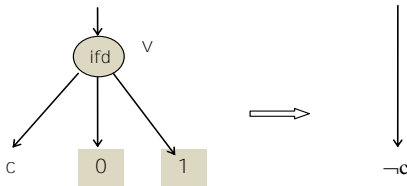


Рисунок 7.28 – Правило редуцирования lf_{01}

Правило lf_{not} . Если для вершины $v \in V$ выполняется $if(v)=\neg c$, то вершины $high(v)$ и $low(v)$ меняются местами, а вершина $if(v)$ метится литералом c . Правило lf_{not} иллюстрируется рис.7.29.

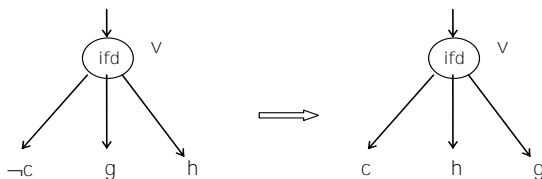


Рисунок 7.29 – Правило редуцирования If_{not}

Правило If_{high} . Если для вершины $v \in V$ выполняется $if(v) = high(v) = c$, то вершина $high(v)$ заменяется на терминальную вершину с меткой 1. Правило If_{high} иллюстрируется рис.7.30.

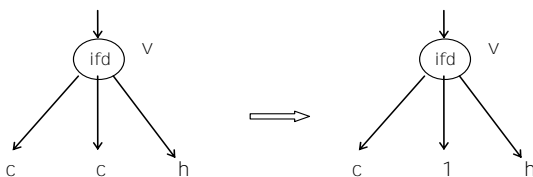


Рисунок 7.30 – Правило редуцирования If_{high}

Правило If_{low} . Если для вершины $v \in V$ выполняется $if(v) = low(v) = c$, то вершина $low(v)$ заменяется на терминальную вершину с меткой 0. Правило If_{low} иллюстрируется рис.7.31.

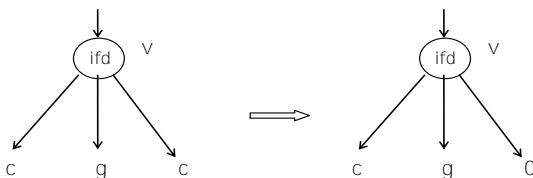


Рисунок 7.31 – Правило редуцирования If_{low}

Пример 7.13. (продолжение примера 7.8). На рис.7.32 изображена СД|fДР, полученная из |fДР, представленной на рис.7.15, путем ее интерпретации в виде Д|fДР с последующим сокращением. Размер СД|fДР уменьшился по сравнению с исходной Д|fДР. □

Упорядоченные двоичные if диаграммы решений. Построение упорядоченных ДДР связано с фиксацией порядка переменных при проходе от корня диаграммы к ее листьям. Построение упорядоченных Д|fДР (УД|fДР) связано с фиксацией порядка переменных

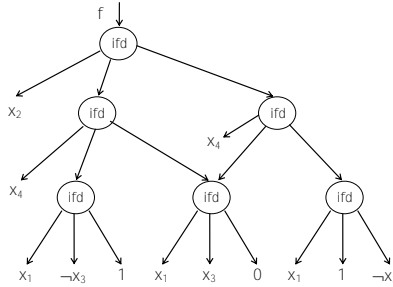


Рисунок 7.32 – Пример CIfDP

при обходе графа слева направо. Например, DIfDP является упорядоченной, если для любой вершины v все переменные, входящие в подграф $if(v)$, предшествуют переменным, входящим в подграфы $high(v)$ и $low(v)$.

Двоичные if диаграммы решений с инвертированными дугами.

В основе построения DIfDP с инвертированными дугами (ИDIfDP) лежит использование операции инвертирования булевой функции, представленной диаграммой решений.

Если функция f представлена DIfDP G , то инверсная функция $\neg f$ представляется DIfDP $G_{not} = not(G)$, где not – операция инвертирования диаграммы, определяемая рекурсивно следующим образом:

1. если v есть терминальная вершина в диаграмме G , то в диаграмму G_{not} вводится терминальная вершина v' , для которой:

$$\begin{aligned} label(v') &= \neg x_i, \text{ если } label(v) = x_i, \\ label(v') &= x_i, \text{ если } label(v) = \neg x_i, \\ label(v') &= 1, \text{ если } label(v) = 0, \\ label(v') &= 0, \text{ если } label(v) = 1, \end{aligned}$$

2. если v есть нетерминальная вершина в диаграмме G , то в диаграмму G_{not} вводится нетерминальная вершина v' такая, что $if(v') = if(v)$, $high(v') = not(high(v))$ и $low(v') = not(low(v))$.

Операция инвертирования DIfDP иллюстрируется рис.7.33. Дуга e в ИDIfDP, входящая в вершину v , называется *инвертированной*, если ей соответствует булева функция $f_e = \neg f_v$. Инвертированная дуга изображается штриховой линией. Следующее правило вводит инвертированные дуги в ИDIfDP.

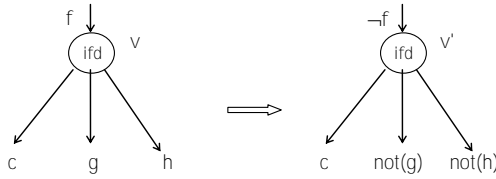


Рисунок 7.33 – Операция инвертирования ДИfДР

Правило C_{if} . Если для вершин $v, u \in V$ выполняется $if(u) = if(v)$, $high(u) = not(high(v))$ и $low(u) = not(low(v))$, то вершины u , $high(u)$ и $low(u)$ исключаются из графа диаграммы, а все дуги, входящие в вершину u , заменяются инвертированными дугами и переориентируются на вершину v .

Правило C_{if} иллюстрируется рис.7.34. В ИДИfДР множество типов меток терминальных вершин сокращается до двух. Метками являются переменная x_i и константа 0.

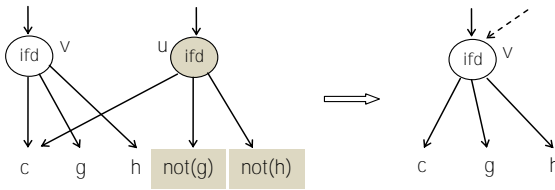


Рисунок 7.34 – Правило редуцирования C_{if}

Пример 7.14. (продолжение примера 7.9). На рис.7.35 изображена ИДИfДР, полученная путем инвертирования дуг в ДИfДР, представленной на рис.7.17. Поскольку для функций u_2 и u_3 выполняется $u_3 = \neg u_2$, то вершина, которой соответствует функция u_3 , удаляется, а входящая в нее дуга становится инвертированной и переориентируется на вершину, которой соответствует функция u_2 . □

7.4. Частные случаи двоичных if диаграмм решений

Прежде всего, частные случаи двоичных if диаграмм решений ДИfДР образуются путем конкретизации и фиксации вида функции c в разложении ME2F.6.67. Важнейшим частным случаем ДИfДР являются диаграммы двоичных решений ДДР. Действительно, если в разложении ME2F.6.67 функции f выбирать на уровне i разложения

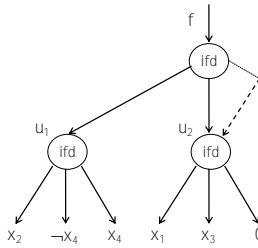


Рисунок 7.35 – Пример ИД|fДР

в качестве функции C переменную x_i , то получим УД|fДР, эквивалентную УДДР. Структура такой УД|fДР показана на рис.7.36. Применение к полученной УД|fДР правил редуцирования $I_{if-f_{low}}$, описанных в предыдущем разделе, дает сокращенную упорядоченную СУД|fДР, эквивалентную СУДДР для функции f . Построенная СУД|fДР, также как и СУДДР, является канонической формой представления булевых функций.

Пример 7.15. (продолжение примера 6.9). Построим СУДДР (рис.7.37) для булевой функции, определенной картой Карно на рис.6.3. Эквивалентная ей СУД|fДР показана на рис.7.38. Она содержит 5 нетерминальных вершин и 9 терминальных вершин. Для сравнения, описывающая ту же булеву функцию Д|fДР, содержит 3 нетерминальные вершины и 9 терминальных вершин. □

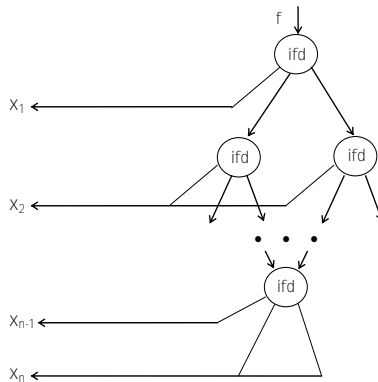


Рисунок 7.36 – УД|fДР, эквивалентная УДДР

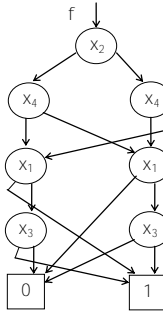


Рисунок 7.37 – Пример СУДП

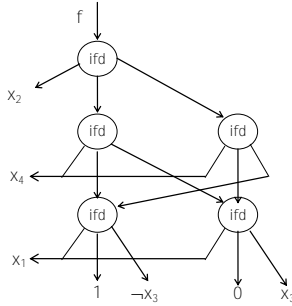


Рисунок 7.38 – Пример СУДПifD

Другая каноническая форма строится на базе ДifDП посредством выбора в разложении ME2F.6.67 на уровне иерархии i функции $C = X_i \oplus X_{i+1}$. Структура получаемой УДifDП показана на рис.7.39. Назовем ее УДifDП $\oplus 2$.

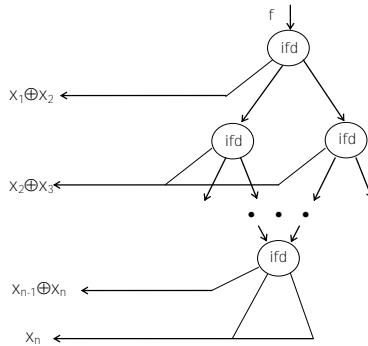


Рисунок 7.39 – Каноническая форма УДifDП $\oplus 2$

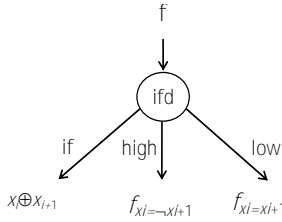


Рисунок 7.40 – Базовый фрагмент УД|fДР⊕2

Базовый фрагмент УД|fДР⊕2 показан на рис.7.40. В нем использованы исключающий отрицательный кофактор $f_{x_i = -x_j}$ и исключающий положительный кофактор $f_{x_i = x_j}$ (см. раздел 3.7). Каждая операция \oplus легко представляется одной вершиной Д|fДР. Корректность построения УД|fДР⊕2 непосредственно вытекает из результатов раздела 3.7. На уровне иерархии i диаграммы переменная x_i становится несущественной для всех последующих уровней, так как вершинам последующих уровней соответствуют исключающие кофакторы исходной функции.

Пример 7.16. (продолжение примера 7.15). Построим УД|fДР⊕2 (рис.7.41) для булевой функции, описываемой СУД|fДР, показанной на рис.7.38. Использование на верхнем уровне функции $c = x_2 \oplus x_4$ делает несущественными сразу две переменные x_2, x_4 . Диаграмма содержит 5 нетерминальных и 10 терминальных вершин. □

Третья каноническая форма строится на базе Д|fДР посредством выбора в разложении МЕ2F.6.67 на уровне иерархии i функции $c = x_i \oplus \dots \oplus x_n$. Назовем эту форму УД|fДР⊕. Структура получаемой УД|fДР⊕ показана на рис.7.42. Корректность построения

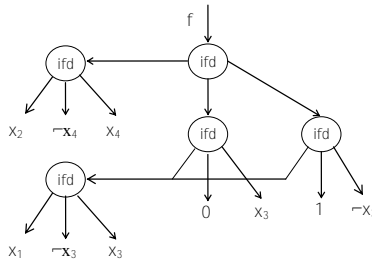


Рисунок 7.41 – Пример УД|fДР⊕2

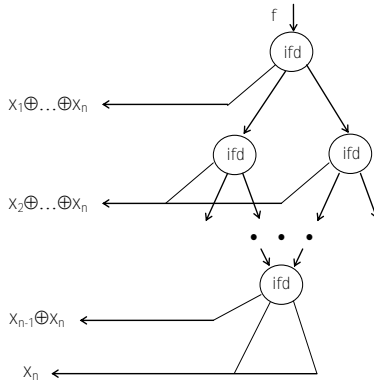


Рисунок 7.42 – Каноническая форма УДfДР⊕

УДfДР⊕ обусловлена тем, что в остаточных функциях, получаемых при использовании функции $C = X_1 \oplus \dots \oplus X_n$, одна из переменных X_1, \dots, X_n (в частности, переменная X_i) становится несущественной. Заменяя последовательность многоместных операций $X_1 \oplus \dots \oplus X_n, X_2 \oplus \dots \oplus X_n, \dots, X_{n-1} \oplus X_n, X_n$ иерархией вершин *ifd* и используя инвертированные дуги, изображаемые штриховой линией, получаем структуру, представленную на рис.7.43.

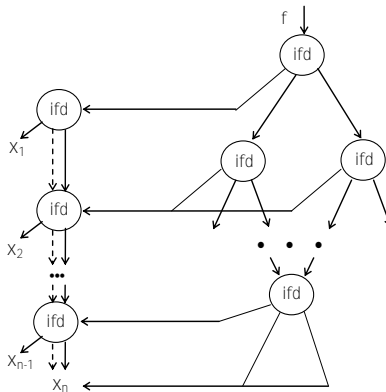


Рисунок 7.43 –УДfДР⊕ с заменой операций ⊕ вершинами *ifd*

7.5. Функциональные *if* диаграммы решений

Рассмотрим типы и изучим свойства функциональных *if* диаграмм решений.

Сокращенные функциональные if диаграммы решений. Одна булева функция может быть представлена различными функциональными *if* диаграммами решений, отличающимися размером и глубиной. *Сокращенная функциональная if диаграмма решений* (СФ|fДР) не содержит двух изоморфных поддиаграмм. СФ|fДР строится по исходной Ф|fДР путем применения правил редуцирования. Положительные и отрицательные СФ|fДР расширяют множество альтернативных представлений функции по сравнению с положительными и отрицательными СФДР. Нетерминальную вершину Ф|fДР, семантика которой определяется разложением ME2F.6.76, будем метить меткой *pf*. Нетерминальную вершину, семантика которой определяется разложением ME2F.6.77, будем метить меткой *nf*. Рассмотрим правила редуцирования для вершин типа *pf*. Правила редуцирования для вершин типа *nf* строятся аналогичным образом.

Правило I_{if} , введенное в разделе 7.3, применимо как к Д|fДР, так и к Ф|fДР. Оно выполняет слияние изоморфных в диаграмме вершин.

Правило D_{if} . Если для вершины $v \in V$ выполняется $low(v)=0$ и $u=high(v)$ то вершины v , $low(v)$ и вершины подграфа, начинающегося в вершине $if(v)$, исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину u . Правило D_{if} иллюстрируется рис.7.44. Как и прежде, заливкой указаны удаляемые вершины.

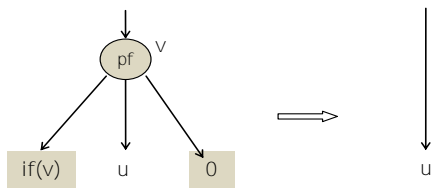


Рисунок 7.44 – Правило редуцирования D_{if}

Правило F_{if} . Если для вершины $v \in V$ выполняется $high(v)=low(v)=g$, то вершина $low(v)$ заменяется терминальной вершиной с

меткой 0, а вершина $if(v)$ заменяется вершиной $not(if(v))$, где not – операция инвертирования поддиаграммы пФлДР. Правило F_{if} иллюстрируется рис.7.45.

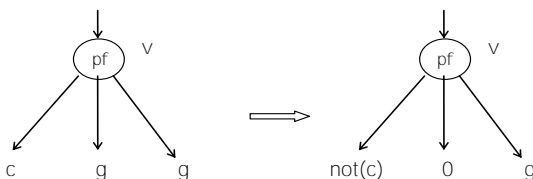


Рисунок 7.45 – Правило редуцирования F_{if}

Правило LOW_{F_0} . Если для вершины $v \in V$ выполняется $if(v)=0$ либо $high(v)=0$, то вершина v , вершина и подграф $if(v)$, а также вершина и подграф $high(v)$ исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину $h=low(v)$. Правило LOW_{F_0} иллюстрируется рис.7.46.

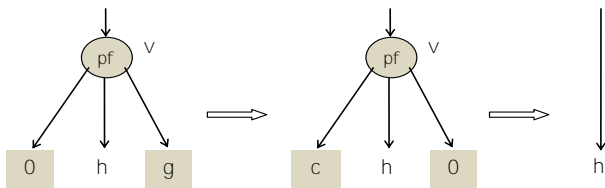


Рисунок 7.46 – Правило редуцирования LOW_{F_0}

Правило lF_{10} . Если для вершины $v \in V$ выполняется одно из четырех условий:

- $g=high(v)=0$, $h=low(v)=1$ и $c=if(v)$,
- $g=if(v)=1$, $c=low(v)=0$ и $h=high(v)$,
- $g=if(v)$, $c=high(v)$ и $h=low(v)=0$,
- $g=if(v)=0$, $c=high(v)$ и $h=low(v)$,

то вершины v , g и h исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на вершину c . Правило lF_{10} иллюстрируется рис.7.47.

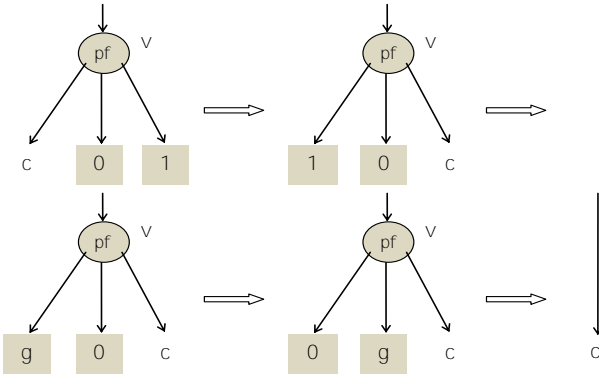


Рисунок 7.47 – Правило редуцирования lf_{01}

Правило lf_{F_1} . Если для вершины $v \in V$ выполняется одно из трех условий:

- $g = if(v)$, $h = high(v) = 1$ и $c = low(v) = 0$,
- $g = if(v) = 0$, $h = high(v) = 1$ и $c = low(v)$,
- $g = if(v) = 1$, $h = low(v) = 1$ и $c = high(v) = 0$,

то вершины v , g и c исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на терминальную вершину c меткой 1. Правило lf_{F_1} иллюстрируется рис.7.48.

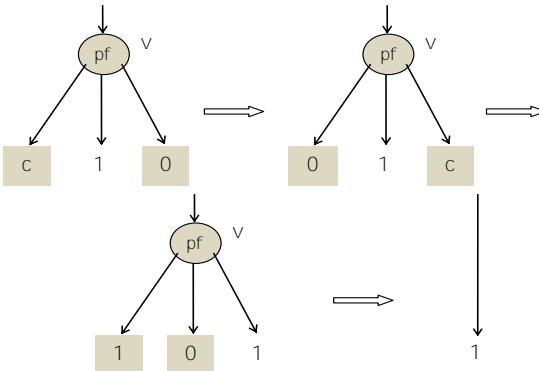


Рисунок 7.48 – Правило редуцирования lf_{F_1}

Правило lf_{low} . Если для вершины $v \in V$ выполняется $if(v) = low(v) = c$, то либо вершина $if(v)$, либо вершина $low(v)$ заменяется

терминальной вершиной с меткой 1. Правило lfF_{high} иллюстрируется рис.7.49.

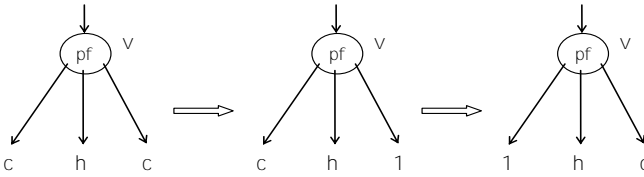


Рисунок 7.49 – Правило редуцирования lfF_{high}

Правило lfF_{high} . Если для вершины $v \in V$ выполняется $if(v) = low(v) = c$, то вершина $high(v)$ заменяется терминальной вершиной с меткой 0, а вершина $low(v)$ заменяется вершиной $not(low(v))$. Правило lfF_{low} иллюстрируется рис.7.50.

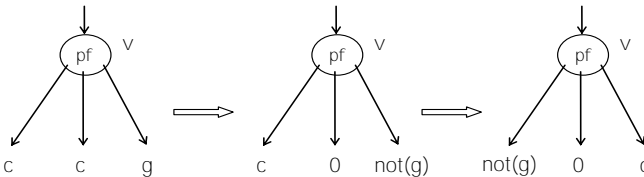


Рисунок 7.50 – Правило редуцирования lfF_{high}

Правило lfF_0 . Если для вершины $v \in V$ выполняется одно из пяти условий:

- $c = if(v) = high(v) = low(v)$,
- $if(v) = high(v) = 0$ и $c = low(v)$,
- $c = if(v)$, $low(v) = high(v) = 0$,
- $g = if(v) = 1$, $c = high(v) = low(v)$,
- $c = if(v) = high(v)$ и $g = low(v) = 1$,

то вершины v , g и c исключаются из графа диаграммы, а все входящие в вершину v дуги переориентируются на терминальную вершину с меткой 0. Правило lfF_0 иллюстрируется рис.7.51.

Функциональные if диаграммы решений с инвертированными дугами (ИФifДР). Имеются различия в выполнении операции not на терминальных и нетерминальных вершинах ФifДР. Если $v -$

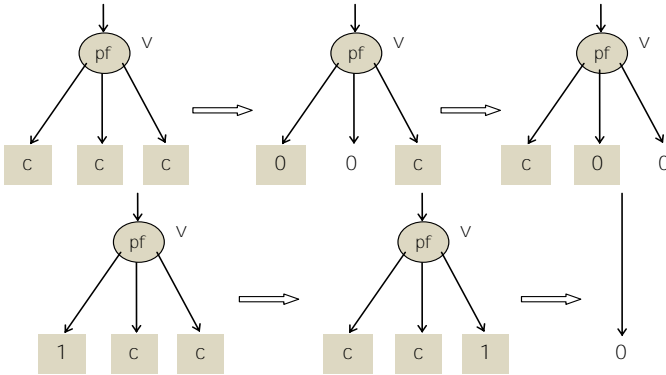


Рисунок 7.51 – Правило редуцирования lfF_0

терминальная вершина, то операция $not(v)$ выполняется точно также, как выполняется операция not на терминальной вершине ДлфДР. Пусть v – нетерминальная вершина пФлфДР, которой соответствует булева функция f . Пусть вершинам $if(v)$, $high(v)$, $low(v)$ соответствуют булевы функции c , h , g . Тогда $f = h \oplus c \wedge g$. Инвертирование функции f дает: $\neg f = \neg(h \oplus c \wedge g) = h \wedge c \wedge \neg g \vee \neg h \wedge \neg c \vee \neg h \wedge \neg g$. Построим положительное разложение Давио функции $\neg f$ по переменной c . Вычисляем кофакторы $\neg f_{c=0} = \neg h$, $\neg f_{c=1} = \neg(h \oplus g)$ и $\neg f_{c=0} \oplus \neg f_{c=1} = g$. Собирая вместе части разложения, получаем: $\neg f = \neg h \oplus c \wedge g$. Таким образом, операция $not(v)$ генерирует нетерминальную вершину v' , для которой $if(v') = if(v)$, $low(v') = low(v)$ и $high(v') = not(high(v))$. Операция инвертирования нетерминальной вершины пФлфДР иллюстрируется рис.7.52.

Следующее правило вводит инвертированные дуги в пФлфДР.

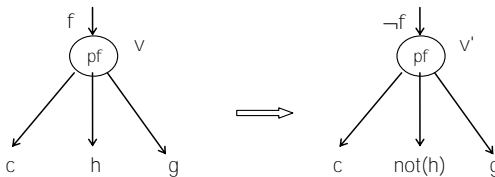


Рисунок 7.52 – Операция инвертирования нетерминальной вершины пФлфДР

Правило C_{iff} . Если для вершин $v, u \in V$ выполняется $\text{if}(u) = \text{if}(v)$, $\text{high}(u) = \text{not}(\text{high}(v))$ и $\text{low}(u) = \text{low}(v)$, то вершины u и $\text{high}(u)$ исключаются из графа диаграммы, а все дуги, входящие в вершину u , заменяются инвертированными дугами и переориентируются на вершину v .

Правило C_{iff} иллюстрируется рис.7.53. В ПФ|fДР множество типов меток терминальных вершин сокращается до двух. Метками являются переменная x_i и константа 0.

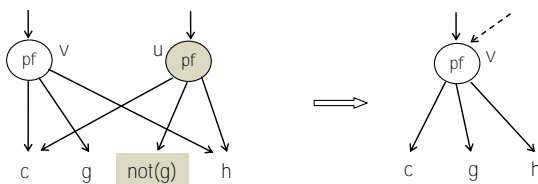


Рисунок 7.53 – Правило редуцирования C_{iff}

Частные случаи Ф|fДР образуются путем конкретизации и фиксации вида функции c в разложениях ME2F.6.76 и ME2F.6.77. Важнейшим частным случаем Ф|fДР являются функциональные диаграммы решений ФДР. Функция c в этом случае заменяется в разложениях ME2F.6.76 и ME2F.6.77 на переменную x_i . Сокращенные упорядоченные положительные и отрицательные СУФ|fДР, также как и положительные и отрицательные СУФДР, являются каноническими формами представления булевых функций.

Если в разложениях ME2F.6.76 и ME2F.6.77 использовать $c = x_i \oplus x_{i+1}$, то аналогично УД|fДР $\oplus 2$ получаем каноническую положительную и отрицательную упорядоченную УФ|fДР $\oplus 2$.

Если в разложениях ME2F.6.76 и ME2F.6.77 на уровне иерархии i использовать $c = x_i \oplus \dots \oplus x_n$, то аналогично УД|fДР \oplus получаем каноническую положительную и отрицательную упорядоченную УФ|fДР \oplus .

Пример 7.17. (продолжение примера 7.16). Построим УпФ|fДР $\oplus 2$ (рис.7.54) для булевой функции, описываемой УД|fДР $\oplus 2$, показанной на рис.7.41. Диаграмма содержит 4 нетерминальные вершины и 9 терминальных вершин. □

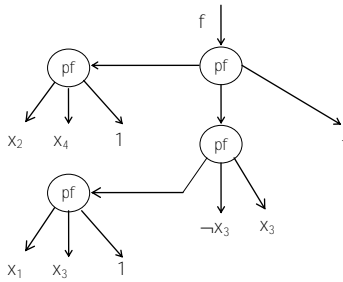


Рисунок 7.54 – Пример УпФ|fДР⊕2

7.6. Диаграммы решений для частично определенных функций

Пусть задана частично определенная функция $\varphi(x)$, зависящая от булевых переменных $x = x_1, \dots, x_n$ и принимающая значение на множестве $M = \{1, 0, dc\}$.

Представление частично определенной функции парой ДДР (Ч2ДДР). Каждая из булевых функций v и d , входящих в VCR форму $\varphi = (v|d)$ частично определенной функции φ , может быть представлена отдельной ДДР (СУДДР). ДДР v и d описывают множества φ^{on} , φ^{off} и φ^{dc} конститuent 1, 0, и dc следующим образом: $\varphi^{on} = v^{on} \cap d^{pn}$, $\varphi^{off} = v^{off} \cap d^{pn}$ и $\varphi^{dc} = d^{off}$. Достоинство Ч2ДДР заключается в гибком разбиении систем частично определенных булевых функций на части. Недостатком является некоторое усложнение алгоритмов оперирования двумя диаграммами по сравнению с одной диаграммой. Диаграммы в Ч2ДДР могут быть полными, свободными, сокращенными упорядоченными ДДР.

Пример 7.18. Пусть частично определенная функция φ задана первыми двумя столбцами табл.7.1. Представление функции φ в VDR форме дает функцию значения v , определенную третьим столбцом табл.7.1, и функцию области определенности d , определенную четвертым столбцом таблицы. Представление каждой из функций v и d в виде СУДДР дает Ч2ДДР частично определенной функции φ , показанную на рис.7.55. □

Таблица 7.1

Пример частично определенной функции ϕ

| x_1 | x_2 | x_3 | Функция ϕ | Функция ν | Функция d |
|-------|-------|-------|----------------|---------------|-------------|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | dc | 0 | 0 |
| 0 | 1 | 1 | dc | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

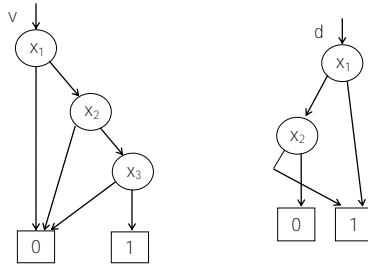


Рисунок 7.55 – Пример Ч2ДДР (пары СУДДР)

Двоичная диаграмма решений с тремя типами терминальных вершин $T=\{0, 1, dc\}$ (ЧТДДР). Она представляет частично определенную функцию ϕ и описывает не только множества ϕ^{on} и ϕ^{off} конститuent 1 и 0, но и множество ϕ^{dc} конститuent dc . Для заданного набора значений булевых переменных x_1, \dots, x_n ЧТДДР выбирает одно из значений множества M при проходе от корня графа диаграммы до терминальной вершины. С нетерминальной вершиной ν ЧТДДР ассоциируется не полностью определенная булева функция ϕ_ν . В разложении Шеннона, определяющем семантику нетерминальной вершины ЧТДДР, булевы операции инверсии, конъюнкции и дизъюнкции заменяются соответствующими частичными операциями, определенными в главе 2. Как следствие, графу диаграммы соответствует формула частичной алгебры, построенная на множестве булевых переменных x_1, \dots, x_n и множестве истинностных значений $\{0, 1, dc\}$. Она является частным случаем формулы частичной алгебры общего вида, в которой переменные принимают три значе-

ния, а не два. По сравнению с VDR формой $\varphi=(\vee d)$, достоинством ЧТДДР является интеграция всех аспектов описания функции φ в рамках одной диаграммы. Это облегчает доопределение частично определенной функции φ до некоторой булевой функции f и повышает эффективность поиска минимальной функции $f=\min(\vee d)$. Недостаток кроется в возможном увеличении размера ЧТДДР за счет интеграции в диаграмму описания множества φ^{dc} .

Интерес представляют следующие подклассы ЧТДДР:

- полная упорядоченная ПУЧТДДР;
- свободная упорядоченная СвУЧТДДР;
- сокращенная упорядоченная СУЧТДДР.

Правила I и S редуцирования ДДР (см. раздел 7.1) легко обобщаются на ЧТДДР с целью получения СУЧТДДР.

Пример 7.19 (продолжение примера 7.18). Построим ПУЧТДДР (рис.7.56) частично определенной функции φ , описанной табл.7.1. Применение правил редуцирования дает СУЧТДДР, показанную на рис.7.57. Число нетерминальных вершин сократилось с 7 до 4. □

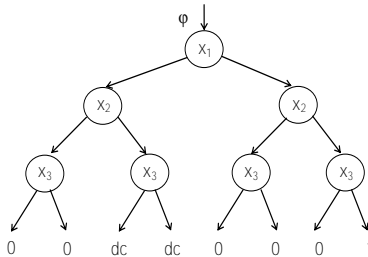


Рисунок 7.56 – Пример полной упорядоченной ПУЧТДДР

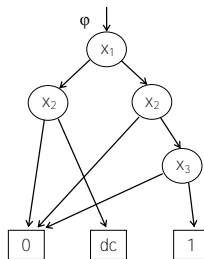


Рисунок 7.57 – Пример сокращенной упорядоченной СУЧТДДР

Троичная диаграмма решений (ТДР). В ТДР каждая нетерминальная вершина метится буквой S (что обозначает использование разложения Шеннона) и имеет три исходящие дуги. Диаграмма использует три типа терминальных вершин $T = \{0, 1, dc\}$. Она представляет частичную функцию (функцию Клини), определенную на множестве переменных, принимающих три значения. Очевидно, что ТДР является избыточной для представления частично (не полностью) определенных булевых функций. Как следствие, манипулирование ТДР приводит к перерасходу памяти и, возможно, процессорного времени. Кроме того, всегда остается актуальным вопрос, является ли частичная функция, представляемая диаграммой, корректной? Задача проверки корректности функции должна рассматриваться как при спецификации диаграммы, так и в процессе ее преобразования.

Пример 7.20. Пример ТДР приведен на рис.7.58. Диаграмма описывает частичную функцию (функцию Клини) Ψ . Частичные переменные y_1, y_2, y_3 принимают три значения 0, 1, dc. Функция также принимает три значения. Вершины диаграммы помечены переменными. Исходящие дуги помечены переменными и принимаемыми ими значениями. \square

Не полностью специфицированная диаграмма решений (НПСДР). НПСДР определяется как ориентированный ациклический граф, каждая нетерминальная вершина которого имеет хотя бы одного последователя. Другими словами, не требуется, чтобы из каждой вершины НПСДР исходили три дуги. Это свойство позволяет строить НПСДР пошагово по заданному множеству кубов

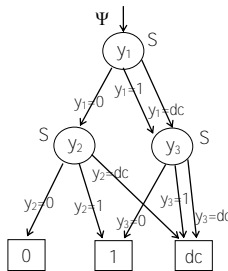


Рисунок 7.58 – Пример ТДР

булева пространства, на которых функция принимает различные значения. Задача динамической проверки корректности определяемой частичной функции актуальна и для этого типа диаграмм.

Пример 7.21. Пример НПСДР дан на рис.7.59. Эта диаграмма построена на множестве $\{y_1^1 y_2^{dc}, y_1^{dc} y_2^0\}$ кубов, на которых функция Ψ принимает значение 0, и на множестве $\{y_1^{dc} y_2^1\}$ кубов, на которых функция принимает значения 1. Но возникает вопрос, корректно ли определена функция Ψ и корректно ли построена диаграмма? \square

Двоичная if диаграмма решений на множестве из трех значений-меток $T=\{0, 1, dc\}$ (ЧТД|fДР). Она представляет частично определенную функцию ϕ и для заданного набора значений булевых переменных x_1, \dots, x_n вычисляет одно из значений множества M . По графу диаграммы для функции ϕ строится формула частичной алгебры. Алгоритм вычисления значения функции ϕ_u для вершины u ЧТД|fДР имеет следующее рекурсивное описание:

- если u есть терминальная вершина, то значение ϕ_u определяется меткой $label(u)$;
- если u есть нетерминальная вершина, то вычисляется значение $\phi_{if(u)}$; если $\phi_{if(u)}=dc$, то возвращается $\phi_u=dc$; если $\phi_{if(u)}=1$, то вычисляется значение $\phi_{high(u)}$ и возвращается $\phi_u=\phi_{high(u)}$; если $\phi_{if(u)}=0$, то вычисляется значение $\phi_{low(u)}$ и возвращается $\phi_u=\phi_{low(u)}$.

Достоинством ЧТД|fДР является интеграция всех аспектов описания функции ϕ . Важным подклассом ЧТД|fДР являются диаграммы, в которых для каждой нетерминальной вершины u подграф $if(u)$ не содержит терминальных вершин, помеченных dc .

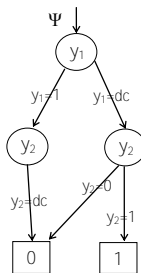


Рисунок 7.59 – Пример НПСДР

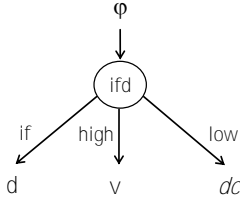


Рисунок 7.60 – Построение ЧТД|fДР по VCR форме $\varphi=(\vee d)$

Если частично определенная функция φ представлена в VCR форме $\varphi=(\vee d)$, а каждая из булевых функций v и d представлена в виде Д|fДР, то структура ЧТД|fДР, представляющей функцию φ , изображена на рис. 7.60.

Пример 7.22. (продолжение примеров 7.18-7.19). На рис.7.61 изображена ЧТД|fДР, описывающая частично определенную функцию φ , специфицированную в табл.7.1. Если СУЧТДДР, изображенная на рис.7.57, имеет четыре нетерминальные вершины, то ЧТД|fДР имеет на одну нетерминальную вершину меньше. □

Представление частично определенной функции парой Д|fДР (Ч2Д|fДР). Каждая из булевых функций v и d , входящих в VDR форму частично определенной функции φ , может быть представлена отдельной Д|fДР (СУД|fДР). Достоинство Ч2Д|fДР заключается в гибком разбиении систем частично определенных и булевых функций на части. Недостатком является некоторое усложнение алгоритмов оперирования двумя диаграммами по сравнению с одной диаграммой. Диаграммы в Ч2Д|fДР могут быть сокращенными и упорядоченными Д|fДР.

Пример 7.23 (продолжение примера 7.22). Замена значения dc на значение 0 в табл.7.1 проецирует частично определенную

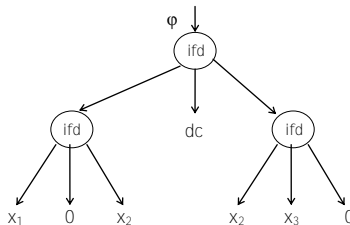


Рисунок 7.61 – Пример ЧТД|fДР

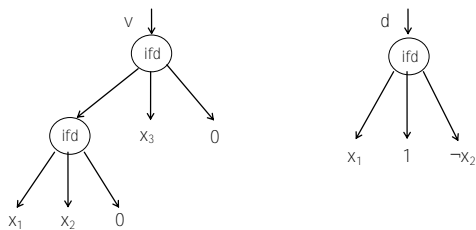


Рисунок 7.62 – Пример Ч2Д|fДР

функцию ϕ в один из вариантов булевой функции v . Замена значений 0 и 1 на значение 1 и замена значения d на значение 0 в табл.7.1, проецирует функцию ϕ в функцию области определенности d . Функции v и d представляются парой Д|fДР, изображенной на рис.7.62.

Частично определенные двоичные диаграммы СУЧТДДР и ЧТД|fДР с тремя типами терминальных вершин могут быть расширены до функциональных диаграмм. Аналогично, частично определенная функция может представляться парой полностью определенных функциональных диаграмм.

7.7. Историческая справка

Двоичные диаграммы решений как структуры данных были введены К. Ли [39 (1959)] и популяризированы Б. Акерсом [1 (1978)]. Настоящая значимость диаграмм была осознана после публикации Р. Браянта [13 (1986)], который ввел сокращенные упорядоченные двоичные диаграммы решений СУДДР. После этой публикации количество работ по ДДР стремительно возросло.

Положительные и отрицательные функциональные диаграммы решений были предложены в работе [36 (1992)]. Функциональные диаграммы решений исследованы в работе [25 (1994)]. Функциональные диаграммы решений Кронекер исследованы в работе [26 (1994)].

С момента появления оператора *if-then-else* и операции $c?a:b$ в высокоуровневых языках программирования, эти средства использовались также для описания логических систем. В работе [35 (1988)] оператор *if-then-else* достаточно детально исследовался для

представления полностью определенных булевых функций. Более того, был рассмотрен переход от СУДДР к *if-then-else* представлению, при этом предложено понятие канонической формы, основанное на двухвершинном сечении СУДДР. В работе [35] были обсуждены также вопросы учета неопределенности при построении *if-then-else* диаграмм. Ограничения предложенного класса диаграмм обусловлены тем, что семантика диаграмм не была исследована в данной работе.

Фундаментальные основы *if-then-else* диаграмм исследованы в работе А.А. Прихожего [53 (1997)], где предложены обобщения разложения Шеннона и разложений Давио для частично определенных логических функций. На самом деле, эти разложения были предложены в работе А.А. Прихожего [94 (1991)] применительно к *if-then-else* оператору, используемому в частично определенных логических алгоритмах. Семантическое обоснование *if-then-else* диаграмм открыло широкие возможности по построению и манипулированию диаграммами этого класса при решении задач моделирования и проектирования логических систем. В работе [53] впервые предложено понятие функциональных *if* диаграмм решений. Благодаря четко определенной семантике стало возможным построение ряда канонических форм, отличающихся от СУДДР.

Частично определенные функции описывались СУДДР, начиная с работ [13, 32] и др. Троичные диаграммы решений описаны в работе [23], но они являются избыточными, требующими огромных вычислительных ресурсов в процессе оптимизации. Перспективными являются частичные двоичные и функциональные *if* диаграммы решений, рассмотренные в данной книге.

Глава 8. ОПЕРАЦИИ НАД ДИАГРАММАМИ РЕШЕНИЙ

8.1. Булевы операции над двоичными диаграммами решений

Любая булева функция $f: B^n \rightarrow B$, определенная на множестве переменных $X = \{x_1, \dots, x_n\}$ может быть представлена в виде ДДР и затем преобразована к СУДДР. Одноместные булевы функции очевидным образом представляются в виде СУДДР (рис.8.1).

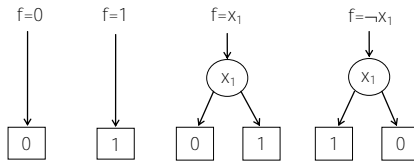


Рисунок 8.1 – Представление одноместных булевых функций в виде СУДДР

Из 16 двухместных булевых функций 2 являются по существу нульместными и 4 являются одноместными. Оставшиеся 10 функций делятся на две равные части – функции одной части являются инверсиями функций другой части. Пять основных булевых функций, представленных в виде СУДДР, изображены на рис.8.2.

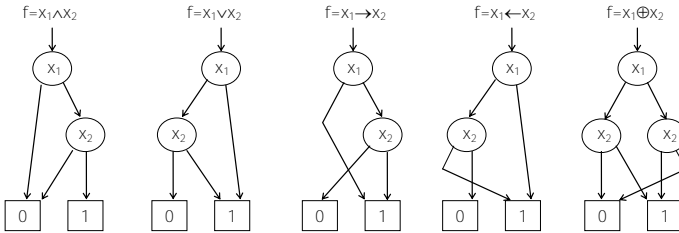


Рисунок 8.2 – Представление двухместных булевых функций в виде СУДДР

Пусть булева функция $f = g \diamond h$ определяется посредством двухместной булевой операции \diamond над булевыми функциями g и h , представленными в виде одинаково упорядоченных ДДР, и пусть необходимо построить для функции f ДДР, имеющую тот же порядок

булевых переменных, что и ДДР для функций g и h . Очевидно, что воспользоваться с этой целью СУДДР, изображенными на рис.8.2, не представляется возможным.

Построение ДДР функции $f = g \diamond h$ выполняется алгоритмом, описываемым рекурсивной функцией $Doper(p, g, h)$, псевдокод которой дан на рис.8.3. Здесь $p \in \{\wedge, \neg, \rightarrow, \leftarrow, \oplus, \vee, \downarrow, \leftrightarrow, \leftarrow, \rightarrow, / \}$ – символ двухместной булевой операции; g, h – УДДР или СУДДР одноименных функций.

```

Doper(p, g, h) begin
  if g или h есть терминальная вершина then
    return Terminal(p,g,h);
  end if;
  if g и h – обе нетерминальные вершины then
     $x_i := index(g)$ ;
     $x_j := index(h)$ ;
    if  $x_i = x_j$  then
      return ReduceS(node( $x_i$ , Doper(p, low(g), low(h)),
        Doper(p, high(g), high(h))));
    else if  $x_i < x_j$  then
      Doper(p, g, node( $x_i$ , h, h));
    else if  $x_i > x_j$  then
      Doper(p, node( $x_j$ , g, g), h);
    end if;
  end if;
end;

```

Рисунок 8.3 – Алгоритм выполнения двухместных булевых операций над УДДР

Таблица 8.1

Реализация функции Terminal

| № | Операция | Условия терминальности | | | |
|----|--|------------------------|----------|----------|----------|
| | | $g=0$ | $g=1$ | $h=0$ | $h=1$ |
| 1 | Конъюнкция $g \wedge h$ | 0 | h | 0 | g |
| 2 | Отрицание импликации $\neg(g \rightarrow h)$ | 0 | $not(h)$ | g | 0 |
| 3 | Отрицание импликации $\neg(g \leftarrow h)$ | h | 0 | 0 | $not(g)$ |
| 4 | Исключающее или $g \oplus h$ | h | $not(h)$ | g | $not(g)$ |
| 5 | Дизъюнкция $g \vee h$ | h | 1 | g | 1 |
| 6 | Стрелка Пирса $g \downarrow h$ | $not(h)$ | 0 | $not(g)$ | 0 |
| 7 | Эквивалентность $g \leftrightarrow h$ | $not(h)$ | h | $not(g)$ | g |
| 8 | Импликация $g \leftarrow h$ | $not(h)$ | 1 | 1 | g |
| 9 | Импликация $g \rightarrow h$ | 1 | h | $not(g)$ | 1 |
| 10 | Штрих Шеффера g / h | 1 | $not(h)$ | 1 | $not(g)$ |

Функция $Terminal(p,g,h)$ вызывается тогда, когда хотя бы одна из диаграмм g и h является терминальной вершиной. Поведение функции описывается табл.8.1. Функция выбирает возвращаемую диаграмму, стоящую на пересечении строки, определяемой параметром p , и столбца, определяемого условиями терминальности диаграмм g и h . Например, для $p=\oplus$ и $g=1$ функция $Terminal$ возвращает $not(h)$, т.е. инвертированную диаграмму h .

Если обе диаграммы g и h являются нетерминальными вершинами, помеченными переменными x_i и x_j соответственно, то возможны три случая выполнения операции p над этими вершинами. Если $x_i=x_j$, то операция p выполняется отдельно над нижними поддиаграммами $low(g)$, $low(h)$ и над верхними поддиаграммами $high(g)$, $high(h)$. Из полученных результатов с помощью функции $node$ конструируется новая вершина результирующей ДДР, которая метится переменной x_i . К построенной вершине применяется функция редуцирования $ReduceS$, которая реализует правило S, описанное в разделе 7.1. Если переменная x_i предшествует переменной x_j , то посредством конструирования промежуточной вершины диаграмма h расширяется до $node(x_i,h,h)$. Расширенная диаграмма замещает диаграмму h в рекурсивном вызове функции $Doper$. Если x_j предшествует x_i , в рекурсивном вызове функции $Doper$ диаграмма g замещается расширенной диаграммой $node(x_j,g,g)$. Начальный вызов функции $Doper$ имеет вид $Reduce1(Doper(p,g,h))$, где функция $Reduce1$ сокращает ДДР, построенную функцией $Doper$, посредством применения правила редуцирования I (см. раздел 7.1).

8.2. Оптимальное упорядочение переменных в сокращенной упорядоченной двоичной диаграмме решений

Вычисление оптимального порядка следования переменных в СУДДР является NP полной задачей. Для ее решения разработаны точные и эвристические алгоритмы. Лучшие точные алгоритмы имеют экспоненциальный рост вычислительной сложности и приемлемы для оптимизации СУДДР лишь небольшого размера. Эвристические алгоритмы делятся на алгоритмы, учитывающие особенности решаемой задачи, и алгоритмы динамического упорядочения переменных. Базовой операцией большинства алгоритмов оптимизации является операция обмена соседних переменных в диаграмме.

Обмен соседних переменных. Операция обмена переменных x_i и x_j выполняется локально на графе СУДДР. Пусть для вершины v имеет место: $index(v)=x_i$, $u=low(v)$, $w=high(v)$, $index(u)=index(w)=x_j$. Тогда операция обмена вводит вместо вершин v , u и w следующий новый фрагмент диаграммы: $node(x_j, node(x_i, low(u), low(w)), node(x_i, high(u), high(w)))$, где $node$ – функция, введенная в разд.8.1. Этот фрагмент группирует нижние $low(u)$, $low(w)$ и верхние $high(u)$, $high(w)$ подграфы вершины v .

Генетический алгоритм оптимального упорядочения переменных. Множество переменных X интерпретируется как *генотип*. Перестановка π переменных из X представляется строкой целых чисел длиной n , рассматриваемой как *индивидуум*. В индивидууме, $\pi(i)$ есть переменная, находящаяся на i -ой позиции. Множество индивидуумов есть *популяция*, эволюционирующая во времени. С каждым индивидуумом π ассоциируется *функция полезности* $\chi(\pi)$, представляющая размер диаграммы π -ДДР. Индивидуумы могут репродуцировать себя путем *рекомбинации* генотипов. Индивидуумы выбираются для рекомбинации случайным образом с учетом функции полезности. Предпочтение отдается индивидуумам с меньшим значением $\chi(\pi)$. Рекомбинация выполняется *скрещиванием* индивидуумов-хромосом, когда хромосомы разрезаются на две или более части каждая и затем обмениваются частями. Новые индивидуумы обновляют популяцию. Для избегания стагнации эволюции популяции используется *мутация* индивидуумов. Для придания устойчивости популяции, лучшие старые индивидуумы

```

GenAlgOrdering() begin
    Сгенерировать начальную случайную популяцию;
    Инициализировать популяцию;
    Оптимизировать популяцию методом просеивания;
do
    Применить операции репродуцирования,
        скрещивания, мутации;
    Оценить функцию полезности;
    Обновить популяцию;
while(Не выполняются условия завершения);
    Выполнить просеивание лучшего индивидуума;
return Лучший индивидуум;
end;
```

Рисунок 8.4 – Генетический алгоритм оптимизации порядка переменных в ДДР

априори включаются в новую популяцию. Псевдокод функции *GenAlgOrdering*, описывающей на абстрактном уровне работу генетического алгоритма, дан на рис.8.4.

Генерация *начальной популяции* выполняется с использованием датчика случайных чисел. Индивидуум $d=(1,2,\dots,n)$, в котором номера переменных следуют в возрастающем порядке, сортируется одновременно с сортировкой вектора $r=(r_1,r_2,\dots,r_n)$ случайных чисел, сгенерированных датчиком по равномерному закону распределения. Полученная перестановка π включается в начальную популяцию P_0 , размер которой варьируется от 40 до 120 индивидуумов.

Начальная популяция *инициализируется* путем генерации π -ДДР для каждого индивидуума-перестановки $\pi \in P_0$ и оценки значения функция полезности $\chi(\pi)$. Затем на каждой сгенерированной π -ДДР выполняется *операция просеивания* переменных, смысл которой заключается в следующем. Выбирается переменная x_i , для которой число вершин в множестве $level(x_i)$ является максимальным. Для переменной x_i в перестановке π ищется новая позиция в диапазоне $1, \dots, n$, дающая новую перестановку π' и диаграмму π' -ДДР, для которых функция полезности $\chi(\pi')$ имеет минимальное значение. Процесс выбора переменных повторяется до тех пор, пока размер диаграммы уменьшается.

Эволюция популяции происходит пошагово в следующем цикле. Пусть на i -ом шаге цикла сгенерирована популяция P_i . Из популяции выбираются случайным образом индивидуумы, к которым применяются операции *репродукции*, *скрещивания* и *мутации*. Репродуцирование копирует индивидуумы без их изменения.

Для хромосом-перестановок применяется *частично подходящее скрещивание*, которое заключается в следующем. Каждая из двух хромосом-родителей ρ_1, ρ_2 разрезается в двух точках на части, как показано на рис.8.5. Хромосомы-дети c_1, c_2 сохраняют средние части и обмениваются крайними частями родителей, при этом полученные перестановки оказываются некорректными: в первую дважды входят переменные 4, 5, во вторую дважды входят переменные 2, 6. Некорректности устраняются путем сохранения детьми своих частей неизменными и сохранения позиций и порядка как

p1: 203 | 45 | 16

c1: 531 | 45 | 04

e1: 631 | 45 | 02

p2: 531 | 26 | 04

c2: 203 | 26 | 16

e2: 403 | 26 | 15

Рисунок 8.5 – Частично подходящее скрещивание двух хромосом

можно большего числа переменных в чужих частях. Так в хромосоме c_1 переменная 5 слева заменяется переменной 6, а переменная 4 справа заменяется переменной 2. В результате получается корректная хромосома e_1 . Аналогично, в хромосоме c_2 переменная 2 слева заменяется переменной 4, а переменная 6 справа заменяется переменной 5. В результате получается корректная хромосома e_2 .

С целью *мутации* в случайно выбранной хромосоме p случайно выбирается одна позиция i . Далее, случайно выбирается номер k переменной и находится позиция j этой переменной в хромосоме p : $p_j = k$. В новой хромосоме c переменные в позициях i и j меняются местами: $c_j = p_i$ и $c_i = p_j$. Для остальных позиций l выполняется равенство $c_l = p_l$. Операция мутации может применяться дважды к одной хромосоме.

При переходе от популяции P_i к популяции P_{i+1} лучшая половина популяции P_i переносится в популяцию P_{i+1} без изменений. Хромосомы из другой половины скрещиваются и мутируют с заданной вероятностью. Новые хромосомы формируют вторую часть популяции P_{i+1} , размер которой не изменяется.

Результаты проведенных *экспериментальных* исследований показывают, что в 90% случаев генетический алгоритм дает оптимальные результаты, в остальных случаях отклонение от оптимального результата составляет не более 3%.

8.3. Операция минимизации частично определенных двоичных диаграмм решений

Частично определенная функция $f(x)$ описывается тремя множествами: f^{on} , f^{off} и f^{dc} . Доопределение функции f до булевой функции f выполняется разбиением множества f^{dc} на две части: f_0^{dc} и f_1^{dc} . Первая часть объединяется с множеством f^{off} , при этом $f^{off} = f^{off} \cup f_0^{dc}$. Вторая часть объединяется с f^{on} , при этом $f^{on} = f^{on} \cup f_1^{dc}$.

Всего существует 2^δ разбиений, где $\delta = |\varphi^{dc}|$. Существует такое же количество доопределений значения dc до значений 0 или 1 на различных наборах значений аргументов.

Пусть функция φ описывается одноименной сокращенной упорядоченной двоичной диаграммой решений с тремя типами терминальных вершин СУЧТДДР (см. раздел 7.6). Ставится задача доопределения *без изменения порядка следования переменных* диаграммы φ до СУДДР f , имеющей минимальный размер среди всех возможных доопределений φ . Доказано, что задача является NP полной.

Метод конструирования изоморфных подграфов. Воспользуемся методом конструирования изоморфных подграфов в графе диаграммы φ путем замены терминальной вершины dc терминальными вершинами 0 и 1 или целыми подграфами. Будем различать два местоположения изоморфных подграфов:

- изоморфными являются дочерние подграфы $low(v)$ и $high(v)$ одной родительской вершины v ,
- изоморфными являются подграфы разных родительских вершин.

В первом случае для сокращения может быть применено правило редуцирования S , при этом размер результирующей диаграммы ψ оценивается выражением $size(\psi) = size(\varphi) - size(low(v)) - 1$. Во втором случае для сокращения может быть применено правило редуцирования I , при этом размер результирующей диаграммы ψ определяется выражением $size(\psi) = size(\varphi) - size(low(v))$. Приведенные соотношения справедливы тогда, когда терминальная вершина dc заменяется только вершинами 0 и 1. Если для получения изоморфизма вершина dc заменяется целыми подграфами, значение $size(\psi)$ будет больше, а эффект от сокращения несколько меньше. Еще больший эффект может дать поиск вариантов конструирования изоморфных подграфов комбинированного местоположения.

Рассмотрим процесс замены вершины dc . Прежде всего, мы должны рассматривать столько независимых копий вершины dc , сколько входных дуг имеет вершина. Независимость копий проявляется в возможности замены каждой из них своим наиболее подходящим подграфом. Выбор подграфа ограничивается тем, что его корень должен быть помечен переменной, которая должна быть по-

следователем переменной, метящей вершину, из которой исходит дуга, входящая в вершину dc . Поиск изоморфных подграфов упрощается тем, что он должен выполняться на одном уровне $level(x_i)$ вершин. Чем больше уровень, тем большее количество изоморфных подграфов можно сконструировать. Чем меньше уровень, тем больше размер конструируемых изоморфных подграфов.

Пример 8.1 (продолжение примера 7.19). В СУЧТДДР ϕ , изображенной на рис.7.57, вершина dc может быть заменена, в силу своего местоположения, четырьмя подграфами, соответствующими функциям $0, 1, x_3, -x_3$. Использование подграфов для функций 1 и $-x_3$ блокирует применение правил редуцирования. Замена вершины dc вершиной 0 (показана пунктиром) делает возможным применение правила S (рис.8.6a). В результирующей СУДДР произошло сокращение нетерминальной вершины, помеченной переменной x_2 , и терминальной вершины dc . Замена вершины dc подграфом, соответствующим функции x_3 (показан пунктиром), также делает возможным применение правила S (рис.8.6b). В результирующей СУДДР сокращены две нетерминальные вершины, помеченные переменными x_1, x_2 , и одна терминальная вершина dc . Следует

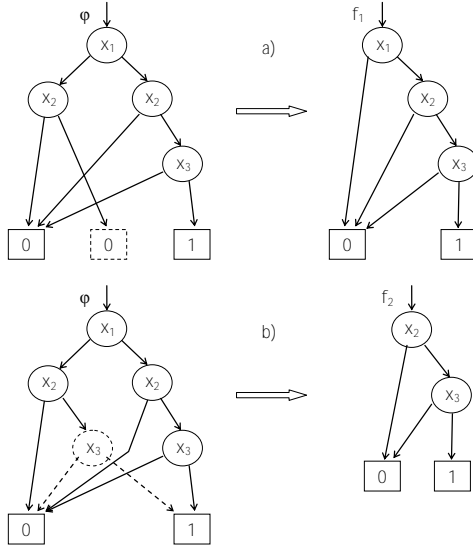


Рисунок 8.6 – Варианты доопределения СУЧТДДР ϕ до СУДДР f_1 и f_2

заметить, что булевы функции f_1 и f_2 , соответствующие сконструированным СУДДР, различаются. В то же время они являются двумя доопределениями одной частично определенной функции φ . \square

Генетический алгоритм минимизации СУДДР для частично определенной функции. Он работает с фиксированным порядком переменных в диаграмме. Индивидуумы популяции представляют подмножества φ_1^{dc} множества φ^{dc} , которые добавляются к множеству φ^{on} с целью получения из функции φ булевой функции f . Индивидуумы описываются соответствующими СУДДР. Для создания начальной популяции, подмножества φ_1^{dc} множества φ^{dc} генерируются с одинаковой вероятностью. Функция полезности $\chi(\varphi_1^{dc})$ описывает размер СУДДР, представляющей функцию f , для которой $\varphi^{on} = \varphi^{on} \cup \varphi_1^{dc}$. Потомки генерируются посредством генетических операций. Для выбора родителей используется правило рулетки. Новые потомки замещают худшие индивидуумы в текущей популяции.

Операция рекомбинации X_AND (аналог скрещивания) принимает на входе два индивидуума-функции g и h , для которых $g^{on}, h^{on} \subseteq \varphi^{dc}$, и создает одного потомка $b = g \wedge h$. Очевидно, что $b^{on} \subseteq \varphi^{dc}$. Существенным является то, что операция \wedge выполняется эффективно над двумя СУДДР, представляющими функции g и h .

Операция мутации принимает одного индивидуума-функцию g и создает одного потомка b . Она реализуется в двух вариантах. В варианте M_OR случайным образом выбирается набор $a \in \varphi^{dc} \setminus g^{on}$ и строится минтерм m_a , а потомок $b = m_a \vee g$. В варианте M_EXOR случайным образом выбирается набор $a \in g^{on}$ и строится минтерм m_a , а потомок $b = m_a \oplus g$. Оба варианта генерируют корректные потомки. Операции \vee и \oplus эффективно выполняется над СУДДР m_a и g .

Псевдокод функции *GenAlgAssignment*, описывающей работу генетического алгоритма, дан на рис.8.7. Алгоритм поддерживает популяцию с числом индивидуумов, в два раза превышающим число переменных. Алгоритм завершает работу, если не произошло улучшения популяции на 200-ых итерациях. Генетическая операция скрещивания X_AND выбирается с вероятностью 0.2, операции мутации M_OR и M_EXOR выбираются с вероятностью 0.4 каждая.

Приведем результаты вычислительных экспериментов на 15 не полностью определенных проектах. По каждому проекту дано


```

GenAlgAssignment () begin
    Популяция := Сгенерировать_начальную_популяцию;
    Вычислить_функцию_полезности(Популяция);
    do
        Операция := Выбрать_генетическую_операцию( X_AND,
            M_OR, M_EXOR);
        Родители := Выбрать_родителей(Популяция);
        Потомок := Выполнить_операцию(Операция, Родители);
        Вычислить_функцию_полезности(Потомок);
        Обновить_популяцию(Популяция, Потомок);
    while(Не выполняются условия завершения);
    return Лучший индивидуум;
end;

```

Рисунок 8.7 – Генетический алгоритм минимизации не полностью определенной булевой функции

сравнение трех альтернативных вариантов: (1) $f^{pn} = \varphi^{pn}$, (2) $f^{pn} = \varphi^{pn} \cup \varphi^{dc}$ и (3) решение, найденное генетическим алгоритмом. В 33.3% случаев варианты (1) и (3) дали одинаковые результаты. В среднем вариант (3) превосходит вариант (1) на 3.1%. Вариант (2) уступает варианту (3) в среднем на 23.4%. Незначительное сокращение (только 3.1%) сложности проектов генетическим алгоритмом может быть объяснено двумя причинами: характер присутствующей в них неопределенности объективно не приводит к существенному сокращению числа вершин СУДДР; данный генетический алгоритм не всегда приводит к нахождению глобального оптимума.

Метод конструирования изоморфных подграфов и генетический алгоритм применимы также для случая представления функции f двумя СУДДР v и d . Поиск изоморфных подграфов выполняется в процессе параллельного обхода графов двух диаграмм. При таком представлении функции метод реализует операцию минимизации $f = \min(v|d)$. Эвристическая версия метода конструирования изоморфных подграфов с целью применения правил редуцирования состоит в следующем. Рассматриваются четыре основных случая в выполнении операции $\min(v|d)$, иллюстрируемые рис.8.8. Если ДДР v есть лист 1, результатом минимизации является ДДР g . Если корни ДДР v и d помечены одной переменной x_i и $high(d)=0$, то результатом является подграф $low(v)$, минимизированный по подграфу $low(d)$. Аналогичный результат получаем при $low(d)=0$. Если $high(d) \neq low(d) \neq 0$, то результат операции конструируется из левой и

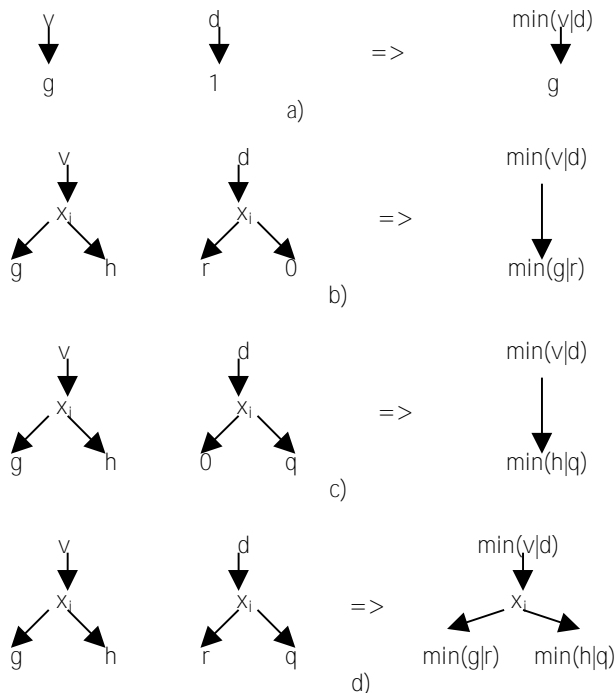


Рисунок 8.8 – Операция минимизации над СУДПР

правой частей ДДР v , минимизированных по левой и правой частям ДДР d .

8.4. Булевы операции над *if* диаграммами решений

Поскольку двоичные *if* диаграммы решений в общем виде не являются канонической формой представления булевых функций, одноместные и двухместные булевы операции над Д|fДР дают Д|fДР, совпадающую по своей структуре с Д|fДР, представляющей соответствующие булевы функции. Возникающая избыточность результирующей диаграммы устраняется посредством использования разложений, предложенных в разделе 6.4.

Как установлено в разделе 7.3, операция инвертирования Д|fДР, корневой вершине которой соответствует функция

$f = c \wedge g \vee \neg c \wedge h$, сводится к инвертированию поддиаграмм, которым соответствуют функции g и h , поскольку $\neg f = c \wedge \neg g \vee \neg c \wedge \neg h$. Заметим, что операция инвертирования ДДР, сводящаяся к взаимной замене терминальных вершин 0 и 1, реализуется иначе.

Из десяти полноценных двухместных булевых операций пять основных представлены в виде Д|fДР, изображенных на рис.8.9. Оставшиеся пять операций представляются Д|fДР, являющимися инверсиями пяти первых Д|fДР. Все построенные диаграммы являются избыточными и могут быть минимизированы. Для минимизации диаграмм используются разложения МЕО.6.35-МЕО.6.42. Минимизированные неизбыточные диаграммы для четырех основных двухместных операций (операция \oplus не входит в их число) показаны на рис.8.10. Для операции \oplus устранить избыточность подобным образом не удастся. Конкретизируем соотношение (6.25) до

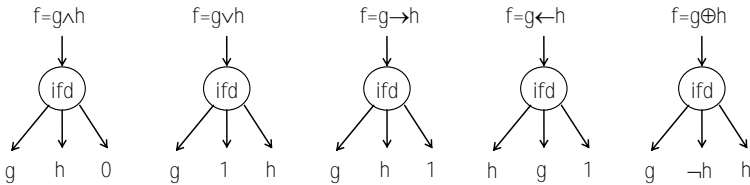


Рисунок 8.9 – Представление двухместных булевых операций в виде Д|fДР

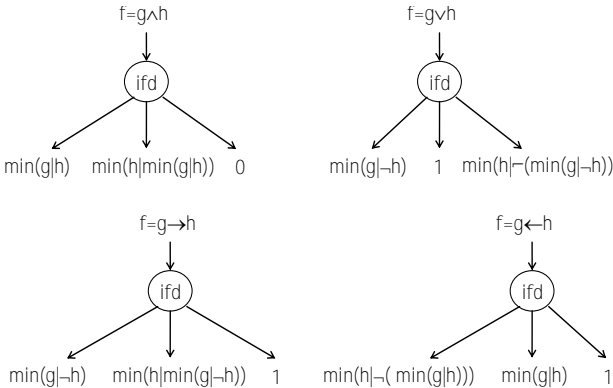


Рисунок 8.10 – Неизбыточные Д|fДР для двухместных булевых операций

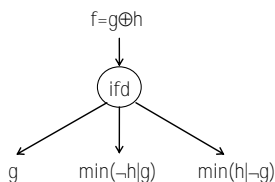


Рисунок 8.11 – Не избыточная Д|fДР для операции \oplus

равенства $g \oplus h = g \wedge \min(-h|g) \vee -g \wedge \min(h|-g)$, позволяющего построить Д|fДР, изображенную на рис.8.11. Применение самого равенства (6.25) даст граф на трех вершинах *ifd*, представляющих одну операцию дизъюнкции и две операции конъюнкции.

Таким образом, устранение избыточности диаграмм, получаемых при выполнении булевых операций, базируется на использовании операции минимизации $\min(\vee|d)$ булевой функции \vee по булевой функции d .

8.5. *Операция минимизации частично определенных if диаграмм решений*

Пусть частично определенная функция ϕ описывается одноименной двоичной *if* диаграммой решений с тремя типами терминальных вершин ЧТД|fДР (см. раздел 7.6). Ставится задача *доопределения* диаграммы ϕ до Д|fДР f , имеющей минимальный размер среди всех возможных доопределений ϕ . Напомним, что для СУДДР задача является NP трудной, при этом СУДДР является частным случаем Д|fДР, а Д|fДР расширяет множество структурных решений, генерируемых посредством СУДДР.

Метод конструирования изоморфных подграфов может быть использован и в этом случае, однако он дает не глобальный оптимум, а эвристическое решение. Терминальные вершины *dc* могут быть заменены в ЧТД|fДР произвольными подграфами. Снова будем различать два местоположения изоморфных подграфов:

- изоморфными являются дочерние подграфы одной родительской вершины \vee ,
- изоморфными являются подграфы разных родительских вершин.

Первое местоположение предпочтительнее тем, что здесь размер последующих сокращений больше чем во втором местоположении. Еще больший эффект дает поиск вариантов конструирования изоморфных подграфов комбинированного местоположения.

Каждая копия вершины dc заменяется своим наиболее подходящим подграфом. Шансы на конструирование изоморфных подграфов тем выше, чем меньше размер подграфов. Шансы уменьшаются с увеличением размера.

Пример 8.2. (продолжение примера 7.22). В ЧТД|fДР ϕ , изображенной на рис.7.61, вершину dc целесообразно заменить правым нижним подграфом диаграммы ϕ . В этом случае может быть использовано правило редуцирования S_{if} (см. раздел 7.3). Оно дает сокращенную Д|fДР, показанную на рис.8.12. Размер сокращений составил две нетерминальные вершины ifd и четыре терминальные вершины. \square

Генетический алгоритм минимизации СУДДР для частично определенной функции может быть расширен и адаптирован к минимизации Д|fДР. Адаптация выполняется путем построения для каждого индивидуума популяции не только СУДДР, но и Д|fДР. Функция полезности описывает уже не размер СУДДР, а размер Д|fДР. Ключевой операцией в этом алгоритме является генерация минимальной Д|fДР по СУДДР.

Метод конструирования изоморфных подграфов и генетический алгоритм применимы также для случая представления функции ϕ двумя Д|fДР v и d . В этом случае они реализуют операцию $min(v|d)$. Поиск изоморфных подграфов выполняется в процессе параллельного обхода графов двух диаграмм. Предложена эвристическая версия метода конструирования изоморфных подграфов с

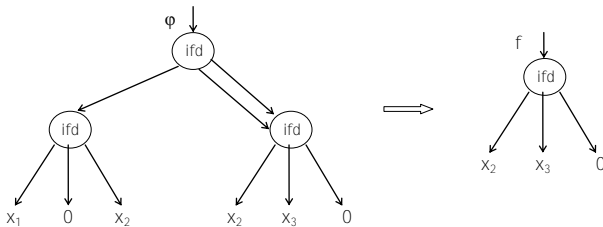


Рисунок 8.12 – Операция минимизации ЧТД|fДР

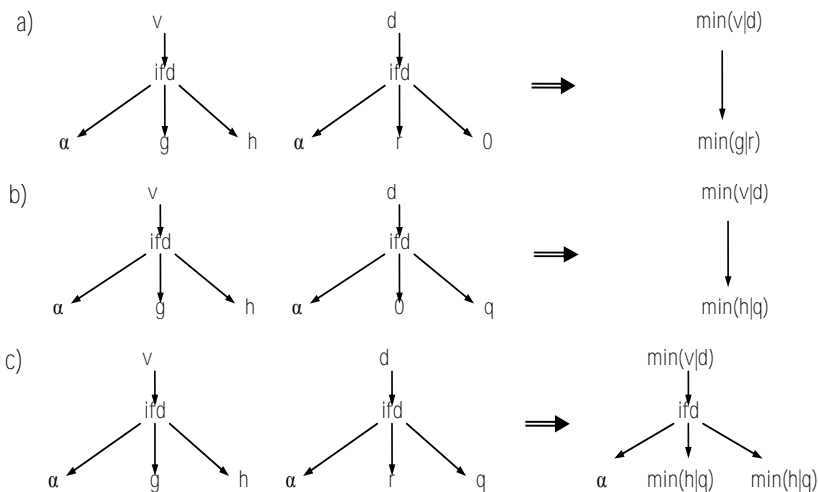


Рисунок 8.13 – Операция минимизации над ДИФДР v и d при $if(v) = if(d)$

целью применения правил редуцирования ДИФДР. На рис 8.13. показаны основные случаи в выполнении операции $min(v|d)$ при равенстве $if(v) = if(d) = \alpha$.

Если $low(d) = 0$, то результатом является граф $high(v)$, минимизированный по графу $high(d)$. Аналогичный результат получаем при $high(d) = 0$. Если $high(d) \neq low(d) \neq 0$, то результат операции конструируется из α , а также из $high(v)$ и $low(v)$, минимизированных по $high(d)$ и $low(d)$ соответственно.

В случае, когда одна из функций v или d представлена диаграммой $ifd(c, g, h)$, операция минимизации реализуется эвристическим алгоритмом, рекурсивное описание которого дано на рис.8.14. Поведение алгоритма при $d = ifd(c, g, h)$ определяется равенством

$$min(v|d) = c \wedge min(min(v|c)|g) \vee \neg c \wedge min(min(v|\neg c)|h), \quad (8.1)$$

в котором v минимизируется дважды: в первом операнде по c и g , во втором операнде по $\neg c$ и h . Поведение алгоритма при $v = ifd(c, g, h)$ определяется равенством

$$min(f|d) = min(c|d) \wedge min(g|d \wedge min(c|d)) \vee \neg min(c|d) \wedge min(h|d \wedge \neg min(c|d)). \quad (8.2)$$

$$\min(v/d) = \begin{cases} \text{произвольная ДлфДР} & \text{если } d=0 \\ 1 & \text{если } v=d \\ 0 & \text{если } v=-d \\ \text{длфДР } d=\text{ifd}(c,g,h) & \text{длфДР } d=\text{ifd}(c,g,h) \\ \min(\min(v|-c)|h) & \text{если } g=0 \\ \min(\min(v|c)|g) & \text{если } h=0 \\ \text{reduce}(\text{ifd}(c, & \text{иначе} \\ \min(\min(v|c)|g), & \\ \min(\min(v|-c)|h))) & \end{cases}$$

$$\begin{cases} \min(h|d) & \text{длфДР } v=\text{ifd}(c,g,h) \\ \min(g|d) & \text{если } \min(c|d)=0 \\ \min(h|d \wedge \neg \min(c|d)) & \text{если } \min(c|d)=1 \\ \min(g|d \wedge \min(c|d)) & \text{если } d \wedge \min(c|d)=0 \\ \text{reduce}(\text{ifd}(\min(c|d), & \text{иначе} \\ \min(g|d \wedge \min(c|d)), & \\ \min(h|d \wedge \neg \min(c|d))) & \text{иначе} \end{cases}$$

Рисунок 8.14 – Операция минимизации над ДлфДР

в котором g минимизируется по $\min(c|d)$ и d , а h минимизируется по $\neg \min(c|d)$ и d . Функция *reduce* сокращает результирующую диаграмму, однако после сокращения диаграмма все равно может быть не оптимальной.

В силу того, что ДлфДР общего вида не является канонической формой, выполнение логических операций в процессе ее минимизации может оказаться неэффективным. Более эффективным является построение и минимизация ДлфДР в процессе перехода к ней от СУДДР, являющейся канонической формой.

8.6. Преобразование двоичной диаграммы решений в if диаграмму

Если функция $f(x)$ существенно зависит от n переменных, она представляется СУДДР, имеющей глубину n . Этой СУДДР соответствует эквивалентная ДлфДР, имеющая глубину $n-1$.

Сокращение размера ДлфДР по сравнению с размером исходной СУДДР базируется на поиске наиболее часто используемых в СУДДР подфункций функции f . Таким подфункциям соответствуют подграфы g диаграммы, имеющие наибольшее количество входя-

щих дуг. Для построения вершины ifd подграф g может использоваться двумя способами: 1) в качестве if -подграфа; 2) в качестве $high$ -подграфа. В первом случае в графе f все дуги, входящие в g , перенаправляются в вершину 1 при конструировании $high$ -подграфа и перенаправляются в вершину 0 при конструировании low -подграфа. Во втором случае для известного $high$ -подграфа конструируется подходящий if -подграф, который затем используется для вычисления low -подграфа посредством операции минимизации.

Сокращение глубины Д|fДР по сравнению с глубиной исходной СУДР базируется на использовании *метода сечений*. Пусть на графе исходной СУДР существует сечение между корнем и листьями, разделяющее диаграмму на верхнюю и нижнюю части так, что все дуги, исходящие из верхней части, направлены только в две вершины нижней части, как показано на рис.8.15 слева. В этом случае СУДР разбивается на три подграфа d_1, d_2, d_3 . Построим из них смешанную Д|fДР-СУДР путем введения новой вершины ifd , как показано на рис.8.15 справа. Диаграмма d_1' отличается от d_1 тем, что исходящие дуги, направленные в корень диаграммы d_2 , перенаправляются в лист с меткой 1, а дуги, направленные в корень диаграммы d_3 , перенаправляются в лист с меткой 0. Результатом операции минимизации $\min(f|d_1')$ является диаграмма d_2 , результатом $\min(f|d_1')$ является диаграмма d_3 . Глубина смешанной Д|fДР-СУДР диаграммы равна $\max(dp(d_1'), dp(d_2), dp(d_3))+1$, где $dp(d)$ – глубина диаграммы d .

Сечение, построенное на k вершинах графа СУДР, требует введения $k-1$ дополнительных ifd -вершин. Глубина конструируемой

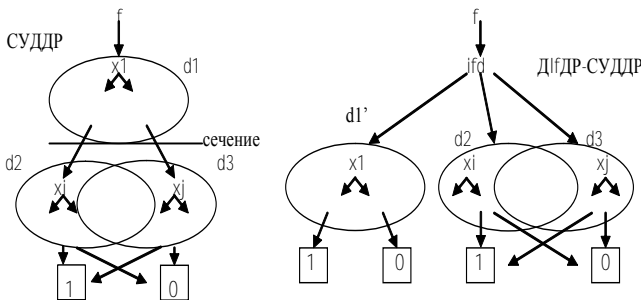


Рисунок 8.15 – Метод сечений для сокращения глубины Д|fДР

ДифДР-СУДДР не превышает при этом значения $\max(dp(d_1), \dots, dp(d_{k+1})) + \lfloor \log_2 k \rfloor$, где $\lfloor z \rfloor$ – ближайшее целое, большее либо равное z . СУДДР с сечением на трех вершинах показана на рис.8.16. Подграф d_1 имеет исходящие дуги, направленные в корни подграфов d_2 , d_3 и d_4 . Эквивалентная ДифДР-СУДДР показана на рис.8.17, где d_1' (d_1'') есть СУДДР, полученные из d_1 путем переориентации исходящих дуг в лист 1, если они входят в подграф d_2 (d_3), и в лист 0, если они входят в подграфы d_3 и d_4 (d_2 и d_4). После переориентации дуг объединенная диаграмма d_1' - d_1'' сокращается.

Пример 8.3. На рис.8.18 приведен пример СУДДР f с сечением на трех вершинах. Сконструированный *if*-подграф s показан на этом же рисунке. Исходная диаграмма f имеет размер 15 и глубину 7 нетерминальных вершин. Результирующая ДифДР-СУДДР, представленная на рис.8.19, также имеет размер 15 нетерминальных вершин, однако ее глубина сокращена до 5 единиц. □

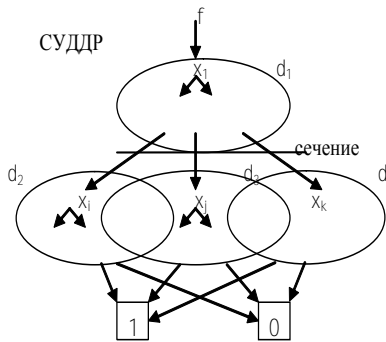


Рисунок 8.16 –СУДДР с сечением на трех вершинах

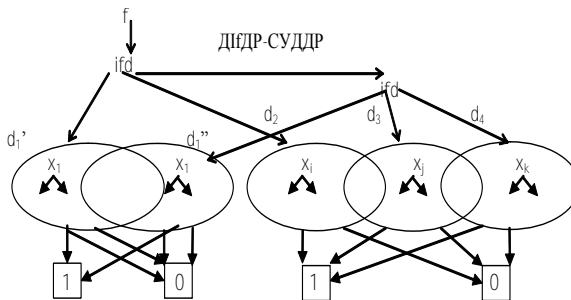


Рисунок 8.17 – Эквивалентная ДифДР-СУДДР сокращенной глубины

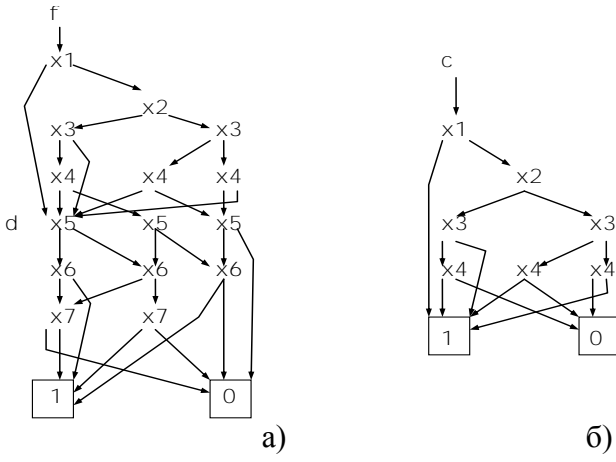


Рисунок 8.18 – Исходная СУДП f глубины 7 и сконструированный if подграф c

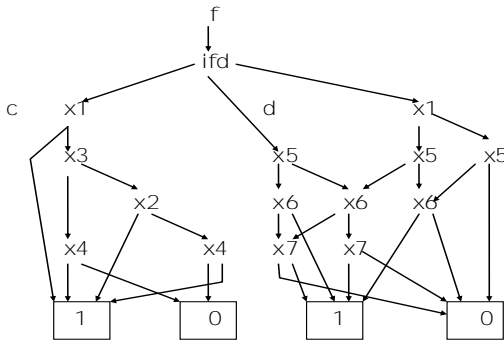


Рисунок 8.19 – Эквивалентная ДИФДР-СУДП глубины 5

8.7. Сокращение глубины if диаграммы

Пусть дана ДИФДР, описывающая функцию f и имеющая глубину $p = depth(f)$. Наша цель состоит в построении новой ДИФДР с корнем в вершине v , ссылающейся на три более простых подграфа $c = if(v)$, $high(v) = min(f|c)$ и $low(v) = min(f|-c)$, максимальная глубина которых не превышает $p-2$. В этом случае общая глубина построенной ДИФДР сокращается, по меньшей мере, на одну ifd вершину.

Сбалансированность трех подграфов по глубине может дать более значительное сокращение общей глубины.

Предлагаемый метод сокращения глубины базируется на поиске и разрезании длинных (*критических*) путей на графе исходной Д|fДР. Если Д|fДР f имеет один длинный путь глубиной ρ , то следует разрезать его таким образом, чтобы получить по возможности одинаковую глубину всех трех подграфов новой *ifd* вершины. Глубина генерируемой Д|fДР может быть сокращена до $\lfloor \rho/2 \rfloor + 1$. Это достигается выбором подходящей Д|fДР c .

Проиллюстрируем работу метода на примере Д|fДР $c7$, показанной на рис.8.20 и имеющей путь на восьми *ifd*-вершинах. Для того чтобы сгенерировать Д|fДР глубиной пять, воспользуемся Д|fДР c , показанной на рис.8.21. Она включает путь, построенный из *ifd* вершин, составляющих первую половину длинного пути исходной диаграммы. Однако *high*-подграфами этих *ifd* вершин становятся литья 0, а *low*-подграфом последней *ifd* вершины становится лист 1. Для новой вершины v подграф $high(v) = \min(f/c7)$ остается такими же, как в исходной Д|fДР. Последняя *ifd* вершина подграфа

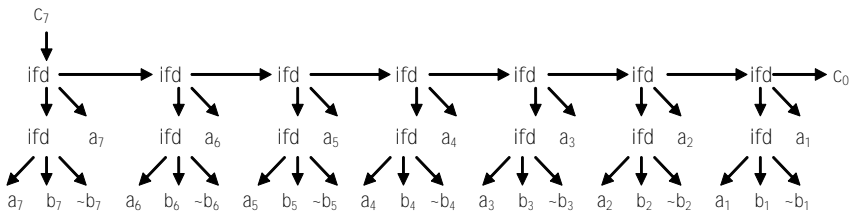


Рисунок 8.20 – Д|fДР глубиной восемь *ifd*-вершин

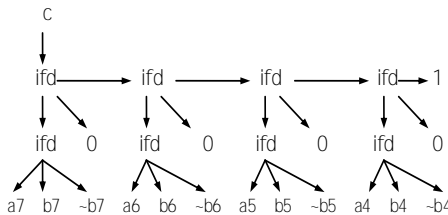


Рисунок 8.21 – Д|fДР $c = if(v)$

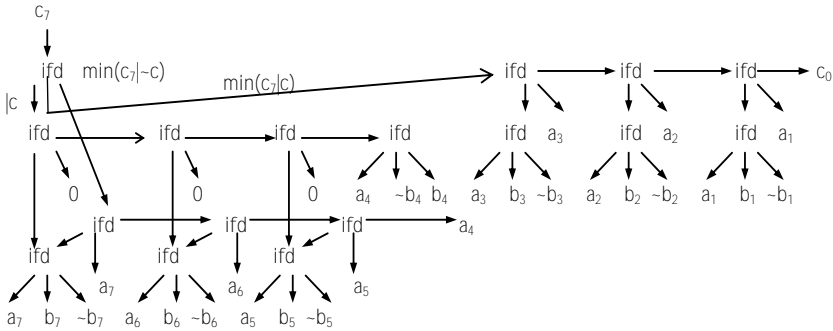


Рисунок 8.22 –Преобразованная ДfDP глубины пять *ifd* вершин

$low(v) = \min(f|\neg c7)$ модифицируется. Ею становится лист, помеченный переменной a_4 . После выполнения операций $\min(c7|c)$ и $\min(c7|\neg c)$ и слияния всех трех генерируемых частей получаем сокращенную ДfDP, показанную на рис.8.22 и имеющую глубину 5.

Предложенный метод разрезания длинных путей обобщается на случай наличия в исходной IFD нескольких длинных путей.

8.8. Историческая справка

Операции над СУДДР и алгоритмы их реализации предложены в работе Р. Браянта [13 (1986)]. Булевы операции над СУДДР являются эффективными и выполняются за полиномиальное время. Задача минимизации СУДДР путем оптимального упорядочения переменных является NP полной [8]. Для ее реализации предложены как точные, так и эвристические алгоритмы. Генетический алгоритм оптимального упорядочения переменных предложен в работе [27].

Задача доопределения не полностью определенной функции, представленной парой СУДДР, до полностью определенной булевой функции, представленной одной СУДДР минимального размера, сохраняющей порядок следования переменных, является NP полной [77]. Эвристические методы решения этой задачи предло-

жены в работах [79, 16]. Точный метод решения задачи предложен в работе [47]. Генетический алгоритм поиска оптимального упорядочения предложен в работе [24]. Эвристический метод конструирования изоморфных подграфов в СУДДР при решении задачи доопределения предложен в исследован в работах [32, 33, 53, 54]. Генетический алгоритм поиска СУДДР минимального размера, реализующей заданную не полностью определенную булеву функцию, предложен в работе [24].

Булевы операции над *if-then-else* диаграммами исследованы в публикациях [24, 53, 35]. Методы генерации *if-then-else* диаграммы путем разрезания СУДДР предложены в [54, 59, 35]. Эвристический метод минимизации *if* диаграмм исследован в работе [53]. Метод сокращения глубины *if* диаграммы путем разрезания критических путей на графе диаграммы предложен в работах [56, 59, 67, 101].

Глава 9. ПРИМЕНЕНИЕ ДИАГРАММ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ

9.1. Моделирование цифровых систем

Отображение одной *ifd* вершины, представляющей двухместную булеву операцию в ДfДР диаграмме, в логический вентиль показано на рис.9.1. Отображение зависит от конфигурации ДfДР. Диаграмма может быть отображена в мультиплексор MUX, OR-, EQV-, EXOR-, AND-, NAND-, NOR-вентиль. Многовершинная ДfДР отображается в логическую схему, построенную из вентиляей.

Положительные и отрицательные функциональные *if* диаграммы решений отображаются в вентильные схемы аналогичным образом. Сказанное относится к другим типам *if* диаграмм решений, а также к двоичным диаграммам решений ДДР.

Отображение может строиться не только на отдельных вершинах диаграммы, но и на целых фрагментах. В этом случае фрагменты отображаются как в простые, так и сложные компоненты технологического базиса.

Диаграммы решений могут отображаться в логические схемы не только прямым, но и косвенным образом. Один из способов отображения состоит в том, что сначала по минимизированной диаграмме строится двухуровневое или многоуровневое формульное представление логического проекта, затем это представление отображается в технологический базис.

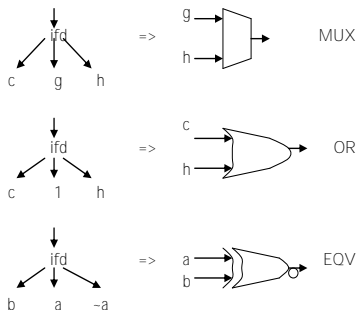


Рисунок 9.1 – Отображение *ifd* вершин в вентили

Диаграммы решений используются не только для моделирования комбинационных схем, но также и для моделирования последовательностных схем. Структурные автоматы, их функции переходов и выходов эффективно описываются двоичными и *if* диаграммами решений. Расширенные многозначные диаграммы решений позволяют моделировать абстрактные автоматы.

9.2. Распараллеливание вычислений в логических системах

Параллельные вычисления в цифровых схемах и системах моделируются и реализуются на системном, алгоритмическом, архитектурном и логическом уровнях. Известно два основных типа параллелизма: параллелизм в пространстве и параллелизм во времени (конвейерный параллелизм). В первом случае параллелизм проявляется в параллельном выполнении операций над одним набором входных данных. Во втором случае последовательно соединенные операции выполняются параллельно над разными наборами входных данных. На практике два вида параллелизма используются совместно.

Предложенные в предыдущих параграфах модели, в частности, новые типы диаграмм решений, позволяют повысить *уровень распараллеленности* цифровых систем.

Проанализируем уровень параллелизма в вычислениях, построенных на бвзе СУДДР. Вершины СУДДР могут выполняться попарно как последовательно, так и параллельно. Это описывается *матрицей распараллеленности* вершин $C_{\text{ддр}}$, в которой элемент $c_{ij}=1$, если вершины i и j выполняются взаимно последовательно, и $c_{ij}=0$, если вершины i и j выполняются взаимно параллельно. Для СУДДР, изображенной на рис.9.2, матрица распараллеленности

$$C_{\text{ддр}} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ & & 0 & 1 & 1 & 1 & 1 & 1 \\ & & & 0 & 0 & 1 & 1 & 1 \\ & & & & 0 & 1 & 1 & 1 \\ & & & & & 0 & 1 & 1 \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{vmatrix}$$

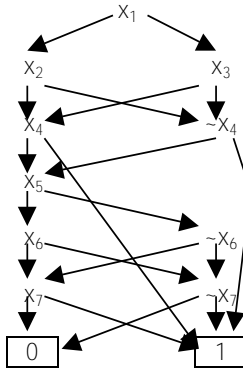


Рисунок 9.2 – В СУДДР доминирует последовательное выполнение (через ~ обозначена булева операция инвертирования)

Мы оцениваем уровень параллелизма в СУДДР через множество D пар распараллеленных вершин. Так для приведенной диаграммы $D = \{(2,3), (4,5), (7,8), (9,10)\}$. Множество D лежит в основе вычисления коэффициента распараллеленности:

$$k_c = \frac{2}{n \cdot (n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \neg c_{ij}, \quad (9.1)$$

где $\neg c_{ij}$ – инверсия элемента c_{ij} . Для рассматриваемой СУДДР формула (9.1) дает $k_c = 0.089$. Это значение говорит о низком уровне параллелизма в диаграмме. Время выполнения диаграммы можно оценить посредством множества U_{-D} клик графа $G_{seq} = (N, \neg D)$ последовательного выполнения. Для нашего примера множество клик описывается матрицей

$$U_{-D} = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{vmatrix} \begin{matrix} 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \end{matrix}$$

Критический путь = 5,

в которой для каждой строки указан размер клики, а максимальный размер дает длину максимального пути на графе.

Стоимость диаграммы, выраженная в числе функциональных узлов или процессорных элементов, необходимых для последовательно-параллельного исполнения вершин, оценивается посредством множества V_D клик графа $G_{conc}=(N, D)$ параллельного выполнения. Для нашего примера множество клик описывается матрицей

$$V_D = \left| \begin{array}{cccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{array} \right|$$

Критическое сечение = 2,

где для каждой строки указан размер клики, а максимальный размер дает ширину максимального сечения на графе.

СУДДР может быть непосредственно отображена в функциональную схему, построенную из мультиплексов. Для нашего примера такая схема изображена на рис.9.3.

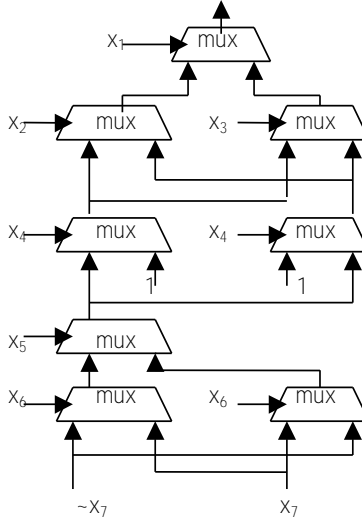


Рисунок 9.3 –Функциональная схема из мультиплексов, построенная по СУДДР

СУДДР является также источником для построения дизъюнктивной нормальной формы ДНФ булевой функции, описываемой диаграммой. ДНФ включает 92 литерала и 91 операцию конъюнкции и дизъюнкции:

$$\begin{aligned}
 f = & X_1 X_2 X_5 X_6 \neg X_7 \vee X_1 X_2 X_5 \neg X_6 X_7 \vee X_1 X_2 \neg X_5 \neg X_6 \neg X_7 \vee X_1 X_2 \neg X_5 X_6 X_7 \vee \\
 & X_1 X_2 \neg X_4 \vee X_1 \neg X_2 X_5 X_6 \neg X_7 \vee X_1 \neg X_2 X_5 \neg X_6 X_7 \vee X_1 \neg X_2 \neg X_5 \neg X_6 \neg X_7 \vee \\
 & X_1 \neg X_2 \neg X_5 X_6 X_7 \vee X_1 \neg X_2 X_4 \vee \neg X_1 X_3 X_5 X_6 \neg X_7 \vee \neg X_1 X_3 X_5 \neg X_6 X_7 \vee \\
 & \neg X_1 X_3 \neg X_5 \neg X_6 \neg X_7 \vee \neg X_1 X_3 \neg X_5 X_6 X_7 \vee \neg X_1 X_3 \neg X_4 \vee \neg X_1 \neg X_3 X_5 X_6 \neg X_7 \vee \\
 & \neg X_1 \neg X_3 X_5 \neg X_6 X_7 \vee \neg X_1 \neg X_3 \neg X_5 \neg X_6 \neg X_7 \vee \neg X_1 \neg X_3 \neg X_5 X_6 X_7 \vee \\
 & \neg X_1 \neg X_3 X_4.
 \end{aligned}$$

Ключевым шагом в переходе от СУДДР к ДИФДР является выбор функции разложения $c(x)$. С одной стороны, функция должна рассекать критические пути в СУДДР и увеличивать уровень параллелизма в вычислениях. С другой стороны, функция должна покрывать фрагменты диаграммы, имеющие много входных дуг и уменьшать число вершин и дуг в результирующей ДИФДР. Если мы ставим целью разрезание длинных путей в рассматриваемой СУДДР, то подходящей является функция $c(x)$, СУДДР которой изображена на рис.9.4. Выполняя операции минимизации $\min(f \wedge c)$ и $\min(\neg f \wedge \neg c)$, получаем смешанную СУДДР-ДИФДР, изображенную на рис.9.5.

После нескольких шагов выбора функции $c(x)$ и применения разложения МЕ2F.6.67 получаем ДИФДР, представленную на рис.9.6. Построенная диаграмма отображается в распаралелленную функциональную схему из мультиплексоров (рис.9.7).

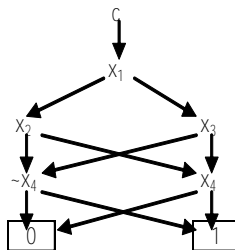


Рисунок 9.4 – СУДДР функции $c(x)$ (через \sim обозначена булева операция инвертирования)

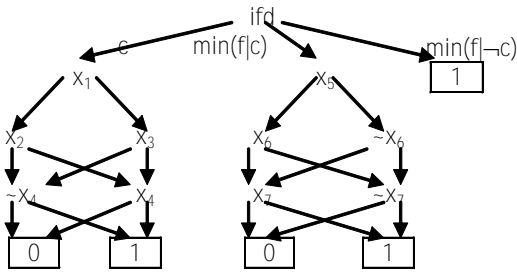


Рисунок 9.5 – Смешанная СУДДР-ДИФДР функции f (через \sim обозначена булева операция инвертирования)

Схема состоит из 5 мультиплексоров, на 3 мультиплексора меньше, чем схема, построенная по СУДДР. Глубина схемы измеряется тремя уровнями, в то время как глубина ранее построенной схемы измеряется пятью уровнями. Таким образом, выигрыш получен как по быстродействию, так и по сложности.

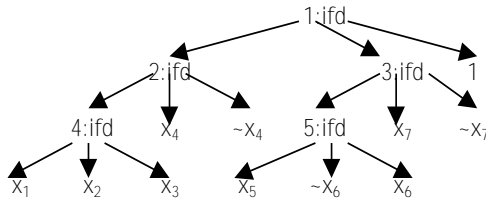


Рисунок 9.6 – Результирующая распараллеленная ДИФДР

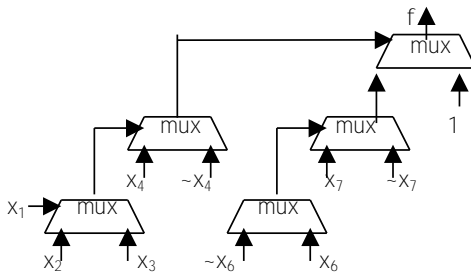


Рисунок 9.7 – Распараллеленная функциональная схема из мультиплексоров

Таблица 9.1

Сравнение ДНФ для СУДДР и ДИФДР

| Диаграмма решений | Число литералов в ДНФ |
|-------------------|-----------------------|
| СУДДР | 92 (100%) |
| ДИФДР | 18 (20%) |

Распараллеленная ДИФДР позволяет построить следующую систему ДНФ булевых функций, в которой символ \wedge конъюнкции опущен:

$$\begin{aligned}
 y_1 &= x_1 x_2 \vee \neg x_1 x_3, \\
 y_2 &= y_1 x_4 \vee \neg y_1 \neg x_4, \\
 y_3 &= x_5 \neg x_6 \vee \neg x_5 x_6, \\
 y_4 &= y_3 x_7 \vee \neg y_3 \neg x_7, \\
 f &= y_4 \vee \neg y_2.
 \end{aligned}$$

Табл.9.1 показывает, что система ДНФ, полученная из ДИФДР, содержит значительно меньшее число литералов, чем ДНФ, построенная по исходной СУДДР.

Выполним анализ уровня параллелизма в ДИФДР. Последовательно-параллельное исполнение вершин диаграммы описывается матрицей

$$C_{\text{ИФД}} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 1 & 0 \\ & & 0 & 0 & 1 \\ & & & 0 & 0 \\ & & & & 0 \end{vmatrix}$$

Уровень параллелизма характеризуется множеством пар распараллеленных вершин $D = \{(2,3), (2,5), (3,4), (4,5)\}$ и коэффициентом распараллеленности $k_c = 0.4$. Граф $G_{\text{conc}} = (N, D)$ изображен на рис.9.8 слева, граф $G_{\text{seq}} = (N, \neg D)$ изображен на том же рисунке справа. Множество клик графа G_{seq} описывается матрицей

$$U_{\neg D} = \begin{vmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{vmatrix} \begin{matrix} 3 \\ 3 \end{matrix}$$

Критический путь = 3,

которая определяет время исполнения диаграммы. Множество клик графа G_{conc} описывается матрицей

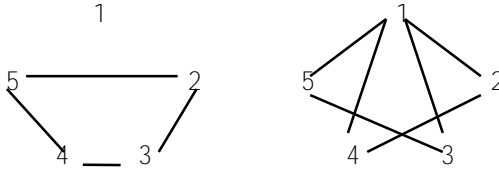


Рисунок 9.8 – Графы G_{conc} и G_{seq}

$$V_D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix}$$

Критическое сечение = 2.

которая определяет стоимость реализации ДлфДР:

Результаты (рис.9.9), полученные на ряде СУДДР и ДлфДР показывают, что использование распараллеленной ДлфДР сокращает число вершин на 16%, число дуг – на 37%, длину критического пути на 33%. Увеличение уровня распараллеленности составило 78%.

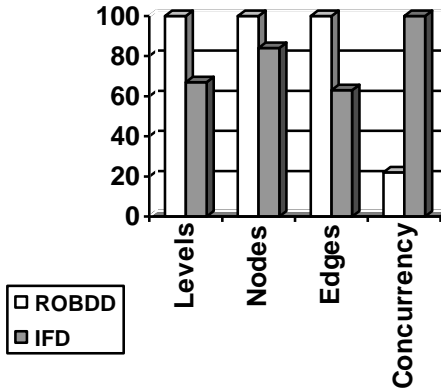


Рисунок 9.9 – Сравнение СУДДР и ДлфДР

9.3. Моделирование и синтез сумматоров

Логические проекты, описываемые диаграммами решений, характеризуются параметрами, среди которых важнейшими являются временные и сложностные. *Временные параметры* могут быть оценены через глубину диаграммы, определяемую максимальной длиной пути на графе от корня до листьев и измеряемую в числе *ifd* вершин. Если в функции $f(x)$ все переменные являются существенными, глубина СУДДР всегда равна n . Глубина ДlfДР может быть как больше, так и меньше n . При «правильном» построении глубина не может быть больше $n-1$. Это на единицу меньше, чем глубина СУДДР, поскольку листьями ДlfДР могут быть переменные, в то время как листьями СУДДР могут быть только константы. Глубина диаграммы характеризует время вычисления функции в ее аппаратной или программной реализации при условии, что операции, лежащие на разных путях диаграммы, выполняются параллельно. Техническая сложность логического проекта может быть оценена через число нетерминальных вершин в графе диаграммы. Поскольку ДlfДР расширяет число альтернативных представлений, размер минимальной ДlfДР не больше размера минимальной СУДДР для одной и той же функции.

Остановимся на классе цифровых устройств, называемых *сумматорами*, и выполним моделирование, синтез и преобразование последовательных и параллельных n -разрядных сумматоров в пространстве время-сложность. Входы и выходы одноразрядного полного сумматора показаны на рис.9.10. Входами являются слагаемые a и b , а также перенос из предыдущего разряда c_0 . Выходами являются сумма s и перенос в следующий разряд c_1 . На рис.9.11 показана система из двух СУДДР, описывающая поведение одноразрядного полного сумматора.

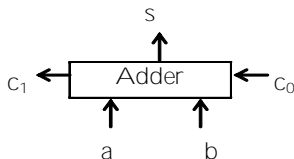


Рисунок 9.10 – Одноразрядный полный сумматор

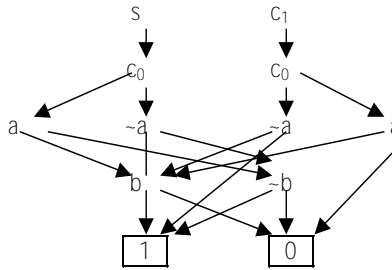


Рисунок 9.11 – Двухкорневая СУДДР для одноразрядного сумматора (через \sim обозначена булева операция инвертирования)

Поведение одноразрядного сумматора эквивалентным образом описывается системой из двух ДИФДР, показанной на рис.9.12. Размер ее намного меньше размера системы СУДДР: три нетерминальные вершины в ДИФДР вместо восьми нетерминальных вершин в СУДДР. Система является базисом для моделирования n -разрядных сумматоров. ДИФДР отображается в функциональную схему, состоящую из двух EXOR-вентилей и одного мультиплексора. Далее мы используем также другие системы ДИФДР, описывающие поведение сумматоров и порождающие различные структуры из вентилей.

Наиболее медленно работающим, но, в то же время, имеющим минимальную сложность, является n -разрядный сумматор с *последовательным переносом*, структура которого для $n=7$ показана на рис.9.13. В нем сумма вычисляется вентилями EXOR. Главной его частью является схема переноса, которая наиболее сильно влияет на временные и сложностные параметры.

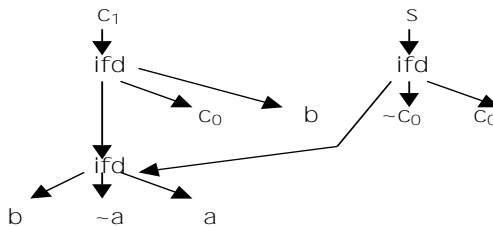


Рисунок 9.12 – Двухкорневая ДИФДР для одноразрядного полного сумматора

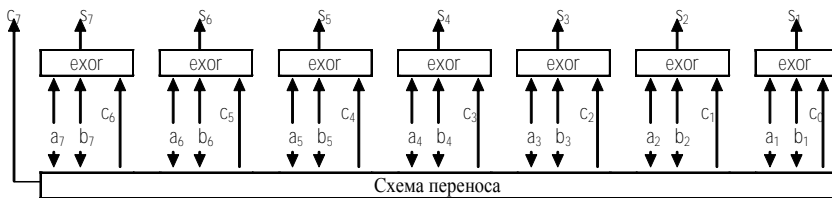


Рисунок 9.13 – Структура 7-разрядного сумматора с последовательным переносом

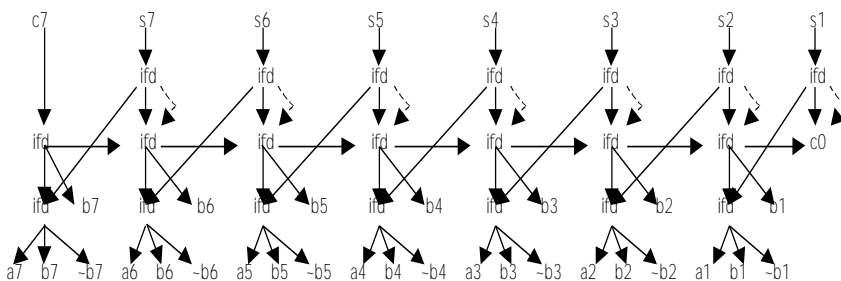


Рисунок 9.14 – Система ДИФДР 7-разрядного последовательного сумматора

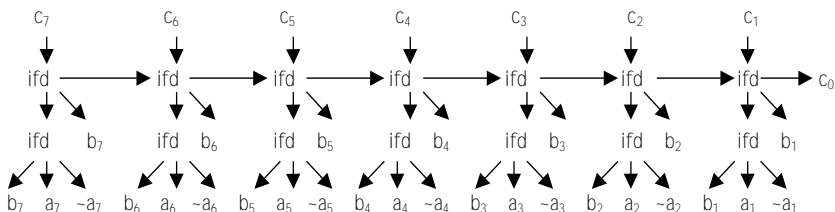


Рисунок 9.15 – Система ДИФДР схемы переноса

Система ДИФДР, описывающая поведение всего 7-разрядного сумматора, изображена на рис.9.14. На рис.9.15 показана система ДИФДР схемы переноса, состоящая из цепочки одинаковых по структуре фрагментов, соответствующих отдельным разрядам.

Распараллеливание *последовательного* n -разрядного сумматора выполняется разрезанием критических путей в системе ДИФДР. После разрезания, некоторые пары вершин, выполнявшиеся последовательно, становятся парами вершин, выполняющимися парал-

тельно. Такой метод обеспечивает генерацию параллельных сумматоров с различным уровнем параллелизма.

Синтез параллельного 3-разрядного сумматора. Длина критического пути в системе ДлфДР схемы переноса 3-разрядного последовательного сумматора равна 4 (рис.9.15). Очевидно, что не представляется возможным построение другой системы ДлфДР, имеющей глубину 2. Поэтому нашей целью является трансформация схемы переноса к системе ДлфДР глубиной 3.

ДлфДР для функции c_3 содержит путь максимальной длины. Воспользуемся операцией минимизации для разрезания этого пути. Выберем характеристическую функцию α_3 , изображенную на рис.9.16, для использования в разложении МЕ2F.6.67. Выполняя операции минимизации $min(c_3|\alpha_3)$ и $min(c_3|\neg\alpha_3)$, получаем ДлфДР, изображенные на рис.9.17 и рис.9.18 соответственно. Глубина трех построенных ДлфДР равна двум. Путем введения новой *ifd* вершины и присоединения к ней трех ДлфДР c_3 , $min(c_3|\alpha_3)$, $min(c_3|\neg\alpha_3)$ получаем диаграмму, изображенную на рис.9.19. Она представляет функцию переноса c_3 . Система ДлфДР, изображенная на рис.9.20, описывает одновременно три функции переноса c_1 , c_2 и c_3 .

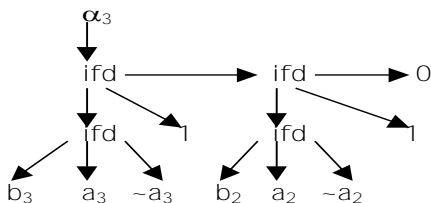


Рисунок 9.16 – ДлфДР α_3 (через \sim обозначена булева операция инвертирования)

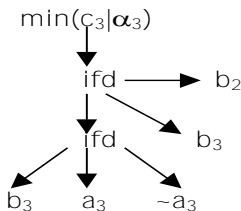


Рисунок 9.17 – ДлфДР $min(c_3|\alpha_3)$

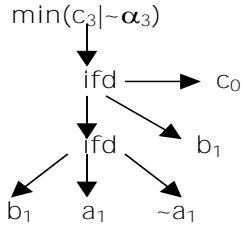


Рисунок 9.18 – ДІfDP $\min(c_3|\sim\alpha_3)$

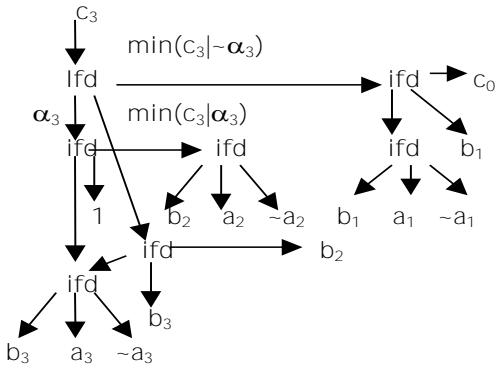


Рисунок 9.19 – ДІfDP функции переноса c_3

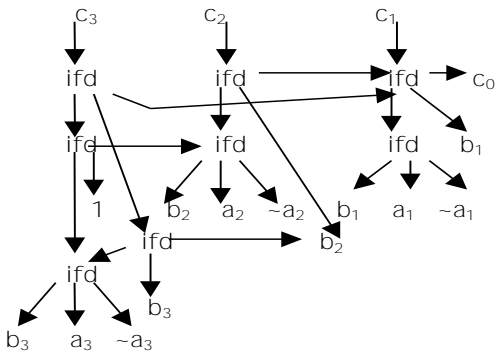


Рисунок 9.20 – Система ДІfDP для функций переноса c_1, c_2 и c_3

Синтез параллельного 7-разрядного сумматора. Критические пути в ДИФДР для функций C_7 , C_6 и C_5 имеют длину 8, 7 и 6 соответственно. В распараллеленной системе ДИФДР они должны быть разрезаны. С целью конструирования более быстрой ДИФДР для функции C_7 мы выбираем характеристическую функцию α_7 , изображенную на рис.9.21. Выполняя операции минимизации и собирая воедино полученные части, строим распараллеленную ДИФДР функции C_7 (рис.9.22). Для функций C_6 и C_5 распараллеленные ДИФДР показаны на рис.9.23 и 9.24 соответственно. В результате слияния трех ДИФДР для функций C_7 , C_6 и C_5 получаем систему ДИФДР для схемы переноса, изображенную на рис.9.25. Она имеет глубину 5 и рассматривается как промежуточный результат в процессе синтеза 7-разрядного параллельного сумматора с максимальным быстродействием. Разрезание критических путей в этой системе ДИФДР дает новую систему глубиной 4 (рис.9.26).

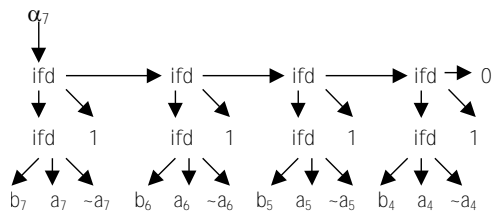


Рисунок 9.21 – ДИФДР характеристической функции α_7

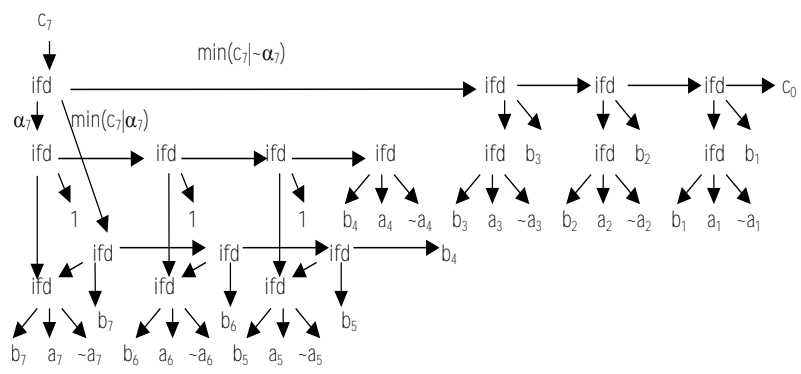


Рисунок 9.22 – Распараллеленная ДИФДР функции C_7

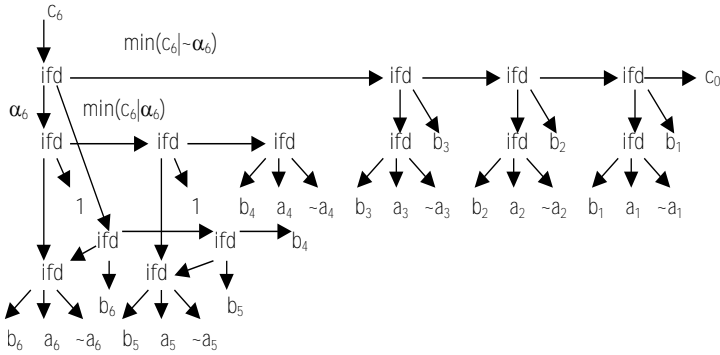


Рисунок 9.23 – Распараллеленная ДИДФР функции C_6

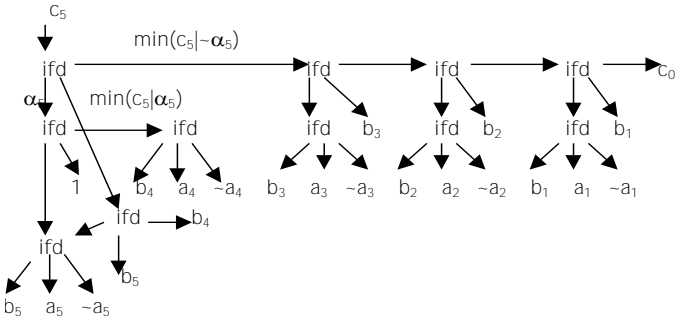


Рисунок 9.24 – Распараллеленная ДИДФР функции C_5

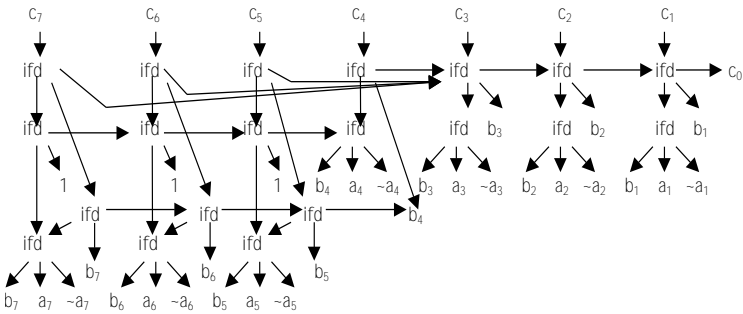


Рисунок 9.25 – Система ДИДФР глубины 5 для схемы переноса

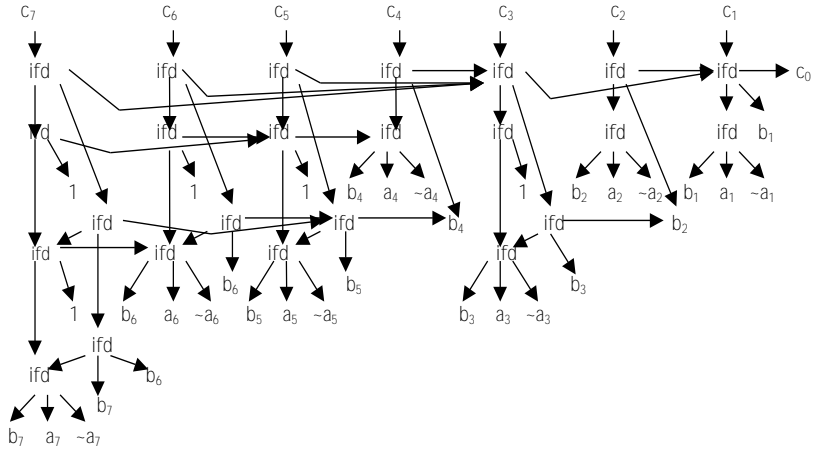


Рисунок 9.26 – Система ДfDP глубины 4 для схемы переноса (через ~ обозначена булева операция инвертирования)

Выполненные преобразования показывают, что предложенный механизм распараллеливания обеспечивает балансирование между быстродействием и сложностью сумматора. Построенные сумматоры имеют различную структуру, уровни распараллеленности и параметры (рис.9.27), при этом увеличение быстродействия сумматора влечет увеличение его сложности.

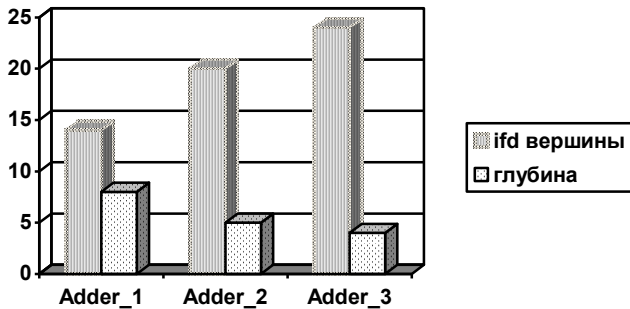


Рисунок 9.27 – Изменение параметров в зависимости от структуры 7-разрядного сумматора

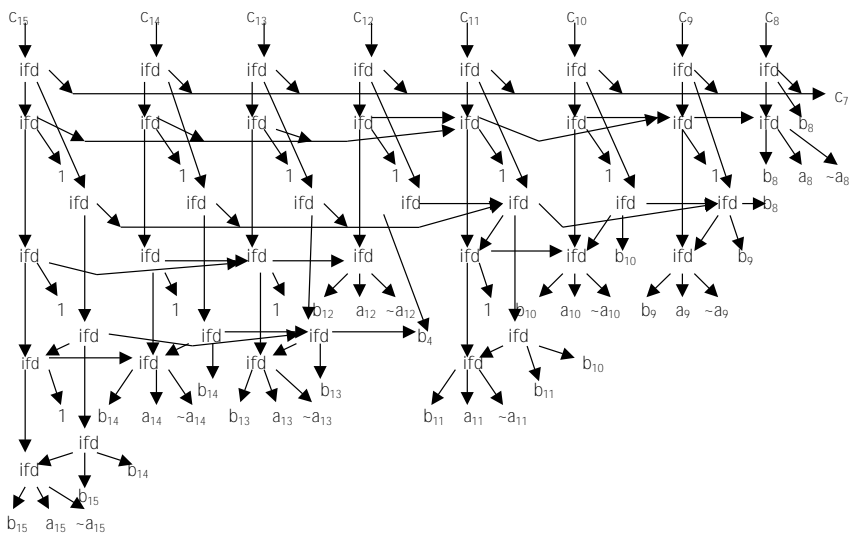


Рисунок 9.28 – Система ДифДР для разрядов 8-15 наиболее быстрого 15-разрядного параллельного сумматора

Конструирование наиболее быстрых n -разрядных сумматоров выполняется посредством сечений между битами 7-8, 15-16, 31-32, 63-64, 127-128, 255-256 и так далее. Для генерации сумматоров с промежуточным уровнем параллелизма используются и другие сечения. Система ДифДР для разрядов 8-15 наиболее быстрого 15-разрядного параллельного сумматора показана на рис.9.28.

Параметры n -разрядных сумматоров. Ключевыми параметрами сумматоров, представленных в виде ДифДР, являются сложность и глубина, выраженные в *ifd* вершинах, мультиплексорах, OR- и EQV-вентилях. Минимальная глубина параллельного n -разрядного сумматора определяется выражением

$$d_{min} = \lceil \log_2 n \rceil + 1. \quad (9.2)$$

Максимальная глубина последовательного n -разрядного сумматора:

$$d_{max} = n + 1. \quad (9.3)$$

Максимальная сложность S_{max} параллельного сумматора в числе ifd вершин оценивается следующими рекуррентными выражениями:

$$S_{max}^{ifd}(n) = S_{max}^{ifd}(n-1) + \lambda(n), \quad (9.4)$$

$$\lambda(n) = 2 * \lambda(n-1) + 2^{n-1}, \quad (9.5)$$

$$\lambda(1)=2 \text{ и } S_{max}^{ifd}(1)=2. \quad (9.6)$$

Максимальное число вентилях в схеме параллельного сумматора оценивается следующими рекуррентными соотношениями:

$$S_{max}^{MUX}(n) = S_{max}^{ifd}(n) / 2, \quad (9.7)$$

$$S_{max}^{EQV}(n) = n, \quad (9.8)$$

$$S_{max}^{OR}(n) = S_{max}^{MUX}(n) - S_{max}^{EQV}(n). \quad (9.9)$$

Минимальная сложность последовательного n -разрядного сумматора:

$$S_{min}^{ifd}(n) = 2 \cdot n. \quad (9.10)$$

Соотношения (9.2)-(9-10) позволяют рассчитать сложность и глубину последовательных и параллельных сумматоров для различных разрядностей. В табл.9.2 даны параметры сумматоров. С увеличением разрядности слагаемых до 511 глубина параллельного сумматора сократилась в 51.2 раза, в то же время сложность

Таблица 9.2

Параметры сумматоров

| № | Биты | Последовательные сумматоры | | Параллельные сумматоры | |
|---|------|----------------------------|-----------|------------------------|-----------|
| | | Глубина | Сложность | Глубина | Сложность |
| 1 | 1 | 2 | 2 | 2 | 2 |
| 2 | 3 | 4 | 6 | 3 | 8 |
| 3 | 7 | 8 | 14 | 4 | 24 |
| 4 | 15 | 16 | 30 | 5 | 64 |
| 5 | 31 | 32 | 62 | 6 | 160 |
| 6 | 63 | 64 | 126 | 7 | 384 |
| 7 | 127 | 128 | 254 | 8 | 896 |
| 8 | 255 | 156 | 510 | 9 | 2048 |
| 9 | 511 | 512 | 1022 | 10 | 4608 |

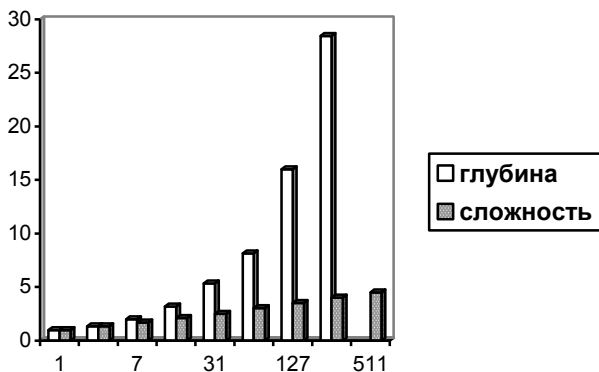


Рисунок 9.29 – Сокращение глубины и увеличение сложности параллельного сумматора по сравнению с последовательным сумматором в зависимости от разрядности операндов

увеличилась только в 4.5 раза по сравнению с последовательным сумматором. Динамику изменения параметров описывает диаграмма, изображенная на рис.9.29. Следует заметить, что для каждой разрядности в промежутке между самым медленным, но дешевым, сумматором и самым быстрым, но дорогим, сумматором предложенная методика позволяет синтезировать огромное количество сумматоров с промежуточными параметрами. Это может быть выгодно использовано в оптимизирующих системах синтеза.

9.4. Моделирование и синтез умножителей

Матричный комбинационный *умножитель* Брауна 4×4 представляет собой регулярную структуру, изображенную на рис.9.30. Операнды операции умножения представляются четырехразрядными векторами $a=(a_3, a_2, a_1, a_0)$ и $b=(b_3, b_2, b_1, b_0)$, описывающими целые числа в двоичной системе счисления. Результат умножения представляется вектором $\rho=(\rho_7, \rho_6, \rho_5, \rho_4, \rho_3, \rho_2, \rho_1, \rho_0)$. Умножитель построен из элементов двух типов: одноразрядного полного сумматора FA и AND-вентилей. Функциональная схема умножителя может быть расширена на произвольное число бит.

Умножитель эффективно моделируется системой двоичных *ifd* диаграмм решений Д1fДР, имеющей размер 48 и глубину 11 *ifd* вершин (рис.9.31). В системе один элемент FA представляется двумя *ifd* вершинами, как показано на рис.9.12. В отличие от *n*-разрядного последовательного сумматора, система Д1fДР умножителя Брауна не имеет регулярной структуры. Тем не менее, метод

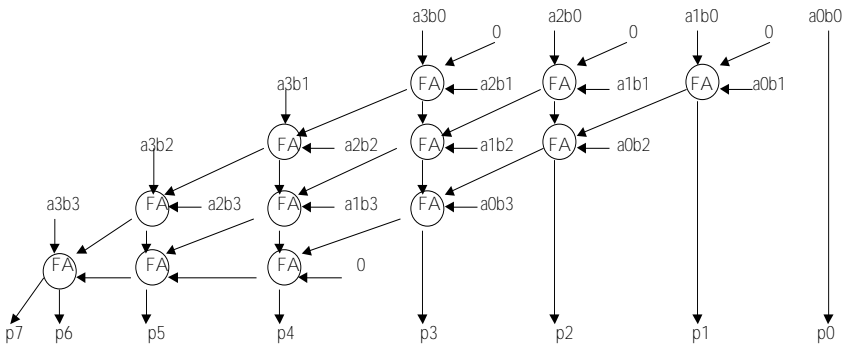


Рисунок 9.30 – Умножитель Брауна 4×4 (FA – одноразрядный полный сумматор)

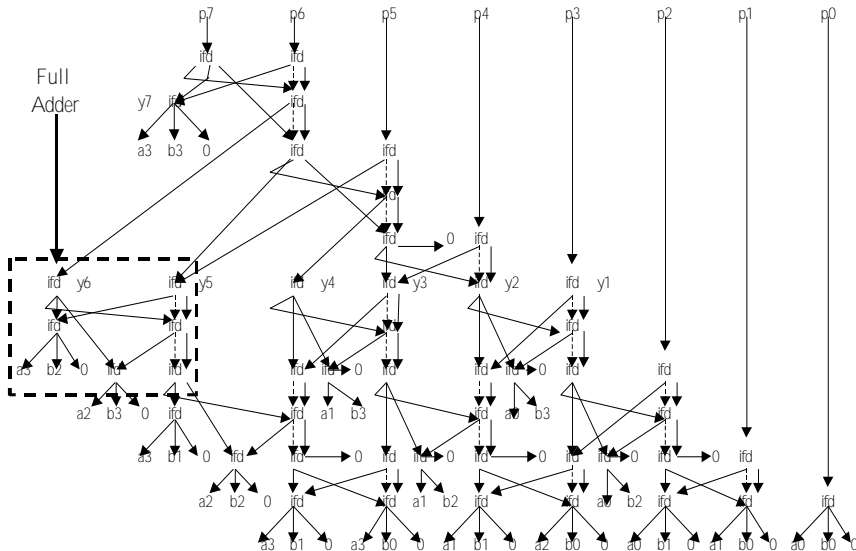


Рисунок 9.31 – Система Д1fДР, моделирующая умножитель Брауна 4×4

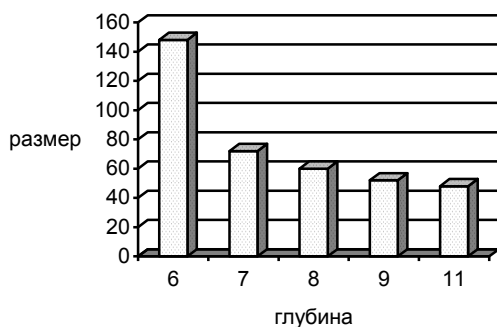


Рисунок 9.32 – Зависимость размера от глубины распараллеливаемой системы ДИФДР

разрезания длинных путей применим и в этом случае для получения более быстрых умножителей. Для распараллеливания и синтеза умножителей с меньшей максимальной глубиной системы ДИФДР использовано программное обеспечение, выполняющее операции над диаграммами, включая операцию минимизации над ДИФДР. Это позволило сгенерировать четыре более быстрых умножителя, имеющих глубину от 6 до 10. Платой за ускорение явилось усложнение умножителя более чем в три раза, что следует из диаграммы, изображенной на рис.9.32.

При уменьшении глубины с 11 to 7 *ifd* вершин (на 36 %) размер системы увеличивался с 48 to 72 *ifd* вершин (на 33 %). Уменьшение глубины с 7 до 6 вызвало скачок в увеличении размера системы ИФД с 72 до 148. Полученная зависимость подтверждает тот факт, что при «правильном» проектировании платой за увеличение быстродействия является возрастание сложности. Эта закономерность наблюдалась также при минимизации систем *if* диаграмм, описывающих случайную логику.

9.5. Синтез частично определенных проектов

Очень часто цифровые системы являются не полностью определенными. Поведение таких систем может быть промоделировано логическими уравнениями, построенными с использованием час-

точных операций. Возможны два подхода к синтезу полностью и не полностью определенных систем:

- полностью определенная система разбивается на не полностью определенные части, которые синтезируются и оптимизируются с интенсивным использованием понятия *don't care*;
- не полностью определенная система моделируется посредством выражений и уравнений частичной и смешанной логики, после чего применяется соответствующий аппарат синтеза и оптимизации.

Для логического проекта *dc*-множество может быть внешним и внутренним. Метод синтеза частично определенных логических проектов на основе диаграмм решений включает следующие шаги:

- представление поведения в виде системы ДИФДР (возможно в виде системы СУДДР);
- представление *dc*-множества посредством частично определенных диаграмм;
- оптимизация частично определенных диаграмм и их доопределение до полностью специфицированных диаграмм;
- отображение полностью специфицированных диаграмм в технологический базис.

Различные описания цифровой системы могут быть легко отображены в систему ДИФДР. Если дана схема из вентилях, то каждый вентиль может быть представлен одной или несколькими *ifd* вершинами. Если даны булевы уравнения, каждая булева операция легко представляется *ifd* вершиной, а структура системы ДИФДР отображает структуру уравнений.

Характеристическая функция области определенности проекта есть инверсия характеристической функции *dc*-множества. Разнообразные внешние и внутренние *dc*-множества, описанные арифметическими или логическими выражениями, а также различного рода отношениями, могут представляться системами ДИФДР.

Пусть, например, *dc*-множество характеризуется предикатом “*a* не равно *b*” или $a \neq b$. Для операндов *a* и *b* разрядностью четыре оно описывается следующей системой характеристических функций определенности $d_0 \dots d_3$:

$$\begin{aligned}
d_0 &= (a_0 \leftrightarrow b_0), \\
d_1 &= d_0 \wedge (a_1 \leftrightarrow b_1), \\
d_2 &= d_1 \wedge (a_2 \leftrightarrow b_2), \\
d_3 &= d_2 \wedge (a_3 \leftrightarrow b_3)
\end{aligned}$$

На рис.9.33 эти функции представляются системой из семи ДИФДР. Для семи бит характеристические функции d_1-d_7 области определенности показаны на рис.9.34.

Пусть теперь dc -множество описывается предикатом “ a меньше b ” или $a < b$. Для 4-разрядных операндов a и b этому множеству соответствует следующая характеристическая функция d области определенности:

$$\begin{aligned}
d = (a \geq b) &= ((a_3 \wedge \neg b_3) \vee (a_3 \leftrightarrow b_3)) \wedge \\
&\quad ((a_2 \wedge \neg b_2) \vee (a_2 \leftrightarrow b_2)) \wedge \\
&\quad ((a_1 \wedge \neg b_1) \vee (a_1 \leftrightarrow b_1)) \wedge \\
&\quad ((a_0 \wedge \neg b_0) \vee (a_0 \leftrightarrow b_0)).
\end{aligned}$$

Пусть характеристические функции d_0, \dots, d_7 для $a=b$ описывают область определенности сумматора, представленного системой ДИФДР на рис.9.14. Тогда требуемое поведение сумматора устанавливается только на части значений слагаемых. Поведение на другой части можно выбирать таким образом, чтобы максимально

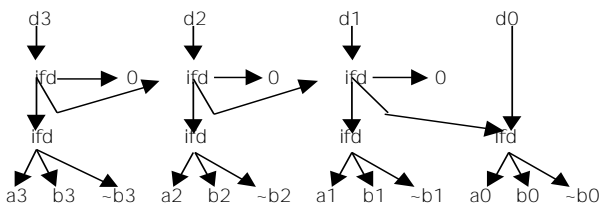


Рисунок 9.33 – Функции определенности d_0-d_3 при $(a=b)$ для 4 бит

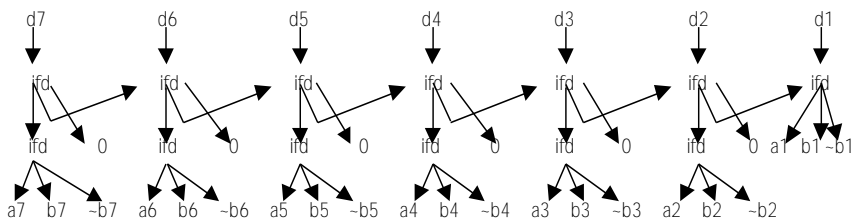


Рисунок 9.34 – Функции определенности d_1-d_7 при $(a=b)$ для 7 бит

упростить систему ДлфДР, а в итоге и стоимость реализации сумматора. С целью упрощения выполняем операцию минимизации $\min(s_i|d_i)$, $i=1, \dots, 7$ для каждого разряда суммы. В результате получаем систему ДлфДР, изображенную на рис.9.35. Применение правил редуцирования к этой системе дает систему, изображенную на рис.9.36 и описывающую фактически сдвиг влево.

Обратимся теперь к умножителю Брауна, представленному системой ДлфДР на рис.9.31. Ограничим область определенности умножителя системой ДлфДР, изображенной на рис.9.33. Применим операцию минимизации к функциям p_0, \dots, p_7 по функциям d_0, \dots, d_3 для $a=b$. В результате получаем минимизированный умножитель, изображенный на рис.9.37. Как видим, произошло существенное снижение сложности умножителя, с 48 до 15 *ifd* вершин. Глубина уменьшилась с 11 до 3 *ifd* вершин.

Минимизация сумматора и умножителя по области определенности $a \geq b$ осуществляется аналогичным образом путем выполнения операций $\min(Adder|a \geq b)$ и $\min(Multiplier|a \geq b)$. Но прежде строится система ДлфДР таким образом, чтобы характеристические функции определенности каждого разряда представлялась двумя *ifd* вершинами.

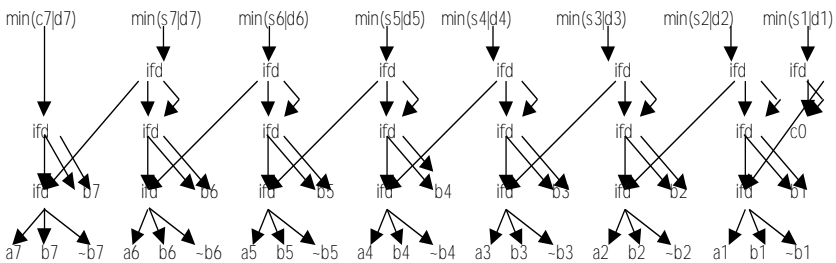


Рисунок 9.35 – Результат выполнения операции $\min(Adder|a=b)$

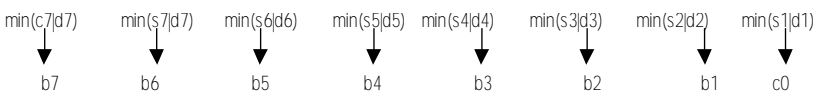


Рисунок 9.36 – Результат редуцирования $\min(Adder|a=b)$

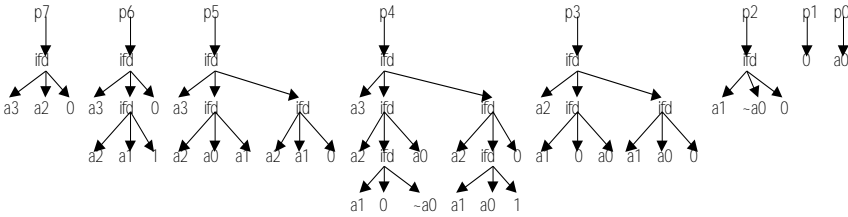


Рисунок 9.37 – Минимизированный умножитель $\min(\text{Multiplier}|a=b)$

9.6. Минимизация мощности, потребляемой цифровой системой

Снижение *мощности*, потребляемой цифровой системой, является актуальной научно-технической проблемой, которая привлекает в последнее время все большее внимание ученых и инженеров. Решение этой проблемы направлено на сокращение энергопотребления, разработку, изготовление и использование портативных электронных приборов различного назначения, увеличение времени жизни переносных источников питания; увеличение плотности размещения компонентов на кристалле и существенное расширение функциональных возможностей одного чипа. Абсолютное большинство проектируемых и изготавливаемых сегодня цифровых схем базируется на использовании КМОП технологии.

Потребляемая в интегральных КМОП схемах мощность P складывается из трех частей:

$$P = P_{load} + P_{shortcircuit} + P_{leakage} \quad (9.11)$$

где P_{load} – мощность, расходуемая на зарядку и разрядку нагрузочных емкостей, $P_{shortcircuit}$ – мощность короткого замыкания, $P_{leakage}$ – мощность, связанная с токами утечки. Мощность P_{load} в выражении (9.11) является доминирующей в современных интегральных КМОП схемах. Она оценивается выражением

$$P_{load} = \frac{1}{2} C_L \cdot V_{DD}^2 \cdot E_w \cdot f_{clk}, \quad (9.12)$$

где C_L – перезаряжаемая емкость; V_{DD}^2 – напряжение питания; E_w – активность переключения, равная среднему числу изменений уровней сигналов за один цикл синхронизации; f_{clk} – тактовая частота. Из выражения (9.12) следует, что мощность P_{load} определяется как статическими параметрами C_L , V_{DD}^2 , f_{clk} , так и переключательной активностью E_w , являющейся динамическим параметром, зависящим от условий применения схемы. Минимизация мощности, потребляемой цифровой схемой, достигается в значительной мере минимизацией ее активности.

Нами предлагаются *вероятностный* метод оценки *активности* цифровых электронных схем, представленных новым классом DlfDR диаграмм решений, и метод минимизации активности схем путем эквивалентных преобразований диаграмм этого класса. Теоретические результаты иллюстрируются конкретными примерами оценки и минимизации активности комбинационных сумматоров.

Базовые логические элементы (вентили), на основе которых строятся цифровые схемы, легко представляются *if* диаграммами решений. Обратное, *if* диаграммы различных конфигураций легко отображаются на структуры из вентилях. Так, одноразрядный полный сумматор, реализованный схемой, построенной из полусумматоров (рис.9.38), моделируется *if* диаграммами, представленными на рис.9.39. Диаграммы включают 5 нетерминальных *ifd* вершин, расположенных на 3 ярусах, 11 терминальных вершин, из которых 3 являются константами, и также включают 15 дуг. Две *ifd* вершины определяют трехместную функцию EXOR, две – операцию AND, одна - операцию OR.

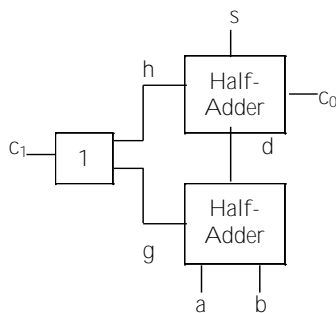


Рисунок 9.38 – Сумматор, построенный из полусумматоров

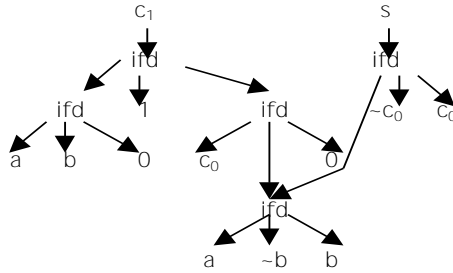


Рисунок 9.39 – Моделирование сумматора *if* диаграммами (через ~ обозначена булева операция инвертирования)

Оценка активности if диаграмм. Активность цифровой схемы оценим через активность *if* диаграмм, представляющих эту схему. Активность *if* диаграмм зависит в свою очередь от частоты переключения входных сигналов. Последовательность значений входных, внутренних и выходных сигналов будем рассматривать как случайный процесс, а значение сигнала s рассматривать как случайную переменную. Обозначим через $p(s) = p(s=1)$ вероятность того, что $s=1$. Вероятность того, что $s=0$, равна $p(s=0) = 1-p(s=1)$. Тогда вероятность того, что сигнал s изменяет значение (говорят, что сигнал активен в этом случае), равна

$$p_w(s) = 2 \cdot p(s=0) \cdot p(s=1). \quad (9.13)$$

Активность константных сигналов $p_w(1) = p_w(0) = 0$. Пусть T – множество терминальных и N – множество нетерминальных вешин *if* диаграммы. Общая активность W диаграммы оценивается суммой

$$W = W_T + W_N, \quad (9.14)$$

в которой активность терминальных вершин оценивается выражением

$$W_T = \sum_{t_i \in T} p_w(t_i), \quad (9.15)$$

а активность нетерминальных вершин - выражением

$$W_N = \sum_{y_i \in N} p_w(y_i) \cdot e(y_i), \quad (9.16)$$

где $e(y_i)$ – число входных дуг для вершины y_i . Вероятность единичного значения сигнала в нетерминальной вершине y_i оценивается выражением

$$\rho(y_i = 1) = \sum_{\substack{a \in B^n \\ y_j(a)=1}} \rho(a), \quad (9.17)$$

где $y_j(x)$ – логическая функция, соответствующая вершине y_j ; $a = (a_1, \dots, a_n)$ – набор значений векторного аргумента x . Вероятность того, что аргумент x принимает значение a , есть

$$\rho_a = \prod_{i=1}^n \rho(x_i = a_i), \quad (9.18)$$

где $\rho(x_i = a_i)$ – вероятность того, что сигнал x_i принимает значение a_i .

Значение сигнала в вершине y_j для векторного значения a входного сигнала x рассчитывается в процессе обхода диаграммы снизу вверх слева направо. Активность диаграммы зависит от вероятностей значений входных переменных. Результаты расчета активности накапливаются в табл.9.3.

Описанный метод является точным, но, в то же время, достаточно трудоемким, поскольку перебирает все значения входного векторного сигнала x . Избежать перебора позволяют аналитические методы, которые наиболее эффективны при отсутствии корреляции между значениями в вершинах. С появлением корреляции эти методы усложняются.

Таблица 9.3

Расчет активности *if*-диаграммы

| N | Переменные | Вершины ДИФДР | Вероятность |
|-------------|---------------------------------------|--|-------------------------|
| | $x_1 \dots x_{n-1} x_n$ | $y_1 \dots y_k$ | $\rho(a_1, \dots, a_n)$ |
| 0 | 0...00 | $y_1(0\dots00) \dots y_k(0\dots00)$ | $\rho(0\dots00)$ |
| 1 | 0...01 | $y_1(0\dots01) \dots y_k(0\dots01)$ | $\rho(0\dots01)$ |
| ... | ... | ... | ... |
| $2^n - 2$ | 1...10 | $y_1(1\dots10) \dots y_k(1\dots10)$ | $\rho(1\dots10)$ |
| $2^n - 1$ | 1...11 | $y_1(1\dots11) \dots y_k(1\dots11)$ | $\rho(1\dots11)$ |
| Вероятность | $\rho(x_1=1) \dots \rho(x_n=1)$ | $\sum \rho(a) \dots \sum \rho(a)$ $y_1(a)=1 \dots y_k(a)=1$ | $\sum \rho(a)=1$ a |
| Активность | $\rho_{sw}(x_1) \dots \rho_{sw}(x_n)$ | $\rho_{sw}(y_1) \dots \rho_{sw}(y_k)$ | $\sum \rho_{sw}$ |

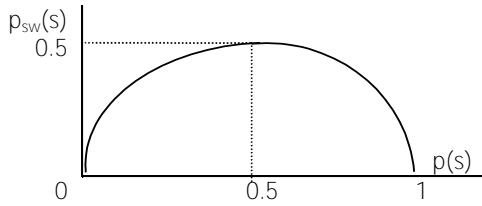


Рисунок 9.40 – Зависимость активности сигнала от вероятности его единичного значения

Преобразование if диаграммы с целью минимизации активности. Из соотношения (9.13) вытекает график зависимости (рис.9.40) активности $p_{sw}(s)$ сигнала s от вероятности $p(s)$ его единичного значения. Из графика следует, что сигнал имеет высокую активность, когда $p(s) \cong 0.5$, и имеет низкую активность при $p(s) \cong 0$ или $p(s) \cong 1$.

Суммарная активность всей диаграммы зависит от числа входных сигналов, количества внутренних вершин и дуг, активности каждой внутренней вершины. С целью минимизации первых двух составляющих целесообразно синтезировать *if* диаграммы минимальной размерности, для минимизации третьей составляющей целесообразно генерировать каждую вершину диаграммы таким образом, чтобы активность соответствующего сигнала была минимальной. Вероятность единичного значения сигнала, соответствующего внутренней вершине f диаграммы, определяется как

$$p(f) = p(\alpha) \cdot p(g|\alpha) + (1 - p(\alpha)) \cdot p(h|\neg\alpha), \quad (9.19)$$

где $p(\alpha)$ – вероятность единичного значения функции α , соответствующей вершине *ifd*(\hat{n}); $p(g|\alpha)$ – вероятность единичного значения функции g , соответствующей вершине *high*(\hat{n}), при условии, что значение $\alpha=1$, $p(h|\neg\alpha)$ – вероятность единичного значения функции h , соответствующей вершине *low*(\hat{n}) при условии, что значение $\alpha=0$. Условия низкой активности на выходе *ifd* вершины приведены в табл.9.4. Эти условия определяют требования к выбору диаграмм α , g и h в момент применения разложения ME2F.6.67, где *dc* обозначает безразличное значение. Например, вероятность $p(\hat{n})$ единичного значения функции f приблизительно равна 0 (а это соответствует низкой активности), если вероятности $p(g|\alpha)$ и $p(h|\neg\alpha)$ близки к 0.

Таблица 9.4

Условия низкой активности *ifd* вершины

| $p(\alpha)$ | $p(g \alpha)$ | $p(h \neg\alpha)$ | $p(f)$ |
|-------------|---------------|-------------------|-----------|
| dc | $\cong 0$ | $\cong 0$ | $\cong 0$ |
| $\cong 0$ | dc | $\cong 0$ | |
| $\cong 1$ | $\cong 0$ | dc | |
| dc | $\cong 1$ | $\cong 1$ | $\cong 1$ |
| $\cong 0$ | dc | $\cong 1$ | |
| $\cong 1$ | $\cong 1$ | dc | |

Выполним синтез и оптимизацию системы *if* диаграмм для одноразрядного полного сумматора. Отобразим систему СУДДР (рис.9.11) в систему Д|fДР, изображенную на рис.9.41. Выбрав функцию $\alpha = a \oplus b$ и выполнив минимизацию функций *s* и *c*₁, получаем систему Д|fДР, показанную на рис.9.42. По сравнению с исходной системой, сгенерированная система имеет иерархическую структуру и включает меньшее число вершин и дуг. Ей соответствует комбинационная схема, изображенная на рис.9.43. Расчет активности системы Д|fДР приведен в табл.9.5.

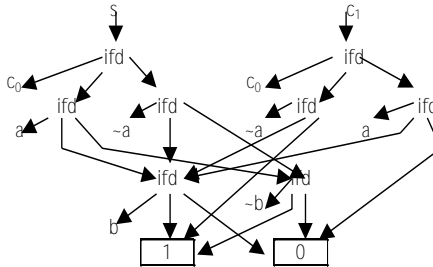


Рисунок 9.41 – Система Д|fДР, построенная по системе СУДДР

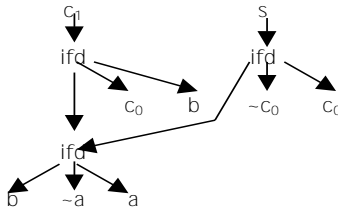


Рисунок 9.42 – Минимизированная система Д|fДР

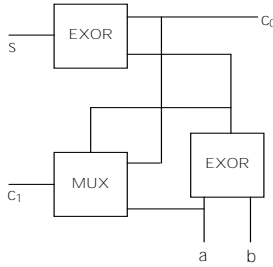


Рисунок 9.43 – Функциональная схема для минимизированного сумматора

Минимизация активности n-разрядных комбинационных сумматоров. Входы и выходы восьмиразрядного комбинационного сумматора показаны на рис.9.44. Активность сумматора зависит от вероятностей единичных значений входов $p(a_0), \dots, p(a_7), p(b_0), \dots, p(b_7)$ слагаемых и входа переноса $p(c_0)$. С целью оценки единичных значений вероятностей входов рассмотрим два случая. Первый случай предполагает, что на входы сумматора подается 80% положительных и 20% отрицательных чисел в дополнительном коде. Плотность распределения вероятностей для положительных чисел показана на рис.9.45, где $p_{max}=1/65, p_{min}=1/(64 \times 65)$. Плотность распределения вероятностей для отрицательных чисел определяется

Таблица 9.5

Расчет активности системы ДИФДР одnorазрядного сумматора

| N | Входные сигналы | | | Вершины ДИФДР | | | | | Вероятность |
|--------------------|-----------------|-----|----------------|---------------|------|------|-----|-----|-----------------|
| | a | b | c ₀ | 1 | 2 | 3 | 4 | 5 | $p(x)$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.189 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0.189 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0.189 |
| 3 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0.189 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0.061 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0.061 |
| 6 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0.061 |
| 7 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0.061 |
| Вероятность | 0.5 | 0.5 | 0.25 | 0.37 | 0.25 | 0.12 | 0.5 | 0.5 | 1 |
| Активность | 0.5 | 0.5 | 0.38 | 0.47 | 0.38 | 0.21 | 0.5 | 0.5 | $\Sigma = 3.44$ |

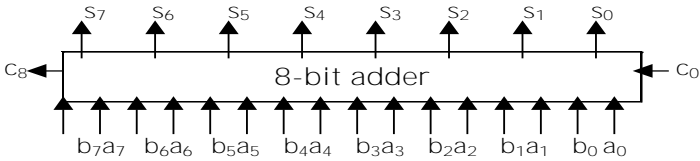


Рисунок 9.44 – Входы и выходы 8-разрядного сумматора

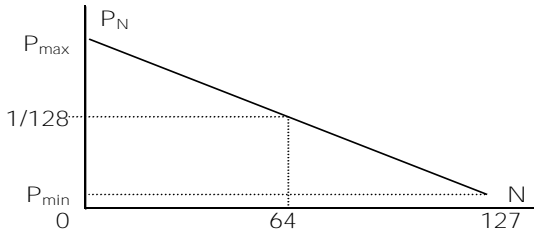


Рисунок 9.45 – Плотность распределения вероятностей положительных чисел

аналогичным образом. Во втором случае на вход сумматора подаются только положительные числа. Вероятности единичного значения каждого разряда, рассчитанные для первого и второго случаев, приведены в табл.9.6.

Результаты *экспериментов* по оценке активности восьмиразрядных сумматоров, базирующихся на трех различных реализациях одноразрядного сумматора и выполненных для двух распределений вероятностей единичных значений входов (табл.9.6), приведены в табл.9.7 и табл.9.8.

Таблица 9.6

Вероятности единичного значения для бит

| Бит | Вероятность | |
|-----|----------------------|-----------------------|
| | 80% положительных | 100% положительных |
| 0 | 0.500 | 0.500 |
| 1 | 0.498 | 0.498 |
| 2 | 0.493 | 0.494 |
| 3 | 0.484 | 0.487 |
| 4 | 0.466 | 0.471 |
| 5 | 0.430 | 0.440 |
| 6 | 0.357 | 0.379 |
| 7 | 0.198 | 0.256 |

Таблица 9.7

Активность сумматора для 80% положительных чисел

| Бит | Вероятность Переноса 1 | Восьми разрядный сумматор | | | | | |
|----------|------------------------------|---------------------------|-------|-----------|-------|-----------|-------|
| | | Вариант 1 | | Вариант 2 | | Вариант 3 | |
| | | W_T | W_N | W_T | W_N | W_T | W_N |
| 0 | 0.25 | 2.50 | 2.25 | 5.00 | 2.63 | 2.00 | 1.88 |
| 1 | 0.37 | 3.62 | 2.56 | 5.75 | 2.72 | 3.12 | 1.97 |
| 2 | 0.43 | 3.90 | 2.66 | 5.93 | 2.74 | 3.40 | 1.99 |
| 3 | 0.45 | 3.97 | 2.69 | 5.98 | 2.74 | 3.47 | 1.99 |
| 4 | 0.44 | 3.97 | 2.68 | 5.97 | 2.74 | 3.48 | 1.99 |
| 5 | 0.40 | 3.93 | 2.62 | 5.89 | 2.72 | 3.44 | 1.98 |
| 6 | 0.31 | 3.74 | 2.44 | 5.55 | 2.63 | 3.28 | 1.92 |
| 7 | 0.14 | 2.58 | 1.85 | 4.04 | 2.13 | 2.56 | 1.60 |
| Σ | | 48.26 | | 65.15 | | 40.06 | |

Таблица 9.8

Активность сумматора для 100% положительных чисел

| Бит | Вероятность Переноса 1 | Восьми разрядный сумматор | | | | | |
|----------|------------------------------|---------------------------|-------|-----------|-------|-----------|-------|
| | | Вариант 1 | | Вариант 2 | | Вариант 3 | |
| | | W_T | W_N | W_T | W_N | W_T | W_N |
| 0 | 0.25 | 2.50 | 2.25 | 5.00 | 2.63 | 2.00 | 1.88 |
| 1 | 0.37 | 3.62 | 2.56 | 5.75 | 2.72 | 3.12 | 1.97 |
| 2 | 0.43 | 3.90 | 2.66 | 5.93 | 2.74 | 3.40 | 1.99 |
| 3 | 0.45 | 3.97 | 2.69 | 5.98 | 2.75 | 3.47 | 2.00 |
| 4 | 0.45 | 3.98 | 2.69 | 5.97 | 2.74 | 3.48 | 1.99 |
| 5 | 0.41 | 3.95 | 2.64 | 5.92 | 2.73 | 3.45 | 1.99 |
| 6 | 0.34 | 3.81 | 2.50 | 5.68 | 2.66 | 3.34 | 1.94 |
| 7 | 0.19 | 3.25 | 2.10 | 4.70 | 2.37 | 2.87 | 1.75 |
| Σ | | 49.08 | | 66.27 | | 40.64 | |

В первой реализации поведение одноразрядного сумматора описывается системой ДИФДР из рис.9.41, во второй – из рис.9.39, в третьей – из рис.9.42.

Активность сумматоров складывается из активности входов W_T и активности внутренних узлов W_N . Активность сумматора, базирующегося на минимизированной ДИФДР (вариант 3), минимальна и составляет 40.06 единиц для 80% положительных чисел и 40.64 единиц для 100% положительных чисел. Активность первого варианта в среднем на 20.6% выше третьего, активность второго вариан-

та выше третьего на 62.8%. Вероятности переноса 1 из крайних разрядов ближе к 0, из средних разрядов – ближе к 0.5. Активность одноразрядных сумматоров, соответствующих средним разрядам, выше активности одноразрядных сумматоров, соответствующих крайним разрядам.

Анализ результатов, вытекающих из табл.9.7, 9.8, показывает, что активность сумматоров зависит от вероятностей значений входов. Сумматоры являются слабо активными при значениях вероятностей, близких к 0 или 1. Сумматоры наиболее активны в случае, когда вероятности значений близки к 0.5. При одних вероятностях значений входов третий сумматор явно менее активен, чем первый и второй, при других - активность первого сумматора приближается к активности третьего сумматоров. Из результатов следует, что для различных разрядов n -разрядных сумматоров в ряде случаев целесообразно использовать различные схемы одноразрядных сумматоров с целью минимизации активности всего многоразрядного сумматора.

9.7. *Верификация логической системы*

Смешанная алгебра, ее законы, разложения частично определенных логических функций, а также вытекающие из нее диаграммы решений являются базисом для решения задач *верификации* логических проектов. Так верификация комбинационной схемы выполняется обычно демонстрацией эквивалентности спецификации и реализации по значениям на входах и выходах. Эквивалентность следует из идентичности описаний поведений спецификации и реализации, представленных в канонической форме, каковой является СУДДР. Для частично определенных проектов требования эквивалентности ослабляются. Для верификации таких проектов смешанная алгебра предоставляет все необходимые инструменты.

Благодаря операции минимизации смешанная алгебра позволяет избежать использования канонической формы. Если в спецификации поведение представляется векторной частично определенной функцией $\Phi = ((v_1|d_1), \dots, (v_m|d_m))$, а реализация представляется векторной полностью определенной булевой функцией $S = (s_1, \dots, s_m)$,

то достаточно доказать $\min(s_i | v_i \wedge d_i) = 1$ и $\min(\neg s_i | \neg v_i \wedge d_i) = 1$ при $i=1, \dots, m$ для того, чтобы верифицировать проект.

9.8. Историческая справка

Моделирование цифровых схем и систем *if-then-else* диаграммами использовалось во многих работах и системах. Распараллеливание вычислений в логических системах посредством *if* диаграмм исследовано в работе [54]. Синтез и распараллеливание сумматоров и умножителей методами частичной логики посредством *if* диаграмм решений предложен в работах [56, 59, 67, 106]. Синтез сложных частично определенных логических проектов путем минимизации систем *if* диаграмм предложен в работе [59]. Сокращению мощности, потребляемой цифровой схемой, представленной и преобразованной посредством *if* диаграмм решений, посвящены работы [101, 102, 104]. Использованию двоичных диаграмм решений для верификации цифровой аппаратуры посвящены работы [10, 24, 59, 83]. Предложенные автором и изложенные в книге методы частичной и смешанной логик, а также диаграмм решений использованы в системах проектирования и оптимизации цифровых устройств и компьютерных программ [49-71, 93-108].

Глава 10. ЛОГИЧЕСКИЙ ВЫВОД В ЧАСТИЧНОЙ ЛОГИКЕ И РЕШЕНИЕ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

В настоящей главе исследуется *частичная логика*, обобщающая логику высказываний и логику предикатов для не полностью специфицированных областей. В частичной логике рассматриваются три значения ложь, истина и не определенное значение *don't care* (*dc*) вместо двух значений ложь и истина в традиционных логиках. Частичные операции, формулы, законы и правила вывода составляют базис для представления знаний и дедуктивного вывода, когда мир не является полностью определенным. *Принцип резолюций Робинсона* обобщается на случай, когда клозы принимают неопределенное значение *dc*. Предлагаются методы перехода от частичного дедуктивного вывода к дедуктивному выводу в логике предикатов первого порядка. Представленные теоретические результаты иллюстрируются примерами. Предложенные модели и методы ускоряют процесс дедуктивного вывода и сокращают число клозов, необходимых для представления знаний.

10.1. Методы представления знаний

Использование технологии построения логических систем, базирующихся на знаниях, упрощает моделирование широкого класса сложных закономерностей и ситуаций посредством символических рассуждений и упрощает задачу формализации сложных понятий в слабо структурированных областях.

В научной и коммерческой сферах разработан и известен ряд систем и сред, базирующихся на *знаниях*. Эти системы и среды используют модели и технологии такие, как правила шаблон-действие, фреймы, различные варианты подключения процедур и другие. В них присутствуют не только внутренние структуры выбранных парадигм, но интерфейсные средства, которые “понимают” и могут манипулировать этими структурами.

Логические системы составляют теоретический базис для технологий, базирующихся на знаниях. Логические подходы могут быть классифицированы следующим образом:

- обоснование множества истинностных значений в зависимости от проблемной области и решаемых задач;
- методы конструирования высказываний и определение семантики высказываний;
- методы рассуждений.

В зависимости от множества истинностных значений, построены и используются следующие логические системы:

- двухзначная булева логика с множеством истинностных значений ложь, истина;
- многозначные логики с конечным или бесконечным множеством T элементов;
- логика, базирующаяся на использовании вероятностей, значения которых принадлежат интервалу $[0...1]$;
- нечеткая логика с множеством истинностных значений, состоящих из нечетких подмножеств множества T ;
- другие логики.

В зависимости от методов конструирования высказываний и методов определения их семантики, построены и используются следующие логические системы:

- двухзначные логики высказываний, предикатов первого и предикатов более высокого порядка;
- многозначные логики высказываний, предикатов первого и предикатов более высокого порядка;
- логики высказываний и предикатов, основанные на использовании вероятностей и нечетких множеств;
- другие логики.

Существующие логики используют следующие методы рассуждений:

- правила вывода (modus ponens и другие);
- принцип резолюций;
- правило Байеса;
- решение нелинейных уравнений в нечеткой логике;
- другие методы.

В системах, основанных на знаниях, ключевой идеей является максимальное отделение знаний предметной области от процедур манипулирования этими знаниями. Представление знаний есть

множество соглашений о формализации закономерностей предметной области. Результатом исследований в рамках направления искусственный интеллект являются эффективные способы и инструменты описания различных частей реального мира, а также методы решения нетривиальных задач из практически значимых предметных областей путем выполнения механических рассуждений, которые способен реализовать компьютер.

Принцип отделения знаний от процедур решения задач лежит в основе многих систем, удовлетворяющих требованиям прозрачности и расширяемости. Независимые схемы представления знаний обладают свойствами простоты, ясности, наращиваемости. К ним относятся логические представления, семантические сети, системы продукций, фреймы.

Вместо описания вычислений в виде предопределенной последовательности выполняемых операций, *продукционные системы* трактуют вычисление как процесс применения правил преобразования в порядке, определяемом обрабатываемыми данными. Классическая продукционная система состоит из трех основных компонентов: глобальной базы данных, базы продукций, интерпретатора продукций.

Фрейм представляет знания путем описания отношений и связей между многочисленными фактами. Он кодирует знания об объекте, включая в код не только свойства (слоты) и значения, но и ссылки на другие фреймы и процедуры.

В логической системе знаний закономерности представляются логическими суждениями. Чаще всего используется логика предикатов первого порядка или ее вариации. Этот способ представления хорошо гармонирует с процедурой вывода, основанной на доказательстве теорем. Строгость логических рассуждений является преимуществом в точной спецификации того, что известно и каким образом знания будут использованы. В дополнение к логике высказываний и предикатов, многозначная, временная и другие логики являются важным инструментом в решении разнообразных задач компьютерных наук и искусственного интеллекта. Недостатком является трудность оперирования неточностью и неопределенностью в процессе рассуждений.

В течение многих лет научные исследования по искусственному интеллекту были сфокусированы на нечетких рассуждениях. Эвристики являются важнейшим средством решения интеллектуальных задач, поскольку точные математические методы применимы к ограниченному кругу проблем.

Очень часто мы имеем дело с не полностью специфицированными областями, где высказывания могут быть сформулированы лишь для части ситуаций. В этой главе мы предлагаем частичную логику, построенную на базе результатов, полученных при построении смешанной алгебры. Частичная логика обобщает традиционные логики высказываний и предикатов первого порядка. Ключевой результат работы Робинсона обобщается нами в частичной логике. Хотя понятие неопределенности, базирующееся на значении dc , используется уже многие годы, новым результатом является то, что частичная логика позволяет эффективно манипулировать частично определенными клозами в процессе логического вывода.

10.2. Частичная логика предикатов первого порядка

Сначала остановимся на определении синтаксиса и семантики *частичной логики предикатов первого порядка*. Затем рассмотрим метод преобразования частичной формулы к форме, которая может быть эффективно использована в процессе логического вывода.

Синтаксис. Алфавит нашего языка состоит из:

- 1) разделителей ‘;’, ‘(’, ‘)’, ‘|’;
- 2) предметных переменных из множества V ;
- 3) полностью определенных логических операторов \neg , \wedge , \vee , \rightarrow , \oplus и других;
- 4) частичных логических операторов \sim , $\&$, $+$, \Rightarrow , \Leftrightarrow и других;
- 5) функциональных символов $f_1^{n_1}, \dots, f_i^{n_i}, \dots$ степени $n \geq 0$; функциональный символ f^0 степени $n=0$ является индивидуальной константой;
- 6) предикатных символов $P_1^{n_1}, \dots, P_i^{n_i}, \dots$ степени $n \geq 0$; предикатный символ P^0 степени $n=0$ является пропозициональной переменной,

7) полностью определенного квантора общности \forall , полностью определенного квантора существования \exists , *частичного квантора общности* $\underline{\forall}$, *частичного квантора существования* $\underline{\exists}$.

Символы алфавита используются для построения выражений, состоящих из *термов, атомарных формул, литералов*.

Термы:

- 1) индивидуальная константа f^{β} есть терм;
- 2) предметная переменная $v \in V$ есть терм;
- 3) функциональный символ $f_i^n(t_1, \dots, t_n)$, за которым в скобках следуют n термов t_i , есть терм.

Атомарные формулы:

- 1) пропозициональная переменная P^{β} есть атомарная формула;
- 2) предикатный символ $P_i^n(t_1, \dots, t_n)$, за которым в скобках следуют n термов t_i , есть атомарная формула.

Литералы:

- 1) атомарная формула является литералом;
- 2) если A есть атомарная формула, то $\neg A$ есть литерал.

Литерал, не содержащий переменных, называется *базовым литералом*. Два литерала A и $\neg A$ составляют дополняющую пару.

Полностью определенные формулы:

- 1) атомарная формула есть полностью определенная формула;
- 2) если F является полностью определенной формулой, то $(\neg F)$ есть полностью определенная формула;
- 3) если F и G являются полностью определенными формулами, то $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, $(F \oplus G)$ и т.д. являются полностью определенными формулами;
- 4) если F есть полностью определенная формула и x есть предметная переменная, то $(\forall x)F$ и $(\exists x)F$ являются полностью определенными формулами.

Иногда мы будем опускать скобки, принимая во внимание приоритет операций.

Частично определенные формулы:

- 1) пара (FG) полностью определенных формул является частично определенной формулой;

2) если R есть частично определенная формула, то $(\sim R)$ есть частично определенная формула;

3) если R и Q являются частично определенными формулами, то $(R \& Q)$, $(R + Q)$, $(R \rightarrow Q)$, $(R \odot Q)$ и т.д. являются частично определенными формулами;

4) если R есть частично определенная формула и x есть предметная переменная, то $(\forall x)R$ и $(\exists x)R$ являются частично определенными формулами.

Семантика. Семантика нашего языка определяется следующим образом. С каждым полностью определенным логическим оператором ассоциируется булева функция $B^n \rightarrow B$ где n есть арность оператора.

Частично определенная булева функция $B \times B \rightarrow M$, описываемая табл.10.1, устанавливает семантику пары $(F|G)$ полностью определенных формул F и G . С каждым частично определенным оператором мы ассоциируем частичную функцию $M^n \rightarrow M$ из табл.2.2 и табл.2.6. Семантика частичных кванторов общности $(\forall x)R$ и существования $(\exists x)R$ описывается функциями из табл.10.2. Эти функции являются отображениями $M^+ \rightarrow M$, где M^+ есть множество всех подмножеств множества M , исключая пустое подмножество. Семантика обыкновенных кванторов общности $(\forall x)R$ и существования $(\exists x)R$ определяется традиционным способом.

Пусть D есть множество определения (предметное множество). Мы назначаем каждому n -местному функциональному символу f_i^n функцию $D^n \rightarrow D$ и назначаем каждому n -местному предикатному символу P_i^n функцию $D^n \rightarrow B$. Каждая переменная из множества V отображается на элементы множества D . Интерпретирующая

Таблица 10.1

Частично определенная функция $(F|G)$

| F | G | $(F G)$ |
|---|---|---------|
| 0 | 0 | dc |
| 0 | 1 | 0 |
| 1 | 0 | dc |
| 1 | 1 | 1 |

Таблица 10.2

Функции для частичных кванторов $(\forall x)R$ and $(\exists x)R$

| Множество значений формулы R | $(\forall x)R$ | $(\exists x)R$ |
|--------------------------------|----------------|----------------|
| {0} | 0 | 0 |
| {1} | 1 | 1 |
| {dc} | dc | dc |
| {0,1} | 0 | 1 |
| {0,dc} | 0 | dc |
| {1,dc} | dc | 1 |
| {0,1,dc} | 0 | 1 |

функция $v_l(R)$ вычисляет значение частично определенной формулы R для интерпретации l . Интерпретация l удовлетворяет частично определенной формуле R тогда и только тогда, когда $v_l(R) \in \{1, dc\}$. Формула R называется невыполнимой тогда и только тогда, когда не существует интерпретации, удовлетворяющей R .

10.3. Преобразование частичной формулы

Предлагается метод преобразования любой заданной частичной формулы к следующей специальной форме:

$$(F|G), \quad (10.1)$$

где F и G есть полностью определенные формулы без каких-либо кванторов, причем каждая формула представляется конъюнкцией высказываний, а каждое высказывание есть дизъюнкция литералов. Процедура преобразований включает следующие шаги:

- удаление полностью определенных логических операторов за исключением отрицания, конъюнкции и дизъюнкции;
- введение уникальной предметной переменной для каждого квантора;
- сокращение зоны действия операций отрицания;
- удаление кванторов существования;
- перемещение кванторов общности в префикс формулы;
- удаление префикса формулы;
- удаление всех частично определенных операторов;

• преобразование частичной формулы к паре множеств полностью определенных кловов.

Проиллюстрируем работу каждого шага процедуры преобразования на следующем примере частичной формулы:

$$\begin{aligned} & (\forall x)((\forall y)(\neg Q(c,x) \rightarrow R(a,y) \mid Q(y,b)) \& \\ & (\exists z)\sim(\exists y)(R(y,z) \vee P(x) \mid P(b))), \end{aligned} \quad (10.2)$$

где x , y и z – предметные переменные; a , d и c – индивидуальные константы; P , Q и R – предикатные символы.

Шаг 1 – удаление полностью определенных логических операторов, исключая \neg , \wedge , \vee . Для удаления используются законы логики высказываний. Выполняя удаление оператора импликации из формулы (10.2), получаем новую формулу:

$$\begin{aligned} & (\forall x)((\forall y)(Q(c,x) \vee R(a,y) \mid Q(y,b)) \& \\ & (\exists z)\sim(\exists y)(R(y,z) \vee P(x) \mid P(b))). \end{aligned} \quad (10.3)$$

Шаг 2 – введение уникальной переменной для каждого квантора. Формула (10.3) использует три переменные x , y и z . Каждая из них связана квантором. Переменная y используется дважды. Это является причиной переименования переменной y под квантором существования, в результате получаем:

$$\begin{aligned} & (\forall x)((\forall y)(Q(c,x) \vee R(a,y) \mid Q(y,b)) \& \\ & (\exists z)\sim(\exists v)(R(v,z) \vee P(x) \mid P(b))). \end{aligned} \quad (10.4)$$

Шаг 3 – сокращение зоны действия полностью определенной и частичной операций отрицания \neg и \sim . Для этого используются законы двойного отрицания и законы де Моргана для конъюнкции и дизъюнкции. В дополнение к ним необходимым является использование следующих законов:

$$\sim(F \mid G) = (\neg F \mid G), \quad (10.5)$$

$$\sim(\forall x)(F \mid G) = (\exists x)\sim(F \mid G), \quad (10.6)$$

$$\sim(\exists x)(F \mid G) = (\forall x)\sim(F \mid G). \quad (10.7)$$

Закон (10.5) доказывается легко. Доказательство законов (10.6) и (10.7) приведено в табл.10.3. Идентичность столбцов 5 и 7 доказывает (10.6), идентичность столбцов 4 и 8 доказывает (10.7).

Таблица 10.3

Доказательство законов (10.6) и (10.7)

| (F G) | ~(F G) | ($\exists x$) (F G) | ~($\exists x$) (F G) | ($\exists x$) ~(F G) | ($\forall x$) (F G) | ~($\forall x$) (F G) | ($\forall x$) ~(F G) |
|----------|----------|--------------------------|---------------------------|---------------------------|--------------------------|---------------------------|---------------------------|
| {0} | {1} | 0 | 1 | 1 | 0 | 1 | 1 |
| {1} | {0} | 1 | 0 | 0 | 1 | 0 | 0 |
| {dc} | {dc} | dc | dc | dc | dc | dc | dc |
| {0,1} | {0,1} | 1 | 0 | 1 | 0 | 1 | 0 |
| {0,dc} | {1,dc} | dc | dc | 1 | 0 | 1 | dc |
| {1,dc} | {0,dc} | 1 | 0 | dc | dc | dc | 0 |
| {0,1,dc} | {0,1,dc} | 1 | 0 | 1 | 0 | 1 | 0 |

Формула (10.4) содержит один оператор отрицания. Применяя к (10.4) законы (10.7), (10.5) и один из законов де Моргана, получаем следующую преобразованную формулу:

$$(\forall x)((\forall y)(Q(c,x) \vee R(a,y) \mid Q(y,b)) \& (\exists z)(\forall v)(\neg R(v,z) \wedge \neg P(x) \mid P(b))). \quad (10.8)$$

Шаг 4 – удаление полностью и частично определенных кванторов существования путем введения функций Сколема вместо предметных переменных, связанных кванторами. Если квантор существования стоит после k кванторов общности, функция Сколема имеет арность k . Если перед квантором существования нет кванторов общности, арность функции равна 0. Анализируя формулу (10.8), приходим к выводу, что частичный квантор существования $\exists z$ стоит после частичного квантора общности $\forall x$. Следовательно, переменная z заменяется функциональным символом $f(x)$:

$$(\forall x)((\forall y)(Q(c,x) \vee R(a,y) \mid Q(y,b)) \& (\forall v)(\neg R(v,f(x)) \wedge \neg P(x) \mid P(b))). \quad (10.9)$$

Шаг 5 – перемещение полностью и частично определенных кванторов общности в префикс формулы. Перемещение выполняется посредством следующих законов:

$$((\forall x)F)G = (\forall x)(FG) \quad (10.10)$$

$$((\forall x)R) \& Q = (\forall x)(R \& Q) \quad (10.11)$$

$$((\forall x)R) + Q = (\forall x)(R + Q). \quad (10.12)$$

Закон (10.10) осуществляет перемещение полностью

Таблица 10.4

Доказательство закона (10.10)

| F | G | $(\forall x)F$ | $((\forall x)F G)$ | $(F G)$ | $(\forall x)(F G)$ |
|-------|---|----------------|--------------------|---------|--------------------|
| {0} | 0 | 0 | dc | {dc} | dc |
| {1} | 0 | 1 | dc | {dc} | dc |
| {0,1} | 0 | 0 | dc | {dc} | dc |
| {0} | 1 | 0 | 0 | {0} | 0 |
| {1} | 1 | 1 | 1 | {1} | 1 |
| {0,1} | 1 | 0 | 0 | {0,1} | 0 |

определенного квантора, относящегося к полностью определенной формуле F , в префикс частично определенной формулы $(F|G)$, при этом полностью определенный квантор заменяется частично определенным квантором. Закон учитывает предварительное переименование переменных, многократно используемых в формуле. Он доказывается идентичностью столбцов 4 и 6 табл.10.4.

Законы (10.11) и (10.12) осуществляют перемещение частных кванторов общности из операндов частичных операций конъюнкции и дизъюнкции вперед операций. Закон (10.11) доказывается идентичностью столбцов 4 и 6 табл.10.5, закон (10.12) - идентичностью столбцов 7 и 9 этой же таблицы.

Для перемещения кванторов общности в префикс формулы необходим еще один закон:

$$(F|(\forall x)G) = (\forall x)(F|G). \quad (10.13)$$

С целью доказательства закона строим табл.10.6. Анализируя идентичность столбцов 5 и 7, замечаем, что в строке 3 формула $(F|(\forall x)G)$ принимает значение dc , а формула $(\forall x)(F|G)$ принимает значение 0. Переход от значения dc к значения 0 является допустимым, следовательно, закон (10.13) может быть использован в процессе преобразований. Применяя закон (10.11) к формуле (10.9), получаем:

$$(\forall x)((\forall y)(\forall v)(Q(c,x) \vee R(a,y) | Q(y,b)) \& (\neg R(v,f(x)) \wedge \neg P(x) | P(b))). \quad (10.14)$$

Шаг 6 – удаление префикса. Удаление приводит к тому, что формула состоит только из матрицы. Удаление префикса из формулы (10.14) дает

Таблица 10.5

Доказательство законов (10.11) и (10.12)

| R | Q | $(\forall x)$ R | $((\forall x)$ R)&Q | R&Q | $(\forall x)$ (R&Q) | $((\forall x)$ R)+Q | R+Q | $(\forall x)$ (R+Q) |
|----------|----|--------------------|------------------------|----------|------------------------|------------------------|----------|------------------------|
| {0} | 0 | 0 | 0 | {0} | 0 | 0 | {0} | 0 |
| {1} | 0 | 1 | 0 | {0} | 0 | 1 | {1} | 1 |
| {dc} | 0 | dc | 0 | {0} | 0 | dc | {dc} | dc |
| {0,1} | 0 | 0 | 0 | {0} | 0 | 0 | {0,1} | 0 |
| {0,dc} | 0 | 0 | 0 | {0} | 0 | 0 | {0,dc} | 0 |
| {1,dc} | 0 | dc | 0 | {0} | 0 | dc | {1,dc} | dc |
| {0,1,dc} | 0 | 0 | 0 | {0} | 0 | 0 | {0,1,dc} | 0 |
| {0} | 1 | 0 | 0 | {0} | 0 | 1 | {1} | 1 |
| {1} | 1 | 1 | 1 | {1} | 1 | 1 | {1} | 1 |
| {dc} | 1 | dc | dc | {dc} | dc | 1 | {1} | 1 |
| {0,1} | 1 | 0 | 0 | {0,1} | 0 | 1 | {1} | 1 |
| {0,dc} | 1 | 0 | 0 | {0,dc} | 0 | 1 | {1} | 1 |
| {1,dc} | 1 | dc | dc | {1,dc} | dc | 1 | {1} | 1 |
| {0,1,dc} | 1 | 0 | 0 | {0,1,dc} | 0 | 1 | {1} | 1 |
| {0} | dc | 0 | 0 | {0} | 0 | dc | {dc} | dc |
| {1} | dc | 1 | dc | {dc} | dc | 1 | {1} | 1 |
| {dc} | dc | dc | dc | {dc} | dc | dc | {dc} | dc |
| {0,1} | dc | 0 | 0 | {0,dc} | 0 | dc | {1,dc} | dc |
| {0,dc} | dc | 0 | 0 | {0,dc} | 0 | dc | {dc} | dc |
| {1,dc} | dc | dc | dc | {dc} | dc | dc | {1,dc} | dc |
| {0,1,dc} | dc | 0 | 0 | {0,-} | 0 | dc | {1,dc} | dc |

Таблица 10.6

Доказательство закона (10.13)

| N | G | F | $(\forall x)G$ | $(F (\forall x)G)$ | $(F G)$ | $(\forall x)(F G)$ |
|---|-------|---|----------------|--------------------|---------|--------------------|
| 1 | {0} | 0 | 0 | dc | {dc} | dc |
| 2 | {1} | 0 | 1 | 0 | {0} | 0 |
| 3 | {0,1} | 0 | 0 | dc | {0,dc} | 0 |
| 4 | {0} | 1 | 0 | dc | {dc} | dc |
| 5 | {1} | 1 | 1 | 1 | {1} | 1 |
| 6 | {0,1} | 1 | 0 | dc | {1,dc} | dc |

$$(Q(c,x) \vee R(a,y) \mid Q(y,b)) \ \& \ (\neg R(v,f(x)) \wedge \neg P(x) \mid P(b)). \quad (10.15)$$

Шаг 7 – удаление частичных операторов и замена их на полностью определенные операторы. Для удаления используются законы (3.36)-(3.45). В результате получаем пару полностью определенных формул в форме (10.1). Применяя закон (3.36) к формуле (10.14), получаем:

$$\begin{aligned}
& (((Q(c,x) \vee R(a,y)) \wedge \neg R(v,f(x)) \wedge \neg P(x) \mid \\
& (Q(y,b) \vee P(b)) \wedge (Q(y,b) \vee \neg(\neg R(v,f(x)) \wedge \neg P(x))) \wedge \\
& (P(b) \vee \neg(Q(c,x) \vee R(a,y))))). \tag{10.16}
\end{aligned}$$

Шаг 8 – преобразование полностью определенных формул F и G в паре $(F|G)$ к конъюнкциям дизъюнкций литералов. Результатом являются два множества полностью определенных кловов. Выполняя этот шаг на формуле (10.16), получаем первое множество

$$\begin{aligned}
& 1) ((Q(c,x) \vee R(a,y)) \wedge \\
& 2) \neg R(v,f(x)) \wedge \\
& 3) \neg P(x) \tag{10.17}
\end{aligned}$$

и второе множество

$$\begin{aligned}
& 1) (Q(y,b) \vee P(b)) \wedge \\
& 2) (Q(y,b) \vee R(v,f(x)) \vee P(x)) \wedge \\
& 3) (P(b) \vee \neg Q(c,x)) \wedge \\
& 4) (P(b) \vee \neg R(a,y)). \tag{10.18}
\end{aligned}$$

Первое множество включает 3 клова, второе – 4 клова. Далее мы используем и другие формы представления частичных формул.

10.4. Представление не полностью определенных знаний

Кроме формы (10.1) мы также будем использовать для представления знаний частичную конъюнкцию фраз, являющихся *частичными кловами*:

$$Knowledge = C_1 \& \dots \& C_m, \tag{10.19}$$

$$C_i = (\forall x_1) \dots (\forall x_n) (P_{i,1} \& \dots \& P_{i,m}), \tag{10.20}$$

где P_{ij} представляется парой $(F|G)$ полностью определенных формул. Формулы F и G могут быть, в частности, литералами. Кванторы общности в (10.20) мы будем опускать. В результате частичный клов

$$(L_1|D_1) + \dots + (L_k|D_k) + (\neg L_{k+1}|D_{k+1}) + \dots + (\neg L_m|D_m) \tag{10.21}$$

содержит k положительных и $m-k$ отрицательных литералов в первой части пар. Пустой клов представляется в виде $(0|D_i)$.

Аналогично клозу Хорна в классической логике предикатов первого порядка мы вводим *частичный кюз Хорна* с одним положительным литералом:

$$(L|D) + (\neg L_1|D_1) + \dots + (\neg L_m|D_m) \quad (10.22)$$

Легко видеть, что частичный кюз (10.21) преобразуется к паре, в которой первой частью является полностью определенный кюз, второй частью является полностью определенная формула:

$$\begin{aligned} (L_1 \vee \dots \vee L_k \vee \neg L_{k+1} \vee \dots \vee \neg L_m | \\ D_1 \wedge \dots \wedge D_m \vee L_1 \wedge D_1 \vee \dots \vee L_k \wedge D_k \vee \neg L_{k+1} \wedge D_{k+1} \vee \dots \vee \neg L_m \wedge D_m). \end{aligned} \quad (10.23)$$

Частичный кюз Хорна (10.22) преобразуется в частичную импликацию:

$$\begin{aligned} (L|D) + (\neg L_1|D_1) + \dots + (\neg L_m|D_m) = \\ (L|D) + \sim((L_1|D_1) \& \dots \& (L_m|D_m)) = \\ (L|D) \leftarrow ((L_1|D_1) \& \dots \& (L_m|D_m)). \end{aligned} \quad (10.24)$$

Импликацию (10.24) назовем *частичным правилом* (закономерностью). Одну пару $(L|D)$ будем называть *частичным фактом*.

10.5. Дедуктивный вывод в частичной логике

В работе Робинсона дана формализация логики предикатов первого порядка, специально разработанная как инструмент компьютерного доказательства теорем. Единое *правило логического вывода* предпочтительнее набору правил по прагматическим и философским причинам. Такое правило базируется на принципе резолюций Робинсона. Оно удобно в программной и аппаратной реализации и составляет полную систему логики предикатов первого порядка. Мы обобщаем принцип резолюций на не полностью определенный *логический вывод*.

Мы используем механизм дедуктивного вывода, базирующийся на интерпретации Эрбрана, невыполнимости и опровержении. Множество клозов выполнимо, если существует модель, не содержащая взаимно исключающей пары литералов L и $\neg L$.

Цели и теоремы. Пусть дано множество $S = \{C_1, \dots, C_m\}$ частичных кловов (кусочков знаний) и целевая частичная формула (цель) R . Мы формулируем следующую теорему:

$$C_1 \& \dots \& C_m \rightarrow R. \quad (10.25)$$

Формула R описывает новый кусочек знаний, который должен быть выведен из кловов C_1, \dots, C_m . С целью доказательства теоремы мы находим модель, представленную множеством значений предметных переменных и удовлетворяющую формуле (10.25). Формула выполняется, если она принимает значение 1 или значение *don't care* (dc), которое может быть заменено на значение 1.

Доказательство путем опровержения. Для доказательства выполнимости формулы (10.25) мы используем процедуру опровержения и доказываем, что следующая формула невыполнима:

$$C_1 \& \dots \& C_m \& \sim R. \quad (10.26)$$

Начиная с кловов C_1, \dots, C_m и $\sim R$ и применяя принцип резолюций, мы генерируем последовательность дополнительных кловов (*резольвент*) до тех пор, пока не появится пустой клов, включающий одну пару вида $(0|L)$.

Обобщение принципа резолюций. Сначала мы обобщим принцип резолюций для базовых частичных кловов. Пусть даны два частичных клова

$$C_1 = (L_1|D_1) + (L_2|D_2) \quad (10.27)$$

и

$$C_2 = (\neg L_1|D_3) + (L_3|D_4), \quad (10.28)$$

где L_1 и $\neg L_1$ – взаимоисключающая пара базовых литералов. Следующий дополнительный частичный клов (*резольвента*) добавляется к системе исходных кловов:

$$C = (0|D_1 \wedge D_3 \vee \neg L_1 \wedge D_1 \vee L_1 \wedge D_3) + (L_2|D_2) + (L_3|D_4). \quad (10.29)$$

Пары $(L_2|D_2)$ и $(L_3|D_4)$ переносятся из исходных кловов C_1 и C_2 в резольвенту в неизменном виде. В резольвенте C появляется новая пара с истинностным значением 0 в первой части. Чтобы доказать логическое следование C из C_1 и C_2 , мы должны показать, что

$C_1 \& C_2$ выполняется тогда и только тогда, когда выполняется $C_1 \& C_2 \& C$. Частичная конъюнкция $C_1 \& C_2$ двух частичных кловов преобразуется следующим образом:

$$\begin{aligned}
C_1 \& C_2 &= ((L_1|D_1)+(L_2|D_2)) \& ((\neg L_1|D_3)+(L_3|D_4)) = \\
&(L_1 \vee L_2 | D_1 \wedge D_2 \vee L_1 \wedge D_1 \vee L_2 \wedge D_2) \& \\
&(\neg L_1 \vee L_3 | D_3 \wedge D_4 \vee \neg L_1 \wedge D_3 \vee L_3 \wedge D_4) = \\
&((L_1 \vee L_2) \wedge (\neg L_1 \vee L_3) | (D_1 \wedge D_2 \vee L_1 \wedge D_1 \vee L_2 \wedge D_2) \wedge \\
&(D_3 \wedge D_4 \vee \neg L_1 \wedge D_3 \vee L_3 \wedge D_4) \vee D_1 \wedge D_2 \wedge \neg L_1 \wedge \neg L_2 \vee \\
&D_3 \wedge D_4 \wedge L_1 \wedge \neg L_3) = \\
&((L_1 \vee L_2) \wedge (\neg L_1 \vee L_3) | D_1 \wedge D_2 \wedge D_3 \wedge D_4 \vee D_1 \wedge D_2 \wedge D_3 \wedge \neg L_1 \vee \\
&D_1 \wedge D_2 \wedge D_4 \wedge L_3 \vee D_1 \wedge D_2 \wedge \neg L_1 \wedge \neg L_2 \vee \\
&D_1 \wedge D_3 \wedge D_4 \wedge L_1 \vee D_1 \wedge D_4 \wedge L_1 \wedge L_3 \vee \\
&D_2 \wedge D_3 \wedge D_4 \wedge L_2 \vee D_2 \wedge D_3 \wedge \neg L_1 \wedge L_2 \vee \\
&D_2 \wedge D_4 \wedge L_2 \wedge L_3 \vee D_3 \wedge D_4 \wedge L_1 \wedge \neg L_3). \tag{10.30}
\end{aligned}$$

Резольвента (10.29) преобразуется следующим образом:

$$\begin{aligned}
C &= (0 | D_1 \wedge D_3 \vee \neg L_1 \wedge D_1 \vee L_1 \wedge D_3) + (L_2 | D_2) + (L_3 | D_4) = \\
&(L_2 \vee L_3 | (D_1 \wedge D_3 \vee \neg L_1 \wedge D_1 \vee L_1 \wedge D_3) \wedge D_2 \wedge D_4 \vee \\
&(L_2 \wedge D_2) \vee (L_3 \wedge D_4)) = \\
&(L_2 \vee L_3 | D_1 \wedge D_2 \wedge D_3 \wedge D_4 \vee D_1 \wedge D_2 \wedge D_4 \wedge \neg L_1 \vee \\
&D_2 \wedge D_3 \wedge D_4 \wedge L_1 \vee L_2 \wedge D_2 \vee L_3 \wedge D_4). \tag{10.31}
\end{aligned}$$

Выполняя частичную операцию конъюнкция над операндом $C_1 \& C_2$, представленным формулой (10.30), и операндом C , представленным формулой (10.31), получаем:

$$\begin{aligned}
C_1 \& C_2 \& C &= ((L_1 \vee L_2) \wedge (\neg L_1 \vee L_3) \wedge (L_2 \vee L_3) | \\
&D_1 \wedge D_2 \wedge D_3 \wedge D_4 \vee D_1 \wedge D_2 \wedge D_3 \wedge \neg L_1 \vee D_1 \wedge D_2 \wedge D_4 \wedge L_3 \vee \\
&D_1 \wedge D_2 \wedge \neg L_1 \wedge \neg L_2 \vee D_1 \wedge D_3 \wedge D_4 \wedge L_1 \vee D_1 \wedge D_4 \wedge L_1 \wedge L_3 \vee \\
&D_2 \wedge D_3 \wedge D_4 \wedge L_2 \vee D_2 \wedge D_3 \wedge \neg L_1 \wedge L_2 \vee D_2 \wedge D_4 \wedge L_2 \wedge L_3 \vee \\
&D_3 \wedge D_4 \wedge L_1 \wedge \neg L_3). \tag{10.32}
\end{aligned}$$

Сравнивая формулы (10.30) и (10.32), легко видеть, что функции области определенности, представленные правыми частями, являются идентичными. Левая часть формулы (10.32) содержит дополнительную полностью определенную резольвенту по сравнению с левой частью формулы (10.30). Следовательно, формула (10.30)

выполняется тогда и только тогда, когда выполняется формула (10.32).

Пример 10.1. Задана частичная формула

$$(A \vee B | \neg C) \& (\neg A | B) \& (C \vee D | A) \& (\neg D | C). \quad (10.33)$$

Докажем, что формула (10.33) является невыполнимой. Формула разбивается на частичные клозы следующим образом:

1. $(A \vee B | \neg C)$ - исходный кюз
2. $(\neg A | B)$ - исходный кюз
3. $(C \vee D | A)$ - исходный кюз
4. $(\neg D | C)$ - исходный кюз

Применяя обобщенное правило резолюций к парам частичных клозов, генерируем частичные резольвенты до тех пор, пока не получим пустой частичный кюз:

5. $(B | B \wedge \neg C)$ - резольвента клозов 1 и 2
6. $(1 | B \wedge \neg C) = (\neg C | B \wedge \neg C)$ - из клоза 5
7. $(D | A \wedge (B \vee D) \wedge (\neg C \vee D))$ - резольвента клозов 3 и 6
8. $(0 | A \wedge D \wedge C)$ - резольвента клозов 4 и 7

Кюз 8 является пустым частичным кюзом, следовательно формула (10.33) невыполнима. \square

Обобщение резольвенты. Пусть исходные частичные клозы имеют следующий вид:

$$C_1 = (L_1 \vee L_2 | D_1) \quad (10.34)$$

и

$$C_2 = (\neg L_1 \vee L_3 | D_2). \quad (10.35)$$

Тогда резольвента приобретает вид:

$$C = (L_2 \vee L_3 | D_1 \wedge D_2 \vee L_2 \wedge D_1 \vee L_3 \wedge D_2). \quad (10.36)$$

Чтобы доказать (10.36), представим клозы (10.34) и (10.35) в виде $C_1 = (L_1 | D_1) + (L_2 | D_1)$ и $C_2 = (\neg L_1 | D_2) + (L_3 | D_2)$. Используя выражение (10.29), получаем:

$$C = (0 | D_1 \wedge D_2 \vee \neg L_1 \wedge D_1 \vee L_1 \wedge D_2) \vee (L_2 | D_1) \vee (L_3 | D_2) = \\ (L_2 \vee L_3 | D_1 \wedge D_2 \vee L_2 \wedge D_1 \vee L_3 \wedge D_2). \quad (10.37)$$

В случае, когда частичные клозы не являются базовыми, должен использоваться механизм унификации термов перед применением правила резолюций.

Правило резолюций для клозов в импликативной форме. Имея множество частичных правил (клозов в импликативной форме) и целевую формулу, мы, используя обобщенный принцип резолюций, пошагово преобразуем цель до тех пор, пока не получим пустой частичный клоз. Без потери общности положим, что текущая цель представляется следующей импликацией:

$$(0 \leftarrow L_1 \wedge L_2 | D_1). \quad (10.38)$$

Положим также, что клоз, используемый для преобразования цели, которую мы имеем на текущем шаге, к цели, получаемой на следующем шаге, имеет форму правила:

$$(L_1 \leftarrow L_3 | D_2). \quad (10.39)$$

Преобразуем цель (10.38) и правило (10.39) к форме дизъюнктов $(\neg L_1 \vee \neg L_2 | D_1)$ и $(L_1 \vee \neg L_3 | D_2)$. Применяя к ним правило резолюций, получаем следующую резольвенту

$$(\neg L_3 \vee \neg L_2 | D_1 \wedge D_2 \vee \neg L_2 \wedge D_1 \vee \neg L_3 \wedge D_2),$$

которая затем трансформируется в цель следующего шага:

$$(0 \leftarrow L_3 \wedge L_2 | D_1 \wedge D_2 \vee \neg L_2 \wedge D_1 \vee \neg L_3 \wedge D_2). \quad (10.40)$$

Первая часть новой цели (10.40) может быть упрощена посредством законов (3.47-3.61).

Пусть теперь клоз, используемый для построения резольвенты, является фактом

$$(L_1 | D_2).$$

Тогда новая цель приобретает следующий более простой вид:

$$(0 \leftarrow L_2 | D_1 \wedge (D_2 \vee \neg L_2)).$$

10.6. Переход от частичной к полностью определенной логике

Для того, что использовать предложенный аппарат на практике для представления знаний и решения задач путем дедуктивного вывода, должно быть разработано соответствующее программное или аппаратное обеспечение. Предлагается метод перехода от частичной логики к полностью определенной логике предикатов первого порядка, состоящий из четырех шагов.

1. Преобразование частичной формулы к паре $(F|G)$ полностью определенных формул.

2. Замена первой части пары F эквивалентной полностью определенной формулой $Value_Part$, выбираемой из диапазона

$$F \wedge G \subseteq Value_Part \subseteq F \vee \neg G. \quad (10.41)$$

Обновленная пара записывается в виде:

$$(Value_Part | G). \quad (10.42)$$

3. Переход от пары $(F|G)$ с функцией области определенности G к паре с функцией области определенности 1, замена всех частичных кванторов общности, которые были опущены, традиционными полностью определенными кванторами общности:

$$(Value_Part | 1). \quad (10.43)$$

4. Преобразование $Value_Part$ к множеству кловов, представленных средствами традиционной логики предикатов первого порядка:

$$C_1 \wedge \dots \wedge C_m. \quad (10.44)$$

Нами предлагается три метода построения $Value_Part$.

- использование обоих значений 0 и 1 для доопределения значения dc (метод 1)
- замена всех вхождений значения dc значением 1 (метод 2)
- замена всех вхождений значения dc значением 0 (метод 3).

Переход к полностью определенной логике методом 1. Когда используются оба значения 0 и 1 для доопределения значения dc ,

Value_Part выбирается из диапазона функций (10.41). В частности, может быть использована функция F .

Пример 10.2. Он является продолжением примера 10.1. Частичная формула (10.33) преобразуется к полностью определенной конъюнкции дизъюнктов:

$$\begin{aligned} & (A \vee B | 1) \& (\neg A | 1) \& (C \vee D | 1) \& (\neg D | 1) = \\ & ((A \vee B) \wedge (\neg A) \wedge (C \vee D) \wedge (\neg D) | 1). \end{aligned} \quad (10.45)$$

Следующее множество из четырех полностью определенных клозов извлекается из формулы (10.45):

1. $A \vee B$
2. $\neg A$
3. $C \vee D$
4. $\neg D$.

(10.46)

Хорошо видно, что из множества клозов (10.46) невозможно логически вывести пустой клоз.

Переход к полностью определенной логике методом 2. Если используется только значение 1 для доопределения значения dc , то левая часть пары $(F|G)$ расширяется и заменяется на $Value_Part = F \vee \neg G$.

Пример 10.3. Преобразуем частичную формулу (10.33) к полностью определенной конъюнкции дизъюнктов путем замены левой части каждой из четырех слагаемых дизъюнкцией левой части и инверсии правой части:

$$\begin{aligned} & (A \vee B \vee C | \neg C) \& (\neg A \vee \neg B | B) \& (C \vee D \vee \neg A | A) \& (\neg D \vee \neg C | C) = \\ & (A \vee B \vee C | 1) \& (\neg A \vee \neg B | 1) \& (C \vee D \vee \neg A | 1) \& (\neg D \vee \neg C | 1) = \\ & ((A \vee B \vee C) \wedge (\neg A \vee \neg B) \wedge (C \vee D \vee \neg A) \wedge (\neg D \vee \neg C) | 1) = \\ & (A \vee B \vee C) \wedge (\neg A \vee \neg B) \wedge (C \vee D \vee \neg A) \wedge (\neg D \vee \neg C). \end{aligned} \quad (10.47)$$

Из формулы (10.47) извлекается следующий набор полностью определенных клозов:

1. $A \vee B \vee C$
2. $\neg A \vee \neg B$
3. $C \vee D \vee \neg A$
4. $\neg D \vee \neg C$.

(10.48)

Анализ кловов (10.48) показывает, что из них нельзя логически вывести пустой кюз.

Переход к полностью определенной логике методом 3. Если используется только значение 0 для доопределения значения dc , то левая часть пары $(F|G)$ сужается и заменяется на $Value_Part = F \wedge G$.

Пример 10.4. Преобразуем теперь частичную формулу (10.33) к полностью определенной конъюнкции дизъюнктов путем замены левой части каждой из четырех слагаемых конъюнкцией левой и правой частей:

$$\begin{aligned} & ((A \vee B) \wedge \neg C | \neg C) \& (\neg A \wedge B | B) \& ((C \vee D) \wedge A | A) \& (\neg D \wedge C | C) = \\ & ((A \vee B) \wedge \neg C | 1) \& (\neg A \wedge B | 1) \& ((C \vee D) \wedge A | 1) \& (\neg D \wedge C | 1) = \\ & ((A \vee B) \wedge \neg C \wedge \neg A \wedge B \wedge (C \vee D) \wedge A \wedge \neg D \wedge C | 1). \end{aligned} \quad (10.49)$$

Из формулы (10.49) извлекается восемь полностью определенных кловов:

1. $A \vee B$
 2. $\neg C$
 3. $\neg A$
 4. B
 5. $C \vee D$
 6. A
 7. $\neg D$
 8. C .
- (10.50)

Легко видно, что достаточно сгенерировать только одну резольвенту, чтобы получить пустой кюз из набора кловов (10.50). В частности, литералы C и $\neg C$ составляют взаимно исключающую пару.

Из трех исследованных методов, третий метод является предпочтительным и может быть использован на практике. Если система частичных кловов выполнима, то он находит решение. При этом он ускоряет получение пустого кюза. Кроме того, в этом случае процедура перехода от частичных кловов к полностью определенным кюзам наиболее эффективна.

Трансляция частичных импликаций. Рассмотрим следующее правило в форме частичной импликации:

$$(A|1) \leftarrow (C|Q). \quad (10.51)$$

В паре $(A|1)$ функция области определенности равна 1. Следовательно, значение пары определяется литералом A . Пара $(C|Q)$ принимает одно из трех значений: ложь, истина и *don't care*. Значение *don't care* может быть заменено на ложь и истину произвольным образом. В результате строится одна из полностью определенных функций, принадлежащих диапазону от $C \wedge Q$ до $C \vee \neg Q$. Для того, чтобы построить все возможные решения для заданной цели мы выбираем функцию $C \vee \neg Q$. Частичная формула (10.51) преобразуется в этом случае к полностью определенной импликации

$$A \leftarrow (C \vee \neg Q). \quad (10.52)$$

Легко показать, что выражение эквивалентно конъюнкции двух импликаций:

$$(A \leftarrow \neg Q) \wedge (A \leftarrow C). \quad (10.53)$$

Следовательно, одна импликация в частичной логике транслируется в две импликации в полностью определенной логике предикатов. Это преобразование формирует способ формализации и описания знаний частично определенных предметных областей средствами полностью определенной логики.

10.7. Дедуктивный вывод на модели частично определенных конечных автоматов

Рассмотрим реальный пример применения разработанной частичной логики для решения задач моделирования и проектирования дискретных систем. Продемонстрируем, каким образом разработанный аппарат позволяет формализовать знания о частично определенной предметной области, и каким образом можно формулировать и решать задачи методом логического вывода. Приведенный ниже пример показывает более компактную формализацию знаний и сокращение числа шагов в логическом выводе.

Не полностью определенный конечный автомат. Модель абстрактного конечного автомата широко используется для модели-

рования и синтеза цифровых последовательных схем. Конечный автомат определяется набором из пяти элементов:

$$A = (X, Y, S, \varphi, \delta), \quad (10.54)$$

где X – множество входных символов; Y – множество выходных символов; S – множество внутренних состояний; $\varphi: S \times X \rightarrow S$ – функция переходов; $\delta: S \times X \rightarrow Y$ – функция выходов. Автомат A является не полностью определенным, если функция переходов φ и/или функция выходов δ является не полностью определенной. Такие функции могут принимать значение dc наряду с элементами множеств S и Y . Пример не полностью определенного конечного автомата дан в табл.10.7, где пара s/y означает следующее состояние s и выходной символ y .

Формализация знаний о конечном автомате. Значение *don't care* (dc), использованное в определении конечного автомата, является скорее значением предметных переменных *next_state* и *output_symbol*, чем значением логических переменных. Модель конечного автомата можно определить клозами, построенными без использования теоретических результатов, полученных в данной книге. Однако формализация описания автомата средствами частичной логики требует меньшего числа клозов.

Входные символы x_1, x_2, x_3, x_4 , выходные символы y_1, y_2, y_3, y_4 и внутренние состояния s_0, s_1, s_2, s_3, s_4 представляются функциональными символами степени 0, т.е. индивидуальными константами. Последовательности входных символов, выходных символов и внутренних состояний представляются списками. Чтобы построить клозы, мы вводим ряд предикатов. Важнейшими предикатами являются $next_state(s,x,sn)$, $state_specified(s,x,sn)$, $state_dontcare(s,x,sn)$,

Таблица 10.7

Не полностью определенный конечный абстрактный автомат

| Состояние | Входной символ | | | |
|-----------|----------------|-------------|-------------|-------------|
| | x_1 | x_2 | x_3 | x_4 |
| s_0 | dc / y_1 | s_0 / y_3 | s_4 / y_2 | s_1 / y_2 |
| s_1 | s_2 / y_3 | s_0 / dc | s_3 / dc | dc / y_1 |
| s_2 | dc / y_2 | dc / y_2 | s_1 / dc | s_2 / y_4 |
| s_3 | s_1 / dc | s_4 / dc | dc / y_4 | dc / y_3 |
| s_4 | dc / y_4 | s_3 / y_1 | s_2 / dc | s_3 / dc |

$output(s,x,y)$, $output_specified(s,x,y)$, и $output_dontcare(s,x,y)$, где s – текущее состояние автомата; sn – следующее состояние автомата; x – входной символ; y – выходной символ. Используя эти предикаты, мы можем записать трехзначные клозы следующего вида:

$$(next_state(s,x,sn) | 1) \text{ if } (state_specified(s,x,sn) \text{ or } not\ state_dontcare(s,x,sn)).$$

Согласно (10.52) и (10.53) кюз расщепляется на следующие два двухзначные клоза:

$$next_state(s,x,sn) \text{ if } state_dontcare(s,x,sn). \\ next_state(s,x,sn) \text{ if } state_specified(s,x,sn).$$

Следуя такому подходу, мы получаем Prolog–программу, показанную на рис.10.1.

Решение задач на конечном автомате посредством логического вывода. Знания о частично определенном автомате, формализованные и представленные на рис.10.1, позволяют решать различные задачи, включая следующие:

- какой является выходная последовательность для заданной входной последовательности?
- какой является входная последовательность для заданной выходной последовательности?
- какой является последовательность внутренних состояний автомата для заданных входной и выходной последовательностей?
- имеет ли автомат недостижимые состояния?
- и другие.

Например, цель

$$mapping(s0,X1,[y3,y1,y2,y3,y2,y4]) \text{ and } write("X1=",X1),$$

определенная на рис.10.1 описывает задачу, состоящую в поиске последовательности $X1$ / входных символов для заданной последовательности выходных символов $[y3,y1,y2,y3,y2,y4]$.

Сравнивая предложенные модели представления знаний и логического вывода, построенные на базе методов частичной логики, с известными моделями и методами, приходим к выводу, что предложенные модели ускоряют процесс решения задач и повышают

| | |
|---|---|
| <p>Domains</p> <p><i>state</i>=s0;s1;s2;s3;s4 <i>input_symbol</i>=x1;x2;x3;x4 <i>output_symbol</i>=y1;y2;y3;y4 <i>sequence_of_states</i>=state* <i>sequence_of_inputs</i>=input_symbol* <i>sequence_of_outputs</i>=output_symbol*</p> <p>predicates</p> <p><i>mapping</i>(state,sequence_of_inputs, sequence_of_outputs) <i>next_state</i>(state,input_symbol,state) <i>state_specified</i>(state,input_symbol,state) <i>state_dontcare</i>(state,input_symbol,state) <i>sx_y_dc</i>(state,input_symbol) <i>each_of_y</i>(output_symbol) <i>output</i>(state,input_symbol,output_symbol) <i>output_specified</i>(state,input_symbol, output_symbol) <i>output_dontcare</i>(state,input_symbol, output_symbol) <i>sx_s_dc</i>(state,input_symbol) <i>each_of_s</i>(state)</p> <p>goal</p> <p><i>mapping</i>(s0,X1,[y3,y1,y2,y3,y2,y4]) and write("X1=",X1).</p> <p>clauses</p> <p><i>mapping</i>(_,[],[]). <i>mapping</i>(S,[X X1],[Y Y1]) if <i>output</i>(S,X,Y) and <i>next_state</i>(S,X,Sn) and <i>mapping</i>(Sn,X1,Y1). <i>Output</i>(S,X,Y) if <i>output_specified</i>(S,X,Y) or <i>output_dontcare</i>(S,X,Y). <i>output_specified</i>(s0,x1,y1). <i>output_specified</i>(s0,x2,y3). <i>output_specified</i>(s0,x3,y2). <i>output_specified</i>(s0,x4,y2). <i>output_specified</i>(s1,x1,y3). <i>output_specified</i>(s1,x4,y1). <i>output_specified</i>(s2,x1,y2). <i>output_specified</i>(s2,x2,y2). <i>output_specified</i>(s2,x4,y4). <i>output_specified</i>(s3,x3,y4). <i>output_specified</i>(s3,x4,y3).</p> | <p><i>output_specified</i>(s4,x1,y4). <i>output_specified</i>(s4,x2,y1). <i>output_dontcare</i>(S,X,Y) if <i>sx_y_dc</i>(S,X) and <i>each_of_y</i>(Y). <i>sx_y_dc</i>(s1,x2). <i>sx_y_dc</i>(s1,x3). <i>sx_y_dc</i>(s2,x3). <i>sx_y_dc</i>(s3,x1). <i>sx_y_dc</i>(s3,x2). <i>sx_y_dc</i>(s4,x3). <i>sx_y_dc</i>(s4,x4). <i>each_of_y</i>(y1). <i>each_of_y</i>(y2). <i>each_of_y</i>(y3). <i>each_of_y</i>(y4). <i>next_state</i>(S,X,Sn) if <i>state_specified</i>(S,X,Sn) or <i>state_dontcare</i>(S,X,Sn). <i>state_specified</i>(s0,x2,s0). <i>state_specified</i>(s0,x3,s4). <i>state_specified</i>(s0,x4,s1). <i>state_specified</i>(s1,x1,s2). <i>state_specified</i>(s1,x2,s0). <i>state_specified</i>(s1,x3,s3). <i>state_specified</i>(s2,x3,s1). <i>state_specified</i>(s2,x4,s2). <i>state_specified</i>(s3,x1,s1). <i>state_specified</i>(s3,x2,s4). <i>state_specified</i>(s4,x2,s3). <i>state_specified</i>(s4,x3,s2). <i>state_specified</i>(s4,x4,s3). <i>state_dontcare</i>(S,X,Sn) if <i>sx_s_dc</i>(S,X) and <i>each_of_s</i>(Sn). <i>sx_s_dc</i>(s0,x1). <i>sx_s_dc</i>(s1,x4). <i>sx_s_dc</i>(s2,x1). <i>sx_s_dc</i>(s2,x2). <i>sx_s_dc</i>(s3,x3). <i>sx_s_dc</i>(s3,x4). <i>sx_s_dc</i>(s4,x1). <i>each_of_s</i>(s0). <i>each_of_s</i>(s1). <i>each_of_s</i>(s2). <i>each_of_s</i>(s3). <i>each_of_s</i>(s4).</p> |
|---|---|

Рисунок 10.1 – Описание частично определенного конечного абстрактного автомата на языке Prolog

выразительность и эффективность представления знаний в не полностью определенных прикладных областях. Так, Prolog-программа, приведенная на рис.10.1, содержит 55 клозов. Без использования предложенных моделей необходимо 90 клозов для описания данного конечного автомата.

10.8. Историческая справка

Исследования в области методов решения интеллектуальных задач на основе формализации знаний предметных областей отражены в многочисленных публикациях, включая журнальные статьи, труды конференций, монографии. Разработан ряд систем, базирующихся на знаниях [28]. Логические системы являются базисом для построения машин логического вывода [28, 48]. Теория нечетких множеств позволяет формализовать неопределенности, присутствующие в естественно языке и рассуждениях человека [85]. Формализация неопределенностей исследуемого в настоящей книге типа с целью построения систем, основанных на знаниях и решающих задачи путем логического вывода, дана в работах [58, 60, 63, 100]. В этих работах частичная трехзначная логика использована для формализации знаний из частично определенных предметных областей, введены понятия частичных кванторов, частичных клозов, частичной резольвенты, обобщен принцип резолюций Робинсона применительно к частично определенному логическому выводу.

Глава 11. ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ АЛГОРИТМЫ

11.1. Понятие частично определенного алгоритма

Используемое нами понятие *частично определенного логического алгоритма* возникло из необходимости алгоритмического описания не полностью специфицированных логических вычислений. К ним относятся все вычисления, реализуемые цифровыми (дискретными) системами в условиях неопределенности. Обычно языки программирования, наряду с логическими операциями, содержат арифметические операции, операции-отношения и др. Однако все эти операции реализуются процессором на логическом уровне. Сказанное в полной мере относится и к различного рода спецпроцессорам.

Следует сразу подчеркнуть одно важное отличие модели частично определенного логического алгоритма от известных алгоритмических моделей, например, схем алгоритмов и программ. В логическом алгоритме логическими средствами описывается не только управляющая часть алгоритма, но и его операционно-преобразующая часть. Благодаря этому частично определенный логический алгоритм может быть преобразован к системе частично определенных логических функций. В этой ситуации проблема эквивалентных преобразований алгоритмов получает новое более широкое звучание. Она затрагивает все части алгоритма, при этом преобразования становятся более глубокими и практически значимыми.

Для представления алгоритмов используются машины Тьюринга, рекурсивные функции. Остановимся на операторной форме представления алгоритма. Благодаря теореме о структуризации программ это не сужает изобразительные возможности такого класса алгоритмов. Другими словами, любой алгоритм может быть описан в рассматриваемой нами форме.

Память алгоритма опишем векторной булевой переменной $X=(x_1, \dots, x_n)$, где n – конечное целое положительное число. В абстрактной трактовке алгоритма n может быть бесконечным. Множество переменных, входящих в X , обозначим X^* . На практике все множество X^* переменных разбивается на подмножества, образуя-

щие слова, над которыми выполняются операции. Слова могут быть интерпретируемыми или не интерпретируемыми. С целью исследования общих свойств частично определенных алгоритмов всю память будем представлять одной векторной переменной, выделяя при необходимости ее отдельные биты. Кроме того, мы будем выделять из вектора X части, рассматриваемые как высеченные векторы Y, Z, \dots

При более глубоком понимании частично определенных алгоритмов, мы могли бы описать память векторной переменной $W=(W_1, \dots, W_n)$, в которой компонент W_i принимает три значения 0, 1, *dc*. Переменную W можно закодировать парой булевых векторных переменных, из которой первая описывает состояние памяти, другая – определенность отдельных бит памяти. Все это влечет необходимость наделения алгоритма способностью выполнять частичные операции над памятью. Такую трактовку частично определенного алгоритма мы оставляем за пределами данной книги, хотя представленные в книге исследования создают базис для построения подобного рода алгоритмов.

Защитным выражением называется конструкция

$$\{d(x) \mid R\}, \quad (11.1)$$

где $d(x)$ – скалярная булева функция, характеризующая область определенности вычислений; R – оператор, выполняющий вычисления над памятью x . Оператор R может быть как простым, так и составным. Если вместо $d(x)$ стоит константа 1, то вычисления полностью определены на всей области. В этом случае характеристическая функция может быть опущена вовсе. Если память x находится в состоянии a , то при $d(a)=1$ вычисления и их результаты определены, оператор R запускается на выполнение. При $d(a)=0$ вычисления не определены, оператор не запускается.

Интерпретация области неопределенности значений переменных, которая описывается функцией $\neg d(x)$, может выполняться по-разному. С одной стороны, это может быть область, на которой происходит заикливание алгоритма. С другой стороны, это может быть область, из которой наборы значений никогда не появляются в качестве исходных данных алгоритма. С третьей стороны, результаты выполнения алгоритма на наборах значений из этой области мо-

гут не интересовать пользователей алгоритма. В любом случае, поведение алгоритма на этой области исходных данных является безразличным. Это может использоваться для оптимизации самого алгоритма и его программных или аппаратных реализаций. Выполнение алгоритма на области определенности приводит к изменению значений некоторых или всех переменных в векторе x .

Далее с каждым оператором алгоритма будем ассоциировать векторную булеву функцию $v(x)$, область определенности которой описывается характеристической скалярной булевой функцией $d(x)$.

Частичным векторным оператором присваивания называется конструкция

$$x := e(x) \mid d(x), \quad (11.2)$$

где $e(x)$ – векторная булева функция, изменяющая значения переменных из вектора x ; $d(x)$ – характеристическая функция области определенности; “:=” – символ оператора присваивания. Левая и правая части оператора (11.2) записываются в векторном покомпонентном виде с использованием агрегатов. Если в вектор входит только один компонент, оператор присваивания становится скалярным. На состоянии памяти $x=a$ семантика оператора (11.2) определяется следующим образом. Если $d(a)=1$, то вычисляется значение векторного выражения $e(x)$, это значение обновляет состояние памяти x . Если $d(a)=0$, то выполнение оператора присваивания не определено, память остается в прежнем состоянии, хотя в условиях неопределенности это не обязательно. Если функцией $d(x)$ является константа 1, то оператор присваивания является полностью определенным, а функция $d(x)$ может вовсе опускаться.

Частичным условным оператором называется конструкция

$$(c(x) \mid d(x)) \rightarrow R_1, R_0, \quad (11.3)$$

где $c(x)$ – тестовая булева функция; $d(x)$ – характеристическая функция области определенности; R_1, R_0 – частично определенные операторы, выполняющие вычисления над памятью x ; “ \rightarrow ” – символ условного оператора. На состоянии памяти $x=a$ семантика оператора (11.3) определяется следующим образом. Если $c(a)=1$ и $d(a)=1$, то для выполнения выбирается оператор R_1 , называемый *then*-частью. Если $c(a)=0$ и $d(a)=1$, то для выполнения выбирается

оператор R_0 , называемый *else*-частью. Если $d(a)=0$, то выполнение условного оператора не определено, а операторы R_1 и R_0 не выполняются. Текущее состояние памяти остается прежним.

Составным последовательным оператором называется конструкция

$$R_i ; R_j, \quad (11.4)$$

где R_i, R_j – частично определенные операторы, выполняющие вычисления и преобразующие состояние памяти x ; “;” – символ последовательной операции. Семантика оператора (11.4) состоит в том, что сначала выполняется оператор R_i , после его завершения выполняется оператор R_j .

Составным параллельным оператором называется конструкция

$$R_i \# R_j, \quad (11.5)$$

где R_i, R_j – частично определенные операторы, выполняющие преобразование состояния памяти x ; “#” – символ параллельного оператора. Семантика оператора (11.5) состоит в том, что операторы R_i и R_j выполняется одновременно, синхронно и параллельно. Оператор # идеально применим тогда, когда R_i и R_j работают с различными областями вектора x . При синхронной работе они могут использовать в качестве входных операндов одни и те же компоненты вектора x , однако результаты должны помещаться в разные компоненты. Применение оператора ограничивается наличием информационных зависимостей между R_i и R_j . Если операторы простые, то их необходимо выполнять последовательно. Если информационные зависимости имеют место между частями составных операторов, то необходим механизм синхронизация работы этих частей.

Частичным циклическим оператором называется конструкция

$$\text{while}(c(x) \mid d(x)) R, \quad (11.6)$$

где $c(x)$ – тестовая булева функция, описывающая условие прохождения цикла; $d(x)$ – характеристическая функция области определенности; R – частично определенный оператор, представляющий тело цикла и преобразующий состояние памяти x . При $x=a$ семантика оператора (11.6) определяется следующим образом. Если

$c(a)=1$ и $d(a)=1$, то начинается следующая итерация цикла, при этом на выполнение запускается оператор R , изменяющий состояние памяти x . Если $c(a)=0$ и $d(a)=1$, то цикл завершает выполнение, сохраняя состояние памяти. Если $d(a)=0$, то выполнение циклического оператора, как и его результат, не определено.

Важнейшим частным случаем цикла (11.6) является цикл с оператором присваивания внутри:

$$\text{while}(c(x) | u(x)) \quad x:=e(x) | b(x). \quad (11.7)$$

В схеме итерации и теле цикла используются две различные характеристические функции области определенности: $u(x)$ и $b(x)$.

Мы можем адаптировать к частично определенным областям и другие типы циклов, например, *for*, *do-until*, однако они могут быть функционально эквивалентным образом представлены циклом *while*.

Частично определенным алгоритмом называется частично определенный оператор. Фактически это есть рекурсивное определение алгоритма. Степень вложенности операторов в этом рекурсивном определении ничем не ограничена.

Функции, входящие в определения (11.1)-(11.7) описываются посредством выражений, которые строятся из констант, переменных, одноместных и двухместных операций, многоместных операций, операции суперпозиции. Многоместные векторные операции позволяют представлять и реализовывать арифметические операции, операции-отношения и др. Мы используем механизм подалгоритмов для построения иерархических представлений частично определенных алгоритмов, где один алгоритм может осуществлять вызов другого алгоритма для выполнения сложной операции. Алгоритмы могут вызываться рекурсивно.

11.2. Эквивалентные преобразования частично определенных алгоритмов

Известно, что даже для полностью определенных областей существует много алгоритмов, решающих одну и ту же задачу. Если область является частично определенной, число альтернативных функционально эквивалентных алгоритмов увеличивается. Важ-

нейшей практически востребованной задачей является поиск алгоритма, обладающего одним или несколькими улучшенными параметрами. Практический интерес представляет также задача улучшения параметров реализаций алгоритма. Для того, чтобы решать такие задачи, необходимо разработать средства пошагового эквивалентного преобразования частично определенных алгоритмов.

Частично определенные алгоритмы A и B , которым соответствуют векторные функции $v_A(x)$ и $v_B(x)$ на областях определенности $d_A(x)$ и $d_B(x)$, называются *эквивалентными*, если выполняются равенства $d_A(x) = d_B(x)$ и $v_A(x) \wedge d_A(x) = v_B(x) \wedge d_B(x)$. Это определение выдвигает, во-первых, требование равенства областей определенности алгоритмов A и B , специфицируемых допустимыми наборами начальных значений вектора x . Во-вторых, выполнение каждого из алгоритмов A и B на каждом из допустимых начальных состояний памяти должно дать идентичные конечные состояния памяти. В области неопределенности поведение каждого из алгоритмов может быть произвольным, и это не нарушает понятие эквивалентности.

Рассматриваемые нами преобразования базируются на использовании *правил преобразования*. Правила применяются к операторам и суперпозициям операторов, при этом они могут выполнить переход от одного типа операторов к другому типу. Базовый набор правил преобразует операторы в рамках одного типа.

Функционально эквивалентным правилом P_i преобразования называется пара частично определенных подалгоритмов α и β , для которых $d_\alpha(x) = d_\beta(x)$ и $v_\alpha(x) \wedge d_\alpha(x) = v_\beta(x) \wedge d_\beta(x)$. Правило P преобразования будем записывать в виде эквивалентности $\alpha \leftrightarrow \beta$. Далее функционально эквивалентное правило преобразования будем также называть просто правилом преобразования.

Формальной системой эквивалентных преобразований частично определенных алгоритмов назовем множество \mathcal{P} правил преобразования. К множеству автоматически добавляются все законы булевой алгебры и смешанной алгебры, позволяющие выполнять преобразования всех алгебраических выражений, входящих в операторы исходного алгоритма или появляющихся в процессе преобразования алгоритма.

Алгоритм B *непосредственно выводим* из алгоритма A , если существует правило $\alpha \leftrightarrow \beta$ такое, что в алгоритме A найдется фраг-

мент, унифицируемый с подалгоритмом α , замена которого на подалгоритм β дает алгоритм B . Алгоритм B выводим из алгоритма A , если существует последовательность правил P_1, \dots, P_k и последовательность алгоритмов A_0, \dots, A_k таких, что $A_0=A$, $A_k=B$ и алгоритм A_i непосредственно выводим из алгоритма A_{i-1} посредством правила P_i при $i=1, \dots, k$. Очевидно, поскольку алгоритмы A_{i-1} и A_i функционально эквивалентны, то алгоритмы A и B также функционально эквивалентны. Значит, понятие выводимости является инструментом в получении разнообразных эквивалентных алгоритмов.

Множество алгоритмов, выводимых из алгоритма A , является подмножеством множества алгоритмов, функционально эквивалентных алгоритму A . Если бы любой эквивалентный алгоритм был выводимым, формальная система преобразования обладала бы свойством *полноты*. В противном случае система преобразований таким свойством не обладает. Мы не требуем наличия свойства полноты у нашей системы преобразований. Для нас важнее всего полезность правил преобразований в плане улучшения ключевых параметров алгоритма и параметров программно-аппаратных реализаций алгоритма. Каждое предлагаемое ниже правило строго обосновывается путем доказательства эквивалентности левой части и правой части, из которых правило состоит.

Основная суть преобразований состоит в добавлении в алгоритм или исключении из алгоритма управляющих конструкций, совмещении частей алгоритма с целью их совместной минимизации, изменении входящих в алгоритм функций при сохранении функции всего алгоритма. Если мы хотим иметь гибкую систему трансформаций алгоритмов, то в состав правил должны входить те, которые ведут к последовательному уменьшению времени выполнения алгоритма, а также те, которые ведут к уменьшению сложности представления и реализации алгоритма. Планирование применения правил ведется в этом случае с учетом решаемой оптимизационной задачи.

Начнем с правил преобразования составного последовательного и составного параллельного операторов.

Правило 1. Составной последовательный оператор

$$x := e(x) \mid d(x) ; x := s(x) \mid b(x) \quad (11.8)$$

функционально эквивалентен оператору присваивания

$$x := s(e(x)) \mid d(x) \wedge b(e(x)) \quad (11.9)$$

Действительно, согласно (11.8) конечное состояние памяти определяется путем последовательного применения векторных функций e и s . Согласно (11.9) суперпозиция функций $s(e(x))$ дает такое же последовательное применение. Согласно (11.8), область определенности всего составного оператора есть пересечение областей определенности двух последовательных операторов присваивания, а это и есть $d(x) \wedge b(e(x))$.

Правило 2. Составной параллельный оператор

$$x' := e(x) \mid d(x) \# x'' := s(x) \mid b(x) \quad (11.10)$$

функционально эквивалентен оператору присваивания

$$x := r(x) \mid d(x) \wedge b(x), \quad (11.11)$$

где $x = (x', x'')$, $r(x) = (e(x), s(x))$. При выполнении операторов в (11.10) результирующее значение вектора x получается за счет параллельного синхронного изменения значений его составляющих x' и x'' . Значения определяются векторными функциями $e(x)$ и $s(x)$. Точно также изменяется значение вектора x оператором (11.11). Параллельный оператор (11.10) определен, если одновременно определены две его части, т.е. если $d(x) \wedge b(x) = 1$. На такой же области определен и оператор (11.11).

Прямое применение правила 2 агрегирует параллельные операторы присваивания, сужая при этом область определенности каждого из них. Из-за сужения областей определенности, применение оператора должно быть обоснованным. Обратное применение правила 2 позволяет расщепить векторный оператор присваивания (11.2) на n параллельно исполняемых независимых частей:

$$x_1 := e_1(x) \mid d(x) \# \dots \# x_n := e_n(x) \mid d(x) \quad (11.12)$$

Правило 3. Другой вид расщепления векторного оператора присваивания (11.2) связан с декомпозицией векторной функции $e(x)$ в виде $r(x, s(x))$. В этом случае оператор (11.2) эквивалентен составному последовательному оператору

$$y:=s(x) \mid d(x) ; x:=r(x, y) \mid d(x) \quad (11.13)$$

Расщепление (11.13) использует дополнительную память y . Правило 3 может применяться и в обратном порядке.

Пример 11.1. Дан частично определенный составной параллельный оператор присваивания, находящийся под защитным выражением:

$$\{ x_1 \vee x_2 \mid x_1 := x_1 \wedge (x_2 \vee x_3) \# x_2 := \neg x_2 \# x_3 := x_1 \oplus (x_2 \vee x_3) \}.$$

Применим к нему с целью вынесения опережающих вычислений правило 3, выбрав функцию $s(x) = x_2 \vee x_3$. В результате появляется составной последовательный оператор:

$$\{ x_1 \vee x_2 \mid y := x_2 \vee x_3 \} ; \{ x_1 \vee x_2 \mid x_1 := x_1 \wedge y \# x_2 := \neg x_2 \# x_3 := x_1 \oplus y \}.$$

Таким образом, выполненные преобразования привели к перестройке последовательно-параллельного выполнения исходного алгоритма, основанной на предварительном выделении общей части. \square

11.3. Преобразование частично определенных ветвящихся алгоритмов

Ветвящиеся алгоритмы описываются условными операторами (11.3) и суперпозициями условных операторов. Прежде, чем предложить правила преобразования частично определенных условных операторов, исследуем переход от полностью определенно-го условного оператора к эквивалентной системе булевых функций.

Запишем условный оператор (11.3) в виде:

$$c(x) \rightarrow y:=t(x), z:=e(x), \quad (11.14)$$

где векторные переменные x, y, z представляют входные данные всего условного оператора, выходные данные *then*-части и выводные данные *else*-части соответственно. Множества x^*, y^*, z^* могут пересекаться. Обозначим через $w^* = y^* \cup z^*$ множество выходных переменных условного оператора.

Условному оператору (11.14) соответствует векторный оператор присваивания $w := v(x)$. Исследуем построение векторной функ-

ции $V(x)$ в зависимости от переменных и функций, входящих в условный оператор.

Если выполняется $y^* \cap z^* = \emptyset$, то функция V , отображающая вектор x в вектор $w=(y, z)$, записывается в виде:

$$V(x) = (c(x) \wedge t(x) \vee \neg c(x) \wedge y, c(x) \wedge z \vee \neg c(x) \wedge e(x)), \quad (11.15)$$

Если выполняется $y^* = z^*$, то функция V , отображающая вектор x в вектор y , имеет вид:

$$V(x) = c(x) \wedge t(x) \vee \neg c(x) \wedge e(x), \quad (11.16)$$

Если одновременно выполняется $y^* \cap z^* \neq \emptyset$ и $y^* \neq z^*$, то функция V отображает вектор x в вектор $w=(f, g, h)$, где $f^* = y^* \setminus z^*$, $g^* = y^* \cap z^*$, $h^* = z^* \setminus y^*$, и записывается в виде:

$$V(x) = (c(x) \wedge t_f(x) \vee \neg c(x) \wedge f, \\ c(x) \wedge t_g(x) \vee \neg c(x) \wedge e_g(x) \\ c(x) \wedge t_h(x) \vee \neg c(x) \wedge e_h(x)), \quad (11.17)$$

где $t_f(x)$, $t_g(x)$, $e_g(x)$, $e_h(x)$ – части векторных функций $t(x)$ и $e(x)$, соответствующие векторам f, g, h .

На основе соотношений (11.15)-(11.17) построим правила преобразования частично определенных условных операторов.

Правило 4. Частично определенный укороченный условный оператор

$$(c(x) \mid d(x)) \rightarrow x := t(x) \mid b(x) \quad (11.18)$$

функционально эквивалентен частично определенному оператору присваивания

$$x := c(x) \wedge t(x) \vee \neg c(x) \wedge x \mid d(x) \wedge (b(x) \vee \neg c(x)). \quad (11.19)$$

Если $c(x)=1$, то новое состояние памяти x определяется векторной функцией $t(x)$, в противном случае состояние памяти остается неизменным. Результат выполнения оператора (11.18) определен, если выполняется $d(x)=1$ и $c(x)=0$, либо при $d(x)=1$ и $c(x)=1$ выполняется $b(x)=1$. Записывая это в формульном виде, получаем функцию определенности $d(x) \wedge (c(x) \wedge b(x) \vee \neg c(x) \wedge 1)$, которая с помощью законов булевой алгебры упрощается до $d(x) \wedge (b(x) \vee \neg c(x))$.

Правило 5. Частично определенный условный оператор

$$(c(x) | d(x)) \rightarrow x:=t(x) | b(x), x:=e(x) | u(x) \quad (11.20)$$

функционально эквивалентен частично определенному оператору присваивания

$$x:=c(x)\wedge t(x)\vee\neg c(x)\wedge e(x) | \quad (11.21) \\ d(x)\wedge(c(x)\wedge b(x)\vee\neg c(x)\wedge u(x)).$$

Если $c(x)=1$, то новое состояние памяти определяется векторной функцией $t(x)$, в противном случае состояние памяти определяется векторной функцией $e(x)$. Результат выполнения оператора (11.20) определен, если при $d(x)=1$ и $c(x)=1$ выполняется $b(x)=1$, либо при $d(x)=1$ и $c(x)=0$ выполняется $u(x)=1$. Записывая это в формульном виде, получаем функцию определенности $d(x)\wedge(c(x)\wedge b(x)\vee\neg c(x)\wedge u(x))$, идентичную функции определенности в операторе присваивания (11.21).

Правило 6. Частично определенный условный оператор

$$(c(x) | d(x)) \rightarrow y:=t(x) | b(x), z:=e(x) | u(x) \quad (11.22)$$

функционально эквивалентен следующим параллельным операторам присваивания, помещенным в защитное выражение:

$$\{ d(x)\wedge(c(x)\wedge b(x)\vee\neg c(x)\wedge u(x)) | \\ f:=c(x)\wedge t_f(x)\vee\neg c(x)\wedge f\# \\ g:=c(x)\wedge t_g(x)\vee\neg c(x)\wedge e_g(x)\# \\ h:=c(x)\wedge h\vee\neg c(x)\wedge e_h(x) \}, \quad (11.23)$$

где как и прежде $f^*=y^*\wedge z^*$, $g^*=y^*\wedge z^*$, $h^*=z^*\wedge y^*$; $t_f(x)$, $t_g(x)$, $e_g(x)$, $e_h(x)$ – части векторных функций $t(x)$ и $e(x)$, соответствующие векторам f , g , h .

Из правил 4-6 вытекают следующие следствия (более конкретные правила, описываемые посредством эквивалентности \leftrightarrow), в которых используется одинаковая функция результирующей области определенности $w(x)=d(x)\wedge(c(x)\wedge b(x)\vee\neg c(x)\wedge u(x))$:

$$(c(x)|d(x))\rightarrow x:=t(x)|b(x), x:=t(x)|u(x) \leftrightarrow x:=t(x) | w(x), \quad (11.24)$$

$$(c(x)|d(x))\rightarrow x:=1|b(x), x:=0|u(x) \leftrightarrow x:=c(x) | w(x), \quad (11.25)$$

$$(c(x)|d(x)) \rightarrow x:=0|b(x), x:=1|u(x) \leftrightarrow x:=\neg c(x) | u(x), \quad (11.26)$$

$$(c(x)|d(x)) \rightarrow y:=0|b(x), z:=0|u(x) \leftrightarrow \{ u(x) | y:=\neg c(x) \wedge y \# z:=c(x) \wedge z \}, \quad (11.27)$$

$$(c(x)|d(x)) \rightarrow y:=1|b(x), z:=1|u(x) \leftrightarrow \{ u(x) | y:=c(x) \vee y \# z:=\neg c(x) \vee z \}, \quad (11.28)$$

$$(c(x)|d(x)) \rightarrow y:=1|b(x), z:=0|u(x) \leftrightarrow \{ u(x) | y:=c(x) \vee y \# z:=c(x) \wedge z \}, \quad (11.29)$$

$$(c(x)|d(x)) \rightarrow y:=0|b(x), z:=1|u(x) \leftrightarrow \{ u(x) | y:=\neg c(x) \wedge y \# z:=\neg c(x) \vee z \}, \quad (11.30)$$

$$(c(x)|d(x)) \rightarrow x:=t(x)|b(x), x:=e(x)|u(x) \leftrightarrow (c(x)|d(x)) \rightarrow x:=t(x)|b(x) \# (\neg c(x)|d(x)) \rightarrow x:=e(x)|u(x). \quad (11.31)$$

Правило (11.24) позволяет заменить условный оператор его частью. Правила (11.25), (11.26) заменяют условный оператор оператором присваивания с тестовым выражением. В правилах (11.27)-(11.30) предполагается, что множества y и z не пересекаются, при этом условный оператор заменяется двумя короткими операторами присваивания. Правило (11.31) расщепляет условный оператор на два параллельно выполняемых коротких условных оператора. Правила (11.24)-(11.31) позволяют значительно упростить условный оператор при конкретном виде входящих в него функций.

Пример 11.2. Дан частично определенный условный оператор

$$(x_1 \vee x_2 | 1) \rightarrow x_1 := x_1 \wedge x_2 | x_1, x_2 := \neg x_1 \vee \neg x_2 | x_2.$$

Воспользуемся правилом (11.31) для расщепления этого оператора на две короткие параллельно выполняемые части:

$$(x_1 \vee x_2 | 1) \rightarrow x_1 := x_1 \wedge x_2 | x_1 \# (\neg(x_1 \vee x_2) | 1) \rightarrow x_2 := \neg x_1 \vee \neg x_2 | x_2.$$

Трансформируем каждую часть к оператору присваивания по правилу 4:

$$x_1 := (x_1 \vee x_2) \wedge x_1 \wedge x_2 \vee \neg(x_1 \vee x_2) \wedge x_1 \mid x_1 \vee \neg(x_1 \vee x_2) \#$$

$$x_2 := \neg(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \vee (x_1 \vee x_2) \wedge x_2 \mid x_2 \vee (x_1 \vee x_2).$$

Упрощая каждый из двух операторов присваивания по законам булевой алгебры, получаем:

$$x_1 := x_1 \wedge x_2 \mid x_1 \vee \neg x_2 \quad \# \quad x_2 := \neg x_1 \vee x_2 \mid x_1 \vee x_2.$$

Функции, входящие в правые части операторов являются частично определенными. Выполняя их минимизацию, окончательно записываем:

$$x_1 := x_2 \mid x_1 \vee \neg x_2 \quad \# \quad x_2 := x_2 \mid x_1 \vee x_2.$$

Таким образом, выполненные преобразования дали существенное упрощение описания логических вычислений. \square

11.4. Преобразование частично определенных циклических алгоритмов

Рассмотрим цикл (11.7) с оператором присваивания внутри. Пусть все переменные $x = (x_1, \dots, x_n)$ являются выходными для цикла. Если подавать на вход цикла различные начальные значения переменных x_1, \dots, x_n , то возможно возникновение следующих ситуаций:

- выполнение цикла не определено;
- происходит заикливание;
- на выходе цикла появляются результирующие значения переменных.

Таким образом, всему циклу (11.7) соответствует частично определенная векторная функция, описываемая парой булевых функций $v(x)$ и $d(x)$. Исследуем задачу построения этих функций в зависимости от переменных и функций, являющихся составными частями цикла.

Пусть n является конечным целым числом. Тогда при отсутствии заикливания число итераций цикла (11.7) имеет некоторое максимальное значение. Это значение устанавливается следующим утверждением:

Если для начального набора $a = (a_1, \dots, a_n)$ значений переменной x цикл имеет конечное число проходов l , то это число не превышает значения $l \leq 2^n - 1$.

Докажем это утверждение. Если на входном наборе a цикл имеет конечное число проходов, то существует конечная последовательность состояний памяти $a^0=a, a^1, a^2, \dots, a^{l-1}, a^l$ такая, что $d(a^0)=d(a^1)=\dots=d(a^{l-1})=1$ и $d(a^l)=0$. Покажем, что для этой последовательности выполняются неравенства $a^0 \neq a^1 \neq a^2 \neq \dots \neq a^{l-1} \neq a^l$. Предположим противное, а именно $a^i = a^j$ при $i < j$. Не может иметь место $j=l$, поскольку в этом случае выполнялось бы $d(a^i) \neq d(a^l)$. Из $a^i = a^j$ следует $a^{i+1} = a^{j+1}$, так как a^{i+1} получается из a^i с помощью того же преобразования, что и a^{j+1} из a^j . Равенство $a^{i+k} = a^{j+k}$ должно выполняться тогда для всех $k=1, \dots$. Поскольку $d(a^{j+k})=1$, цикл имел бы бесконечное число проходов на входном наборе a . Это противоречит условию утверждения. Значит, состояния памяти на различных итерациях различны. Поскольку существует 2^n различных значений векторной переменной x , а все значения различны на различных итерациях цикла, и, кроме того, существует хотя бы один набор, на котором происходит выход из цикла, то максимальное число итераций l не может превышать значения $2^n - 1$.

Вид функции $v(x)$ определяет следующее утверждение:

Для векторной булевой функции $v(x)$ цикла выполняется следующее *рекурсивное* соотношение:

$$v(x) = \neg c(x) \wedge x \vee c(x) \wedge v(e(x)). \quad (11.32)$$

Докажем это утверждение. Если на некотором значении вектора x выполняется $c(x)=0$, то осуществляется выход из цикла (11.7), при этом $v(x)=x$. Если на некотором значении вектора x выполняется $c(x)=1$, то осуществляется пересчет значения x с помощью оператора $x:=e(x)$, после чего происходит переход в начало цикла. Цикл продолжает выполняться со значением $e(x)$ переменной x . Поскольку первому проходу соответствует функция $v(x)$, то второму проходу при $c(x)=1$ и $x=e(x)$ соответствует функция $v(e(x))$. Рассмотренные случаи и описывают соотношение (11.32).

Вид функции $d(x)$ определяет следующее утверждение:

Для скалярной характеристической булевой функции $d(x)$ цикла выполняется следующее *рекурсивное* соотношение:

$$d(x) = u(x) \wedge (\neg c(x) \vee b(x) \wedge d(e(x))). \quad (11.33)$$

Докажем это утверждение. Если $c(x)=0$, то результат выполнения цикла определен на области $u(x)$. Если $c(x)=1$, то результат выполнения цикла определен на пересечении областей $u(x)$, $b(x)$ и $d(e(x))$. Последняя область устанавливает определенность результата на последующих итерациях цикла.

Выполним *раскрутку* рекурсивного уравнения (11.32), последовательно подставляя вместо рекурсивного вхождения $v(e(x))$ полное определение этой функции с заменой переменной x на аргумент $e(x)$:

$$\begin{aligned} v(x) &= \neg c(x) \wedge x \vee c(x) \wedge (\neg c(e(x)) \wedge e(x) \vee c(e(x)) \wedge v(e(x))) = \\ &= \neg c(x) \wedge x \vee c(x) \wedge \neg c(e(x)) \wedge e(x) \vee c(x) \wedge c(e(x)) \wedge v(e(x)) = \\ &= \dots = \\ &= \neg c(x) \wedge x \vee c(x) \wedge \neg c(e(x)) \wedge e(x) \vee c(x) \wedge c(e(x)) \wedge \neg c(e(e(x))) \wedge e(e(x)) \vee \dots \end{aligned}$$

Обозначим $c(x)$ через c^0 , $c(e(x))$ через c^1 , $c(e(\dots e(x)\dots))$ через c^i . Обозначим также $e(x)$ через e^1 , $e(e(x))$ через e^2 , $e(\dots e(x)\dots)$ через e^i . Тогда для функции $v(x)$ можно записать:

$$v(x) = \neg c^0 \wedge x \vee c^0 \wedge \neg c^1 \wedge e^1 \vee c^0 \wedge c^1 \wedge \neg c^2 \wedge e^2 \vee \dots \vee c^0 \wedge c^1 \wedge c^2 \wedge \dots \wedge c^{i-1} \wedge \neg c^i \wedge e^i \vee \dots \quad (11.34)$$

Аналогичным образом раскрутим рекурсивное уравнение (11.33). Для этого обозначим $u(x)$ через u^0 , $u(e(x))$ через u^1 , $u(e(\dots e(x)\dots))$ через u^i . Обозначим также $b(x)$ через b^0 , $b(e(x))$ через b^1 , $b(e(\dots e(x)\dots))$ через b^i . Получаем:

$$\begin{aligned} d(x) &= u^0 \wedge (\neg c^0 \vee b^0 \wedge u^1 \wedge (\neg c^1 \vee b^1 \wedge u^2 \wedge (\neg c^2 \vee \dots))) = \\ &= u^0 \wedge \neg c^0 \vee u^0 \wedge b^0 \wedge u^1 \wedge \neg c^1 \vee u^0 \wedge b^0 \wedge c^1 \wedge b^1 \wedge u^2 \wedge \neg c^2 \vee \dots \vee \\ &= u^0 \wedge b^0 \wedge u^1 \wedge b^1 \wedge \dots \wedge u^{i-1} \wedge b^{i-1} \wedge u^i \wedge \neg c^i \wedge \dots \end{aligned} \quad (11.35)$$

Выражение (11.34) позволяет получить следующее вложенное описание функции $v(x)$:

$$v(x) = v^k(x) = \neg c^0 \wedge x \vee c^0 \wedge (\neg c^1 \wedge e^1 \vee c^1 \wedge (\neg c^2 \wedge e^2 \vee \dots \vee c^{k-1} \wedge (\neg c^k \wedge e^k) \dots)), \quad (11.36)$$

где k – наименьшее натуральное число такое, что $v^{k+1}(x) \wedge d^{k+1}(x) = v^k(x) \wedge d^k(x)$; k удовлетворяет ограничению $k \leq 2^n - 1$.

Выражение (11.35) позволяет получить следующее вложенное описание функции $d(x)$:

$$d(x) = d^k(x) = u^0 \wedge (\neg c^0 \vee u^1 \wedge b^0 \wedge (\neg c^1 \vee \dots \vee u^k \wedge b^{k-1} \wedge (\neg c^k) \dots)) \quad (11.37)$$

где k – наименьшее натуральное число такое, что $d^{k+1}(x) = d^k(x)$; k удовлетворяет ограничению $k \leq 2^n - 1$.

Изучив вид функций $v(x)$ и $d(x)$, рассмотрим эквивалентные преобразования:

- цикла в цикл;
- цикла в условный оператор;
- цикла в оператор присваивания.

Это необходимо для управления такими параметрами алгоритма как быстрдействие и сложность.

Правило 7. Частично определенный оператор цикла (11.7) функционально эквивалентен следующему циклу с меньшим числом итераций:

$$\text{while}(c(x) \mid u(x)) \quad x := c(e(x)) \wedge e(e(x)) \vee \neg c(e(x)) \wedge e(x) \mid \quad (11.38) \\ b(x) \wedge u(e(x)) \wedge \neg c(e(x)) \vee b(e(x)).$$

Доказательство правила 7 проводится путем *раскрутки цикла* (11.7). Для этого одна итерация цикла представляется условным оператором, который вставляется в тело цикла после оператора присваивания с использованием оператора последовательного выполнения. Последующее преобразование полученного составного тела цикла дает выражение (11.38). Вставка условного оператора внутрь цикла приводит к тому, что одна итерация цикла (11.38) заменяет две итерации цикла (11.7). Это может дать двойное сокращение числа итераций и как следствие сокращение общего времени выполнения цикла. Платой является усложнение в общем случае структуры тела цикла. Правило 7 может быть эффективно применено в прямом направлении. В обратном направлении применение правила затруднено.

Пример 11.3. Дан цикл

$$\text{while}(x_1 \wedge x_2) \{ x_1 := x_4 \# x_2 := x_1 \oplus x_4 \# x_3 := x_2 \# x_4 := x_2 \oplus x_3 \}.$$

Применим к циклу правило 7, вычислив следующие промежуточные выражения:

$$\begin{aligned} c(e(x)) &= \neg x_1 \wedge x_4, \\ \neg c(e(x)) &= x_1 \vee \neg x_4, \\ e_1(e(x)) &= x_2 \oplus x_3, \\ e_2(e(x)) &= x_4 \oplus (x_2 \oplus x_3), \\ e_3(e(x)) &= x_1 \oplus x_4, \\ e_4(e(x)) &= (x_1 \oplus x_4) \oplus x_2, \end{aligned}$$

Собирая воедино части выражения (11.38), получаем:

$$\begin{aligned} \text{while}(x_1 \wedge x_2) \{ & x_1 := (x_1 \vee (x_2 \oplus x_3)) \wedge x_4 \# \\ & x_2 := x_1 \wedge \neg x_4 \vee \neg x_1 \wedge x_2 \wedge x_3 \wedge x_4 \vee \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \# \\ & x_3 := \neg x_1 \wedge x_4 \vee (x_1 \vee \neg x_4) \wedge x_2 \# \\ & x_4 := \neg x_1 \wedge \neg x_2 \wedge x_4 \vee (x_1 \vee \neg x_4) \wedge (x_2 \oplus x_3) \} \end{aligned}$$

Число итераций в полученном эквивалентном цикле до двух раз меньше числа итераций в исходном цикле. Платой явилось заметное усложнение тела цикла. \square

Условия и порядок перехода от частично определенного цикла к векторному оператору присваивания устанавливает следующее правило.

Правило 8. Частично определенный оператор цикла (11.7) функционально эквивалентен следующему векторному оператору присваивания

$$x := c(x) \wedge e(x) \vee \neg c(x) \wedge x \mid u(x) \wedge \neg c(x) \vee b(x) \wedge u(e(x)), \quad (11.39)$$

если выполняется условие

$$c(e(x)) = 0. \quad (11.40)$$

Докажем правило 8. В выражении (11.36) терму $c(e(x))$ соответствует обозначение c^1 . Выполняя условие (11.40), подставляем в (11.36) вместо c^1 значение 0 и получаем

$$u(x) = \neg c^0 \wedge x \vee c^0 \wedge (\neg 0 \wedge e^1 \vee 0 \wedge (\dots)) = \neg c^0 \wedge x \vee c^0 \wedge e^1,$$

что совпадает с соответствующей функцией, входящей в (11.39). Подстановка $c^1=0$ в выражение (11.37) дает

$$d(x) = U^0 \wedge (\neg C^0 \vee U^1 \wedge B^0 \wedge (\neg 0 \vee \dots)) = U^0 \wedge (\neg C^0 \vee U^1 \wedge B^0),$$

что совпадает с функцией области определенности, входящей в (11.39). Применение правила 8 в прямом направлении завершает раскрутку цикла, сводя цикл к оператору присваивания. Применение правила в обратном направлении не является простым, хотя в принципе возможно.

Пример 11.4. Дан цикл

$$\text{while}(x_1 \wedge x_2 \wedge x_3) \{ x_1 := x_2 \vee x_3 \# x_2 := x_4 \wedge \neg x_3 \# x_3 := \neg x_2 \wedge \neg x_3 \# x_4 := x_2 \}.$$

Проверим выполнение условия (11.40) для суперпозиции функций $c(x)$ и $e(x)$:

$$c(e(x)) = (x_2 \vee x_3) \wedge (x_4 \wedge \neg x_3) \wedge (\neg x_2 \wedge \neg x_3) = 0.$$

Условие выполняется, поэтому исходный цикл сводится к четырем параллельно выполняемым операторам присваивания:

$$\begin{aligned} x_1 &:= x_1 \# \\ x_2 &:= (\neg x_1 \vee \neg x_3) \wedge x_2 \# \\ x_3 &:= (\neg x_1 \vee \neg x_2) \wedge x_3 \# \\ x_4 &:= x_1 \wedge x_2 \wedge x_3 \vee \neg x_1 \wedge x_4 \vee \neg x_2 \wedge x_4 \vee \neg x_3 \wedge x_4. \end{aligned}$$

Очевидно, что полученный алгоритм проще исходного. \square

Правило 9. Частично определенный оператор цикла (11.7) функционально эквивалентен следующему частично определенному векторному оператору присваивания

$$\begin{aligned} x := & c(x) \wedge \neg c(e(x)) \wedge e(x) \vee \neg c(x) \wedge x \mid \\ & u(x) \wedge (\neg c(x) \vee \neg c(e(x)) \wedge u(e(x)) \wedge b(x)), \end{aligned} \quad (11.41)$$

если выполняется условие

$$c(x) \wedge c(e(x)) \wedge \neg c(e(e(x))) = 0. \quad (11.42)$$

Доказательство правила 9 аналогично доказательству правила 8. Это правило устанавливает еще одно условие завершения раскрутки цикла, сводя цикл к оператору присваивания.

Переход от оператора цикла к условному оператору с вложенным циклом описывает следующее правило.

Правило 10. Частично определенный оператор цикла (11.7) функционально эквивалентен следующему укороченному условному оператору с последовательно соединенным циклом и оператором присваивания внутри:

$$(c(x) \mid u(x)) \rightarrow \{ \text{while}(c(e(x)) \mid u(e(x)) \wedge b(x)) \ x := e(x) ; \quad (11.43) \\ x := e(x) \}$$

Доказательство правила 10 основано на разложении цикла с использованием условного оператора. Правило осуществляет пошаговый переход от цикла к системе вложенных условных операторов.

Переход от оператора цикла к совокупности последовательно соединенных операторов присваивания описывает следующее правило.

Правило 11. Частично определенный оператор цикла (11.7) функционально эквивалентен следующему составному последовательному оператору:

$$x := c(x) \wedge e(x) \vee \neg c(x) \wedge x \mid u(x) \wedge (\neg c(x) \vee b(x)) ; \quad (11.44) \\ \text{while}(c(x) \mid u(x)) \ x := e(x) \mid b(x)$$

Доказательство правила 11 проводится разложением цикла на цикл и условный оператор, который затем преобразуется к оператору присваивания. Из правила 11 вытекает очень важное следствие. Обозначим один экземпляр оператора присваивания через A_i . Тогда многократное применение разложения (11.44) дает последовательность операторов присваивания

$$A_1; A_2; \dots; A_k. \quad (11.45)$$

где k определяется из (11.36). Разложение (11.45) дает возможность трансформировать цикл к *конвейерной* реализации.

11.5. Сведение частично определенного алгоритма к системе частично определенных логических функций

Построить функции $v_A(x)$ и $d_A(x)$, соответствующие частично определенному алгоритму A , можно путем последовательной генерации различных начальных состояний памяти x и выполнения алгоритма A на каждом из состояний. В результате получаем таблицу

ное представление функций. Трудности такого подхода заключаются в:

- возможном заиклиивании алгоритма и невозможности получения в этом случае его выходной реакции;
- высокой трудоемкости вычислений из-за огромного числа входных состояний;
- большом (или огромном) размере таблицы, описывающей функции.

В связи с этим на первый план выдвигается аналитический метод построения функций $v_A(x)$ и $d_A(x)$. Он заключается в преобразовании алгоритма к одиночному частично определенному векторному оператору присваивания. Решающее значение в аналитическом методе имеют предложенные выше правила эквивалентного преобразования алгоритмов.

Прежде чем изложить методику построения функций для всего алгоритма, остановимся на методах построения функций для циклического оператора. К непосредственному построению функций приводят выражения (11.36), (11.37), однако эти выражения требуют k шагов преобразований. Если k близко к значению $2^n - 1$, то с преобразованиями связан большой объем работы.

Более быстрым является метод *раскрутки*, основанный на использовании правил 7 и 8, а также соотношений (11.38)-(11.40). Число требуемых шагов не превышает значения $\lceil \log_2 k \rceil$. Метод заключается в последовательном построении функций $e^0, e^1, e^2, \dots, e^i, \dots$ и функций $b^0, b^1, b^2, \dots, b^i, \dots$. Согласно (11.38) функция e^{i+1} выражается через функцию e^i посредством формулы $e^{i+1} = c(e^i(x)) \wedge e^i(e^i(x)) \vee \neg c(e^i(x)) \wedge e^i(x)$. Функция b^{i+1} выражается через функцию b^i посредством формулы $b^{i+1} = b^i(x) \wedge u(e^i(x)) \wedge \neg c(e^i(x)) \vee b^i(e^i(x))$. Построение функций ведется до тех пор, пока не выполнится условие (11.40). При выполнении условия процесс завершается построением оператора присваивания по соотношению (11.39).

Метод включает следующие шаги преобразований при некотором произвольном начальном состоянии памяти:

0. Присвоить $e^0 := e(x)$, $b^0 := b(x)$, $i := 0$.
1. Если $b^i = 0$, то функции $v(x)$ и $d(x)$ построены. Функция $v(x)$ произвольная, функция $d(x) := 0$.

2. Если $c(e^i(x))=0$, то функции $v(x)$ и $d(x)$ построены. Функция $v(x):=c(x)\wedge e^i(x)\vee\neg c(x)\wedge x$, функция $d(x):=u(x)\wedge\neg c(x)\vee b^i(x)\wedge u(e^i(x))$.
3. В противном случае осуществляется построение функций $e^{i+1}:=c(e^i(x))\wedge e^i(e^i(x))\vee\neg c(e^i(x))\wedge e^i(x)$ и $b^{i+1}:=b^i(x)\wedge u(e^i(x))\wedge\neg c(e^i(x))\vee b^i(e^i(x))$. Счетчик шагов инкрементируется: $i:=i+1$. Выполняется переход к шагу 1.

Пример 11.5. Дан цикл

$$\text{while}(\neg(x_1\wedge x_2)) \{ x_1:=x_1\oplus x_2 \# x_2:=\neg x_2 \# x_3:=0 \# \\ x_4:=x_3 \# x_5:=x_4 \# x_6:=x_5 \# x_7:=x_6 \}.$$

Три итерации метода раскрутки дают следующий составной параллельный оператор, включающий семь операторов присваивания:

$$\begin{aligned} x_1 &:= 1 \# \\ x_2 &:= 1 \# \\ x_3 &:= x_1\wedge x_2\wedge x_3 \# \\ x_4 &:= x_1\wedge x_2\wedge x_4\vee x_1\wedge\neg x_2\wedge x_3 \# \\ x_5 &:= x_1\wedge x_2\wedge x_5\vee x_1\wedge\neg x_2\wedge x_4\vee\neg x_1\wedge x_2\wedge x_3 \# \\ x_6 &:= x_1\wedge x_2\wedge x_6\vee x_1\wedge\neg x_2\wedge x_5\vee\neg x_1\wedge x_2\wedge x_4\vee\neg x_1\wedge\neg x_2\wedge x_3 \# \\ x_7 &:= x_1\wedge x_2\wedge x_7\vee x_1\wedge\neg x_2\wedge x_6\vee\neg x_1\wedge x_2\wedge x_5\vee\neg x_1\wedge\neg x_2\wedge x_4. \end{aligned}$$

Очевидно, что многошаговое последовательное выполнение исходного циклического алгоритма протекает дольше, чем одношаговое параллельное выполнение построенных операторов присваивания. \square

Вернемся к построению функций $v(x)$ и $d(x)$ для всего алгоритма A . Дадим *рекурсивное* описание метода, решающего эту задачу:

1. Если алгоритм A есть оператор присваивания $x:=e(x) \mid b(x)$, то $v_A(x):=e(x)$ и $d_A(x):=b(x)$.
2. Если алгоритм A есть составной последовательный оператор $R_i; R_j$, то сначала путем рекурсивных вызовов строятся функции $v_i(x)$ и $d_i(x)$ для оператора R_i и функции $v_j(x)$ и $d_j(x)$ для оператора R_j , затем генерируются $v_A(x):=v_j(v_i(x))$ и $d_A(x):=d_i(x)\wedge d_j(v_i(x))$.

3. Если алгоритм A есть составной параллельный оператор $R_i \# R_j$, то сначала путем рекурсивных вызовов строятся функции $v_i(x)$ и $d_i(x)$ для оператора R_i и функции $v_j(x)$ и $d_j(x)$ для оператора R_j , затем генерируются $v_A(x) := (v_i(x), v_j(x))$ и $d_A(x) := d_i(x) \wedge d_j(x)$.
4. Если алгоритм A есть условный оператор $(c(x) \mid u(x)) \rightarrow R_1, R_0$, то сначала путем рекурсивных вызовов строятся функции $v_1(x)$ и $d_1(x)$ для оператора R_1 и функции $v_0(x)$ и $d_0(x)$ для оператора R_0 , затем генерируются $v_A(x) := c(x) \wedge v_1(x) \vee \neg c(x) \wedge v_0(x)$ и $d_A(x) := u(x) \wedge (c(x) \wedge d_1(x) \vee \neg c(x) \wedge d_0(x))$.
5. Если алгоритм A есть оператор цикла $while(c(x) \mid u(x)) R$, то сначала путем рекурсивного вызова строится векторная функция $e(x)$ и характеристическая функция $b(x)$ для оператора R . Затем для построения функций $v_A(x)$ и $d_A(x)$ используется описанный выше метод раскрутки цикла.

11.6. Синтез частично определенного алгоритма

Традиционно алгоритмы создаются человеком. Используя набор базовых операторов, человек шаг за шагом описывает процесс преобразования исходных значений переменных, составляющих память алгоритма, в результирующие значения. Алгоритмическая форма представления преобразования создается обычно в расчете на некоторую абстрактную или реальную вычислительную машину. Она и определяет главным образом характеристики алгоритма.

Выше мы исследовали механизм функционально эквивалентных преобразований частично определенного алгоритма с целью улучшения его характеристик. Предельным случаем преобразований является сведение алгоритма к системе частично определенных логических функций. Поскольку преобразования могут быть выполнены в обратном направлении, то возникает практически важная и теоретически интересная проблема *синтеза алгоритмов*, т.е. проблема перехода от системы логических функций к частично определенному алгоритму. Обычно синтез алгоритма уменьшает сложность представления функций, но при этом увеличивает время вычислений. Оказывается, механизмом уменьшения сложности является увеличение степени неопределенности частей алгоритма

по сравнению со степенью неопределенности исходной системы функций.

Пусть дана векторная частично определенная функция, описанная парой $\mathbf{f}(x) = (v(x) | d(x))$. Требуется *синтезировать условный оператор* (11.20), включающий в качестве составных частей функции $c(x)$, $t(x)$, $b(x)$, $e(x)$ и $u(x)$. Сведем эту общую задачу к двум более частным задачам:

1. Задана функция $\mathbf{f}(x)$ и выбрана функция $c(x)$, необходимо найти функции $t(x)$, $b(x)$, $e(x)$ и $u(x)$;
2. Задана функция $\mathbf{f}(x)$ и выбрана функция $t(x)$, необходимо найти функции $c(x)$, $b(x)$, $e(x)$ и $u(x)$.

Решение первой задачи приводит к построению частично определенного условного оператора вида

$$\begin{aligned} (c(x) | d(x)) \rightarrow x: &= v(x) | d(x) \wedge c(x), \\ &x: = v(x) | d(x) \wedge \neg c(x). \end{aligned} \quad (11.46)$$

Как видим, компонентами синтезированного условного оператора являются: $t(x) = e(x) = v(x)$, $b(x) = d(x) \wedge c(x)$ и $u(x) = d(x) \wedge \neg c(x)$. С первого взгляда создается впечатление, что упрощение сложности представления исходной функции невозможно, поскольку $t(x) = e(x) = v(x)$. Однако более внимательный анализ позволяет увидеть, что функция $t(x)$ минимизируется по области неопределенности $\neg(d(x) \wedge c(x))$, а функция $e(x)$ минимизируется по области неопределенности $\neg(d(x) \wedge \neg c(x))$.

Решение второй задачи приводит к построению частично определенного условного оператора вида

$$\begin{aligned} (t(x) \leftrightarrow v(x) | d(x)) \rightarrow x: &= t(x) | d(x) \wedge (t(x) \leftrightarrow v(x)), \\ &x: = \neg t(x) | d(x) \wedge (t(x) \oplus v(x)). \end{aligned} \quad (11.47)$$

Построение функции $c(x) = t(x) \leftrightarrow v(x)$ выполняется посредством выражения

$$c(x) = \bigwedge_{i=1}^n t_i(x) \wedge v_i(x) \vee \neg t_i(x) \wedge \neg v_i(x) | d(x). \quad (11.48)$$

Сложность функции $t(x)$ определяется при ее выборе. Согласно (11.48), сложность функции $c(x)$ определяется исходной функцией

ей $v(x)$, выбранной функцией $t(x)$ и областью определенности $d(x)$. Тестовая функция $c(x)$, функция $t(x)$, стоящая в *then*-части, и функция $\neg t(x)$, стоящая в *else*-части, минимизируются по области неопределенности $\neg d(x)$.

Пусть теперь для частично определенной функции $\phi(x)$ необходимо *синтезировать циклический оператор* (11.7). Оставим в операторе только одну функцию области определенности $b(x)$, приняв $u(x)=1$. В результате получаем:

$$\text{while } c(x) \quad x:=e(x) \mid b(x). \quad (11.49)$$

Синтез цикла (11.49) сводится к совместному решению системы векторных *логических уравнений*

$$v(x) \wedge d(x) = (\neg c(x) \wedge x \vee c(x) \wedge v(e(x))) \wedge d(x), \quad (11.50)$$

$$d(x) = \neg c(x) \vee b(x) \wedge d(e(x)) \quad (11.51)$$

относительно векторной функции $e(x)$ и скалярных функций $c(x)$ и $b(x)$. Для упрощения задачи синтеза функция $c(x)$ может выбираться заранее и фиксироваться, остается найти функцию $b(x)$ и минимизированную функцию $e(x)$. В упрощенной задаче возникает проблема существования решения системы (11.50), (11.51). Если решение не существует, выбирается другая подходящая функция $c(x)$.

Синтез составного последовательно оператора (11.13) сводится к решению задачи *декомпозиции* одной булевой функции или системы булевых функций на части. Различные методы декомпозиции исследованы в известной литературе.

Процесс синтеза частично определенного алгоритма определяется пошагово и рекурсивно. На каждом шаге синтезируется один из операторов, составные части которого являются системами частично определенных функций, используемых в качестве исходных данных для синтеза вложенных операторов.

11.7. Оптимизация частично определенного алгоритма

Если правила – это средства преобразования алгоритмов, то *параметры алгоритмов* и направления изменения параметров – это база для системы планирования применения правил. В дополнение,

параметры алгоритма и требования к значениям параметров – это средства управления процессом синтеза цифровых систем и компьютерных программ с требуемыми характеристиками.

Важнейшими параметрами алгоритма являются время выполнения и структурная сложность. При правильном планировании процесса преобразований, в общем случае уменьшение времени влечет увеличение сложности и, наоборот, уменьшение сложности влечет увеличение времени. Очевидно, что введение в алгоритм составного последовательного оператора направлено на уменьшение сложности, хотя платой может быть увеличение времени. Очевидно также, что введение в алгоритм составного параллельного оператора направлено на уменьшение времени, хотя платой наверняка является увеличение сложности.

Чтобы управлять параметрами времени выполнения и сложности реализации при введении в алгоритм условного оператора, важной является задача выбора тестовой функции $c(x)$. Предпочтительной является функция, обеспечивающая одинаковую степень неопределенности *then* и *else* частей с целью балансирования возможностями вариации значений параметров этих частей. Анализ показывает, что одним из важнейших классов таких функций является

$$c(x, w) = \bigoplus_{i=1}^n x_i \wedge w_i \quad (11.52)$$

где w – вектор из 0 и 1, выполняющий селекцию переменных для функции c из вектора x . Экспериментальное исследование показало, что такие функции дают в среднем максимальное сокращение не только сложности, но и времени выполнения условного оператора.

Чтобы управлять параметрами времени и сложности при введении в алгоритм циклического оператора, важной является задача выбора минимального, среднего и максимального числа итераций цикла за счет построения подходящих функций $c(x)$ и $e(x)$. Введение цикла позволяет, прежде всего, уменьшить сложность алгоритма за счет увеличения времени его выполнения. Для оптимизации применим метод вытягивания цикла, выполняющий построение и минимизацию частей цикла таким образом, чтобы число итераций пошагово увеличивалось.

11.8. Историческая справка

Классические представления алгоритмов машинами Тьюринга и рекурсивными функциями являются полностью определенными. Неопределенность в программе ассоциировалась с исключительными ситуациями, такими как деление на ноль. Понятие частично определенного логического алгоритма в операторной форме введено в работе А.А. Прихожего [93 (1991)]. Методы преобразования, синтеза и оптимизации частично определенных алгоритмов и входящих в них операторов предложены в работах [57, 61, 64, 66, 68-71, 94-98, 105]. Синтез циклических операторов сведен к решению векторных рекурсивных частично определенных систем логических уравнений.

ЛИТЕРАТУРА

1. Akers, B. Binary Decision Diagrams / B. Akers // IEEE Trans. on Comp. – Vol C-27. – June 1978. – P. 509-516.
2. Ashenhurst, R.L. The Decomposition of Switching Functions / R.L. Ashenhurst // Proc. Intern. Symposium on the Theory of Switching Functions. – 1957. – P. 74-116.
3. Ast, A. Data-procedural Languages for FPL-based Machines / Ast A., Becker J., Hartenstein R., Kress R., Reinig H., Schmidt K. // 4th Int.Workshop FPL'94. – Prague, Czech Republic, September 7-9. – 1994.
4. Bartlett, K.A. Multi-level logic minimization using implicit don't cares / K.A. Bartlett, R.K. Brayton, G.D. Hachtel, R.M. Jacoby, C.R. Morrison, R.L. Rudell, A.L. Sangiovanni-Vincentelli, A.R. Wang // IEEE Trans. on CAD of Integrated Circuits and Systems. – 1988. – P. 723-740.
5. Becker, B. How many Decomposition Types do we need? / B. Becker, R. Drechsler // Proc. ED&TC 1995. – IEEE Computer Society Press, Los Alamitos, California. – 1995. – P. 438-443.
6. Bergamaschi, R.A. Efficient use of large don't cares in high-level and logic synthesis / R.A. Bergamaschi, D. Brand, L. Stok, M. Berkelaar, S. Prakash // Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design. – IEEE Computer Society, Washington, USA. – 1995. – P. 272-278.
7. Bertacco, V. Decision Diagrams and Pass-Transistor Logic Synthesis / V. Bertacco, S. Minato, P. Verplaetse, L. Benini, G. De Micheli // International Logic Synthesis Workshop. – Tahoe City, CA.
8. Boling, B. Improving the variable ordering of OBDDs is NP complete / B. Boling, I. Wegener // IEEE Trans. on Computers. – Vol. 45. – 1996. – № 9.– P. 993-1002.
9. Boole, G. An Investigation of the Laws of Thought / G. Boole. – **Great Britain, Queen's College of Cork, 1853.** – 344 p.
10. Borrione D. Fomal proofs from HDL descriptions / D. Borrione, H. Eveking, L. Pierre // Fundamentals and Standards in Hardware

Description Languages [Edt. J.P. Mermet]. – Kluwer Academic Publishers. – 1993. – P. 231-262.

11. Braun, E.L. Digital Computer Design / E.L. Braun. – Academic Press, New York, 1963. – 293 p.

12. Brent, A.R. A Regular Layout for Parallel Adders / A.R. Brent, H.A. Kung // IEEE Trans. On Computers. – Vol. C31. – March, 1982. – P. 260-264.

13. Bryant, R.E. Graph-based algorithms for Boolean function manipulation / R.E. Bryant // IEEE Transactions on Computers. – Vol. 35. – 1986. – № 8. – P. 677– 691.

14. Chang, C. An efficient algorithm for local don't care sets calculation / C. Chang, M. Marek-Sadowska, K.T. Cheng // DAC '95 Proceedings of the 32nd annual ACM/IEEE Design Automation Conference. – N.Y., USA. – 1995. – P. 663-667.

15. Chang, K.-H. Logic synthesis and circuit customization using extensive external don't-cares / K.-H. Chang, V. Bertacco, I.L. Markov, A. Mishchenko // ACM Trans. Design Autom. Electr. Syst. – Vol. 15. – 2010. – № 3. – P. 1-22.

16. Chang, C. Minimizing ROBDD size of incompletely specified multiple output functions / C. Chang, K.T. Cheng, M. Marek-Sadowska // European Design & Test Conference. – 1994. – P. 620-624.

17. Chen, Y.-C. Node addition and removal in the presence of don't cares / Y.-C. Chen, C.-Y. Wang // Design Automation Conference - DAC. – 2010. – P. 505-510.

18. Damiani, M. Efficient Computation of Exact and Simplified Observability Don't Care Sets for Multiple-level Combinational Networks / M. Damiani, G. De Micheli. – Technical report, Stanford University, Computer Systems Laboratory, 1990. – 12 p.

19. Damiani, M. Observability don't care sets and. Boolean relations / M. Damiani, G. De Micheli // Proc. IEEE Int. Conf. on Computer-Aided Design. – 1990. – P. 502–505.

20. Damiani, M. Derivation of Don't Care Conditions by Perturbation Analysis Combinational Multiple-Level Logic Circuits / M. Damiani

ni, G. De. Micheli // Proc. International Logic Synthesis Workshop. – Raleigh. – May 1991. – P. 1-12.

21. Damiani, M. Don't Care Set Specifications in Combinational and Synchronous Logic Circuits / M. Damiani, G. deMicheli // IEEE Trans. on CAD. – Vol.12. – 1993. – P. 365-388.

22. Davio, M. Discrete and Switching Functions / M. Davio, J.P. Deschamps, A. Thayse. – New York: McGraw-Hill, 1978. – 729 p.

23. Yanushkevich, S.N. Decision Diagram Techniques for Micro- and Nanoelectronic Design, Handbook / S.N. Yanushkevich, D.M. Miller, V.P. Shmerko, R. Stancovich. – CRC Press, Boca Raton, 2005. – 920 p.

24. Drechsler, R. Binary Decision Diagrams / R. Drechsler, B. Becker. – Kluwer Academic Publishers, 1998. – 200 p.

25. Drechsler, R. Fast OFDD based minimization of fixed polarity Reed-Muller expressions / R. Drechsler, M. Theobald, B. Becker // European Conf. on Design Automation. – 1994. P. 2-7.

26. Drechsler, R. Efficient representation and manipulation of switching functions based on ordered Kronecker functional decision diagrams / R. Drechsler, A. Sarabi, M. Theobald, B. Becker, M.A. Perkowski // Design Automation Conference - **CAD'94**. – 1994. – P. 415-419.

27. Drechsler, R. A genetic algorithm for variable ordering / R. Drechsler, B. Becker, N. Gockel // IEE Proceedings. – Vol. 143. – 1996. – № 6. – P. 364-368.

28. Filman, R. Reasoning with Worlds and Truth Maintenance in a Knowledge-Based Programming Environment / R. Filman // Communications of the ACM. – Vol. 31. – 1988. – № 4. – P. 382-401.

29. Gulati, K. Efficient don't care computation for hierarchical designs / K. Gulati, M. Lovell, S.P. Khatri // Proceedings of IEEE International Symposium on Circuits and Systems. – 2006. – P. 1-4.

30. Guyot, A. Adders Synthesis / A. Guyot, M. Belrhiti, G. Bosco. – INPG/TIMA, Grenoble, France, 1997. – 7 p.

31.Hachtel, G.D. Logic Synthesis and Verification Algorithms / G.D. Hachtel, F. Somenzi. – Kluwer Academic Publishers, 1996. – 564 p.

32.Hong, Y. **Safe BDD Minimization Using Don't Cares** / Y. Hong, P.A. Beerel, J.R. Burch, K.L. McMillan // 34th Design Automation Conference. – IEEE Computer Society Press, Los Alamitos, California. – 1997. – P. 208-213.

33.Hong, S.U. Sibling-substitution-based BDD minimization using don't cares / S.U. Hong, P.A. Beerel, J.R. Burch, K.L. McMillan // IEEE Trans. on CAD of Integr. Circuits and Systems. – Vol.19. – 2000. – No.1.– P. 44-55.

34.Karnaugh, M. The Map Method for Synthesis of Combinational Logic Circuits / M. Karnaugh // Trans. AIEE. – Vol. 72. – November 1953. – № 9. – P. 593-599.

35.Karplus, K. Using if-then-else DAGs for Multi-level Logic Minimization / K. Karplus // Technical Report UCSC-CRL-99-29. – University of California, Santa Cruz. – 1988. – P. 49-54.

36.Kebschull, U. Multilevel logic synthesis based on functional decision diagrams / U. Kebschull, E. Schubert, W. Rosenstiel // European Conf. on Design Automation. – 1992. – P. 278-282.

37.Kerber, M. Tableau Calculus for Partial Functions / M. Kerber, M.A. Kohlhase // Annals of the Kurt-Godel-Society. – The University of Birmingham. – Vol. 2. – 1996. – P. 21-49.

38.Kleene, S. Introduction to Metamathematics / S. Kleene. – Noth Holand Pub. C., Grjningen, 1952. – 576 p.

39.Lee, C.Y. Representation of switching circuits by binary decision programs / Lee C.Y. // Bell Syst. Techn. J. Vol 38. – 1959. – N 4.– P. 985-1000.

40.Low Power Design for Deep Submicron Electronics / J. Mermet, W. Nebel Eds. – Kluwer Academic Publishers, 1997. – 580 p.

41.Malik, S. Logic verification using binary decision diagrams in a logic synthesis environment / S. Malik, A. Wang, R. Brayton, A. Sangiovanni-Vincentelli // Proc. Int'l Conf. on CAD-7. – 1988. – P. 616-640.

42.Muller, D.E. Application of Boolean algebra to switching circuit design and to error detection / D. E. Muller // IRE Transactions on Electronic Computers. – 1954. № 3. – P. 6–12.

43.McCluskey, E. Theory of Switching Circuits / E. McCluskey. – McGraw-Hill Book Co., New York, 1965. – 318 p.

44.Fundamentals and Standards in Hardware Description Languages / J.P. Mermet Ed. – Boston: Kluwer Academic Publishers, 1993. – 466 p.

45.Mishchenko, A. SAT-based complete don't-care computation for network optimization / A. Mishchenko, R.K. Brayton // Proceedings Design, Automation and Test in Europe. – Vol. 1. – 2005. – P. 412-417.

46.Mishchenko, A. Scalable don't-care-based logic optimization and resynthesis / A. Mishchenko, R.K. Brayton, J.-H.R. Jiang, S. Jang // Proc. FPGA 2009. – P. 151-160.

47.Oliveira, A.L. Exact minimization of binary decision diagrams using implicit techniques / A.L. Oliveira, L. Carloni, T. Villa, A.L. Sangiovanni-Vincentelli // Technical Report, University of Berkeley, 1996. – 16 p.

48.Pliuskevicius, R. The Saturated Tableaux for Linear Miniscope Horn-like Temporal Logic / R. Pliuskevicius // Journal of Automated Reasoning. – Vol. 13. – 1994. – P. 391-407.

49.Prihozhy, A.A. High-Level Synthesis Methodology / A.A. Prihozhy. – Minsk, Inst. Eng. Cybernetics, National Academy of Sciences, Belarus, 1993. – 47 p.

50.Prihozhy, A.A. AHILES: Performance Driven High-Level Synthesis from VHDL Description / A.A. Prihozhy, A.N.Smolsky // Int. Conf. “**Design Methodologies for Microelectronics**”. – Slovakia, Bratislava. – 1995. – P. 45-52

51.Prihozhy, A.A. Net Scheduling in High-Level Synthesis / A.A. Prihozhy // IEEE Design & Test of Computers. – Spring. – 1996. – P. 26-35.

52.Prihozhy, A.A. Use of VHDL-Based Design Methodology and the AHILES System for Education in Belarus / A.A. Prihozhy // Part of

book “Microelectronics Education”. – World Scientific. – 1996. – P. 217-220.

53.Prihozhy, A.A. If-Diagrams: Theory and Application / A.A. Prihozhy // Proc. of the European Conference PATMOS'97. – UCL, Belgium. – 1997. – P. 369–378.

54.Prihozhy, A.A. Parallel Computing with If-Decision-Diagrams / A.A. Prihozhy, P.U. Brancevich // Proc. of the Int. Conference **PARELEC'98**. – Poland, Technical University of Bialystok. – 1998. – P. 179–184.

55.Prihozhy, A.A. Asynchronous Scheduling and Allocation / A.A. Prihozhy // Proc. of the Conference DATE 98. – IEEE Computer Society, Paris, France. – 1998. – P. 934-935.

56.Prihozhy, A.A. If-Decision Diagram Based Synthesis of Digital Circuits / **A.A. Prihozhy // Proc. Int. Conf. “Information Technologies for Education, Science and Business”**. – Minsk, Belarus. – 1999. – P. 65-69.

57.Prihozhy, A.A. Digital System High-Level Synthesis Technology / A.A. Prihozhy, R.M. Merdjani, S.V. Zemlyanik // Proc. Int. Conf. **“Information Technologies for Education, Science and Business”**. – Minsk, Belarus. – 1999. – P. 145-149.

58.Prihozhy, A.A. Methods of Partial Logic / A.A. Prihozhy // Proc. Int. Conf. on Neuro Computers and Artificial Intelligence. – Brest. – 1999. – P. 46-49.

59.Prihozhy, A.A. Methods of Partial Logic and If-Decision Diagrams for Synthesis and Formal Verification of Digital Circuits / A. Prihozhy, B. Becker. – DAAD Technical Report, Freiburg University, Germany, 2000. – 30 p.

60.Prihozhy A.A. Partial Logic for Knowledge Representation and Deductive Reasoning in Incompletely Specified Domains / A.A. Prihozhy, L.P. Prihozhaya // Proc. 4th IEEE Baltic Workshop on Data-bases and Information Systems, – Vilnius, Technika, Vol. 2. – 2000. – P. 122-140.

61.Prihozhy, A.A. High-Level Synthesis through Transforming VHDL Models / A.A. Prihozhy // **Proc. Int. Conf. “FDL’2000 – Forum on Design Languages”**. – Germany, Tubingen, ECSI. – 2000. – P. 87-96.

62.Prihozhy, A.A. Automatic Parallelization of Net Algorithms / A.A. Prihozhy, R. Merdjani, F. Iskandar // **Proc. Int. Conf. on Parallel Computing in Electrical Engineering – PARELEC’2000**. – Canada, IEEE Computer Society Press. – 2000. – P. 24-28.

63.Prihozhy, A.A. Partial Logic for Knowledge Representation and Deductive Reasoning in Incompletely Specified Domains / A.A. Prihozhy, L.P. Prihozhaya // **Chapter in Book “Databases and Information Systems”**. – Kluwer Academic Publishers. – 2001. – P. 275-295.

64.Prihozhy, A.A. High-Level Synthesis through Transforming VHDL Models / A.A. Prihozhy // **Chapter in Book “System-on-Chip Methodologies and Design Languages”**. – Kluwer Academic Publishers. – 2001. – P. 135-146.

65.Prihozhy, A.A. Techniques for Optimization of Net Algorithms / A.A. Prihozhy, D. Mlynek, M. Solomennik, M. Mattavelli // **Proc. Int. Conf. “PARELEC’2002 – Parallel Computing in Electrical Engineering”**. – IEEE CS, Los Alamitos, California. –2002. – P. 211-216.

66.Prihozhy, A.A. Data Dependences Critical Path Evaluation at C/C++ System Level Description / A.A. Prihozhy, M. Mattavelli, D. Mlynek // **Part of book “Integrated Circuit and System Design”**. – LNCS 2799, Springer. – 2003. – P. 569-579.

67.Prihozhy, A.A. If-Decision Diagram Based Modeling and Synthesis of Incompletely Specified Digital Systems / A.A. Prihozhy, B. Becker // **Electronics and communications, Special Issue on Electronics Design**. – Kyiv. – 2005. – P. 103 – 108.

68.Prihozhy, A.A. Evaluation of Parallelization Potential for Efficient Multimedia Implementations: Dynamic Evaluation of Algorithm Critical Path / A.A. Prihozhy, M. Mattavelli, D. Mlynek // **IEEE Trans. on Circuits and Systems for Video Technology**. – Vol. 15. – May 2005. – No. 5.– P. 593-608.

69.Prihozhy, A.A. Design of Parallel Implementations by Means of Abstract Dynamic Critical Path Based Profiling of Complex Sequen-

tial Algorithms / A.A. Prihozhy, D. Mlynek // **Part of book “Integrated Circuit and System Design”**. – LNCS 4148, Springer. – 2006. – P. 1-11.

70.Prihozhy, A.A. Optimization Methodology for Complex FPGA-based Signal Processing Systems with CAL / A. Ab-Rahman, H. Amer, A.A. Prihozhy, C. Lucarz, M. Mattavelli // Int. Conf. on Design & Architectures for Signal and Image Processing – **DASIP’2011**, Tampere, Finland, Nov. 2-4, 2011. – France, ECSI, IEEE. – P. 1-8.

71.Prihozhy A.A. Pipeline Synthesis and Optimization of FPGA-based Video Processing Applications with CAL / A. Ab-Rahman, A.A. Prihozhy, M. Mattavelli // EURASIP Journal on Image and Video Processing. – Vol. 19. – 2011. – November. – P. 1-28.

72.Quine, W.V. Selected Logic Papers / W.V. Quine. – New York: Random House, 1966. – 250 p.

73.Redford, M. DFM – Don't care or competitive weapon? / M. Redford, J.Sawicki, P. Subramaniam, C. Hou, Y. Zorian, K. Michaels // 46th Design Automation Conference DAC '09. – 2009. – P. 296–297.

74.Reed, I.S. A class of multiple-error-correcting codes and the decoding scheme / I.S. Reed // Transactions of the IRE Professional Group on Information Theory. – 1954. – № 4. – P. 38–49.

75.Robinson, J.A. A Machine-Oriented Logic Based on the Resolution Principle /J.A. Robinson //Journal of the Association for Computing Machinery. – Vol. 12. – 1965. – № 1.– P. 23-41.

76.Roth, J.P. Minimization over Boolean graphs / J.P. Roth, R.M. Karp // IBM Journal. – 1962. – P. 227-238.

77.Sauerhoff, M. On the complexity of minimizing the OBDD size for incompletely specified functions / M. Sauerhoff, I. Wegener // IEEE Trans. on CAD. – Vol. 15. – № 11. – 1996. – P. 1435-1437.

78.Shannon, C.E. The Synthesis of Two Terminal Switching Circuits / C.E. Shannon // Bell System Technical Journal. – Vol. 28. – 1949. – № 1.– P. 59-98.

79.Shiple, T. Heuristic Minimization of BDDs Using Don't Cares / T. Shiple, R. Hojati, A.L. Sangiovanni-vincentelli, R.K. Brayton // 31st Design Automation Conference DAC '94. – San Diego, CA. – 1994. – P. 225-231.

80.Scholl, C. On the generation of Multiplexer Circuits for Pass Transistor Logic / C. Scholl, B. Becker // Proc. DATE'1999. – IEEE Press. – 1999.

81.Trullemans, A.-M. Combined approach of ROBDDs and structural analysis in the mapping and matching of logic functions / A.-M. Trullemans, R. Jacobi, Q. Zhang // Logic and Architecture Synthesis [G.Saucier, A.Magnotte Eds]. – Chapman & Hall, GB. – 1995. – P. 41-56.

82.Valenzise, G. Motion prediction of depth video for depth-image-based rendering using don't care regions / G. Valenzise, G. Cheung, R. Galvao, M. Cagnazzo, B. Pesquet-Popescu, A. Ortega // Picture Coding Symposium (PCS). – 2012. – P. 93-96.

83.Vercest, D. Correctness proof of parameterized hardware modules in the Cathedral-II synthesis environment / D. Vercest, L. Claesen, H. de.Man // Proc. European Design Automation Conference. – Glasgo, U.K. – 1990, 12-15 March.

84.Ueno, Y. TITAC: Design of a Quasi-Delay-Insensitive Microprocessor / Y. Ueno, H. Kagotani, M. Kuwako, A. Takamura // Proc. IEEE Design & Test of Computers. – 1994. – Summer. – P. 50-63.

85.Zadeh, L.A. The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems / L.A. Zadeh // Approximate Reasoning in Expert Systems. – Elsevier Science Publishers, North Holland. – 1985. – P. 3-41.

86.Zhu, Qi. SAT sweeping with local observability don't-cares / Qi. Zhu, N. Kitchen, A. Kuehlmann, A. Sangiovanni-Vincentelli // Proc. Design Automation Conference DAC'06. – 2006. – P. 229-234.

87.Zukoski, A. Reliability-driven don't care assignment for logic synthesis / A. Zukoski, M.R. Choudhury, K. Mohanram // Proc. Design, Automation and Test in Europe. – Vol. 1. – 2011. – P. 1-6.

88.Закревский, А.Д. Алгоритмы синтеза дискретных автоматов / А.Д.Закревский. – М.: Наука, 1971. – 511 с.

89.Закревский, А. Д. Алгоритмы синтеза полиномов, реализующих слабо определенные булевы функции и системы /

А.Д.Закревский //Автоматика и телемеханика. – 2004. – № 6. – С. 158–176.

90.Жегалкин, И. И. О технике вычислений предложений в символической логике / И.И. Жегалкин // Математический сборник. – Т. 34. – 1927. – №1.– С. 9–28.

91.Клини С. Математическая логика / С. Клини. – М.: Мир, 1973. – 280 с.

92.Мищенко, В.А. Логическое проектирование БИС / В.А. Мищенко, А. И. Аспидов, В. В. Витер. – М.: Радио и связь, 1984. – 312 с.

93.Прихожий, А.А. Алогические алгоритмы / А.А. Прихожий. – Минск, ИТК НАНБ, 1991. – 48 с.

94.Прихожий, А.А. Синтез логических алгоритмов / А.А. Прихожий. – Минск, ИТК НАНБ, 1991. – 38 с.

95.Прихожий, А.А. Оптимизация логических алгоритмов / А.А. Прихожий. – Минск, ИТК НАНБ, 1991. – 52 с.

96.Прихожий, А.А. Теория эквивалентных преобразований алгоритмов в САПР СБИС / В.А.Мищенко, А.А.Прихожий. – Минск: Наука и техника, 1991. – 260 с.

97.Прихожий, А.А. Методы и средства для алгоритмического и архитектурного проектирования цифровых СБИС / А.А. Прихожий, В.Г. Брич // Радиотехника. – 1992. – Т.35. – №6. – С. 14-28.

98.Прихожий, А.А. Методы эквивалентных преобразований логических алгоритмов в САПР СБИС / А.А. Прихожий // Весці АН Беларусі, сер. ф.-м.н. – 1992. – №2. – С. 86-92.

99.Прихожий, А.А. Алгебра частичной логики / А.А. Прихожий // Новые информационные технологии в машиностроении и вычислительной технике. – Брест. – 1998. – С. 168-174

100.Прихожий, А.А. Логический вывод в частичной логике / А.А.Прихожий // Интеллектуальные системы. – Минск, ИТК НАН Беларусі. – 1998. – С. 123-140.

101. Прихожий, А.А. Минимизация потребляемой мощности в цифровых схемах / А.А. Прихожий, С.В. Земляник // Материалы

межд. конференция «Моделирование интеллектуальных процессов проектирования и производства (CAD/CAM/*98)». – Минск., ИТК НАНБ. – 1998. – С. 187.

102. Прихожий, А.А. Минимизация активности цифровых схем, представленных if–диаграммами решений / А.А. Прихожий, С.В. Земляник // Цифровая обработка информации в чрезвычайных ситуациях. – Минск, ИТК НАН Беларуси, – 1998.

103. Прихожий А.А. Математические методы и программные средства высокоуровневого структурно-параметрического синтеза цифровых систем / А.А. Прихожий: Диссертация доктора технических наук: 05.13.11. – Минск, 1998. – 496 с.

104. Прихожий, А.А. Сокращение энергопотребления за счет снижения активности систем управления / А.А. Прихожий, С.В. Земляник // Энергетика. – 2000. – № 1. – С. 36-44.

105. Прихожий, А.А. Методология преобразования программного кода с целью улучшения параметров параллельных реализаций аудио кодеков / А.А.Прихожий, М. Соломенник, Д. Млынек // Электроника и связь. – Киев.– 2006.– Том 31. – С. 55-63.

106. Прихожий, А.А. Преобразование if-диаграмм решений в пространстве параметров время-сложность / А.А. Прихожий // Автоматизация проектирования дискретных систем: Материалы Седьмой межд. конф. CAD'10, 16-17 ноября 2010 г., Минск. – Минск: ОИПИ НАН Беларуси. – 2010. – С. 15-22.

107. Прихожий А.А. Обобщение разложения Шеннона для частично определенных булевых функций / А.А. Прихожий // Материалы 10-ой международной научно-технической конференции «Наука – образованию, производству, экономике». – Минск, БНТУ. – 2012. – С. 204.

108. Прихожий, А.А. Взаимосвязь частичных, не полностью и полностью определенных булевых функций в логических системах / А.А. Прихожий // Информатизация образования. – 2012. – № 4. – С. 81-91.

109. Ярмолик, В. Н. Контроль и диагностика цифровых узлов ЭВМ / В. Н. Ярмолик. – Минск: Наука и техника, 1988. – 239 с.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

- $x_i, x_1, \dots, x_n, y_i, y_1, \dots, y_n, \dots$ – скалярные переменные;
 x_i, y_i, a, \dots – векторные переменные;
 $f_i, v_i, d_i, g_i, h_i, \dots$ – булевы полностью определенные функции;
 ϕ, φ, π – не полностью и частично определенные функции;
 Φ, Ψ, Ω, Π –частичные функции;
- \neg – булева операция отрицание (инверсия);
 \wedge – булева операция конъюнкция;
 \vee – булева операция дизъюнкция;
 \rightarrow – булева операция прямая импликация,
обозначение условного оператора;
 \leftarrow – булева операция обратная импликация;
 \oplus – булева операция исключающее или;
 \leftrightarrow – булева операция эквивалентность;
 $/$ – булева операция штрих Шеффера;
 \downarrow – булева операция стрелка Пирса;
 dc – неопределенное (безразличное) значение;
 \sim – частичная операция отрицание (инверсия);
 $\&$ – частичная операция конъюнкция;
 $+$ – частичная операция дизъюнкция;
 \Rightarrow – частичная операция прямая импликация;
 \Leftarrow – частичная операция обратная импликация;
 \odot – частичная операция исключающее или;
 \Leftrightarrow – частичная операция эквивалентность;
 $\♂$ – частичная операция штрих Шеффера;
 $\♀$ – частичная операция стрелка Пирса;
 $\#$ – частичная операция идентификации неопределенности,
параллельный оператор;
- $\alpha(i)$ – число частичных операций, сводящихся к булевой операции i ;
 $\beta(i)$ – число склеенных частичных операций, сводящихся к i ;
 $\gamma(i)$ – число развернутых частичных операций, сводящихся к i ;
 $\chi(f)$ – функция стоимости булевой функции f ;
 $SF(\Phi)$ – сильная форма представления частичной функции Φ ;
 $WF(\Phi)$ – слабая форма представления частичной функции Φ ;

- IfДР – if диаграмма решений;
- MEF – минимизирующее разложение по характеристической функции;
- МЕО – минимизирующее разложение по операции;
- ME2F – минимизирующее разложение по двум операциям;
- PEF – частичное разложение по характеристической функции;
- PEO – частичное разложение по операции;
- PE2F – частичное разложение по двум операциям;
- VCR – Value-Care Representation (форма значение/область определенности)
- ДIfДР – двоичная if диаграммы решений;
- ДНФ – дизъюнктивная нормальная форма;
- ДР – диаграмма решений;
- ДДР – двоичная диаграмма решений;
- ИДIfДР – двоичная if диаграммы решений с инвертированными дугами
- ИСУДДР – сокращенная упорядоченная двоичная диаграмма решений с инвертированными дугами
- КIfДР – комбинированная if-диаграмма решений;
- КНФ – конъюнктивная нормальная форма;
- КФДР – функциональная диаграмма решений Кронекер;
- НПСДР – не полностью специфицированная диаграмма решений;
- оФДР – отрицательная функциональная диаграмма решений;
- оФIfДР – отрицательная функциональная if-диаграммы решений;
- оФРМ – форма Рида-Маллера отрицательной полярности;
- ПДДР – полная двоичная диаграмма решений;
- пФДР – положительная функциональная диаграмма решений;
- пФIfДР – положительная функциональная if-диаграммы решений;
- пФРМ – форма Рида-Маллера положительной полярности;
- СвДДР – свободная двоичная диаграмма решений;
- СДIfДР – сокращенная двоичная if диаграммы решений;
- СДР – сокращенная диаграмма решений;
- СУДДР – сокращенная упорядоченная двоичная диаграмма решений;
- СУЧТДДР – сокращенная упорядоченная двоичная диаграмма решений с тремя типами терминальных вершин;
- ТДР – троичная диаграмма решений;

УДДР – упорядоченная двоичная диаграмма решений;
фФРМ – форма Рида-Маллера фиксированной полярности;
ФРМ – форма Рида-Маллера общего вида;
ФРМСП – форма Рида-Маллера смешанной полярности;
ЧТДДР – двоичная диаграмма решений с тремя типами терминаль-
ных вершин;
ЧТД|fДР – двоичная if диаграмма решений на множестве из трех
меток;
Ч2ДДР – представление частично определенной функции парой
ДДР;
Ч2Д|fДР – представление частично определенной функции парой
Д|fДР;

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Активность схемы 263

алгоритм

– генетический 219, 224

– логический 298

– частично определенный 298, 302

атомарная формула 277

Базовые частичные операции

– одноместные 31

– двухместные 39

булева переменная

– векторная 14

– скалярная 14

булева операция над диаграммами решений 216, 226

булева разность 15

булева функция 14

Вентиль 238

верификация 271

вероятность 263

верхняя оценка 169

ветвящийся алгоритм 306

временной параметр 246

выводимость алгоритма 303

Генетический алгоритм

– минимизации диаграммы решений 224

– упорядочения переменных 218

генотип 219

Двоичная диаграмма решений 177

– сокращенная упорядоченная 18

– сокращенная упорядоченная с тремя типами терминальных вершин 210

– с тремя типами терминальных вершин 209

двоичная if диаграмма решений 187, 192

– на множестве из трех меток 212

– с инвертированными дугами 196

– сокращенная 192

– упорядоченная 195

декомпозиция 321

дерево поиска 165

дерево разбора 40

диаграмма решений 176

– для частично определенной функции 208

– не полностью специфицированная 211

– полная 177

– свободная 177

– с инвертированными дугами 185

– сокращенная 183

– упорядоченная 177

диапазон булевых функций 129, 131

дизъюнктивная нормальная форма

– совершенная 16

– сокращенная 20

– кратчайшая 21

дизъюнктивно-конъюнктивная форма частично определенной функции 123

дизъюнкция

– булева 14

– частичная 66

доопределение 20, 228

Задача оптимизации 129

закон

– аннигилятора 41, 82

– ассоциативности 41, 82

– двойного отрицания 41, 83

– де Моргана 42, 83

– дистрибутивности 41, 82

– идемпотентности 41, 82

– идентичности 41, 82

– исключенного третьего 43, 84

– коммутативности 41, 82

– определения частичной дизъюнкции через исключающее или 44, 86

- поглощения 41, 82
- поглощения литерала 44, 85
- склеивания конъюнкций 44, 84
- законы
- преобразования функции значения 75
- смешанной алгебры 74
- защитное выражение 299
- знания 273
- значение
- булево 14
- частичное 19

Идентичность

- булева 31
- частичная 30
- иерархическое частичное разложение 110
- изоморфные подграфы 222
- импликанта 22
- простая 22
- импликация
- обратная булева 33
- обратная частичная 36, 66
- прямая булева 33
- прямая частичная 36, 66
- инверсия
- булева 14
- частичная 30
- инвертированная дуга 185, 196
- индивидуальная константа 276
- индивидуум 219
- исключающее или
- булево 33
- частичное 36, 66
- if диаграмма решений 186

Карта Карно 20

- Квайна-МакКласки метод 22
- Квайна матрица 22, 55
- класс Т-эквивалентных частичных функций 28
- клика графа 240
- клез 284
- в имплекативной форме 289

- не полностью определенный 293
- конвейер 316
- конечный автомат константа 30
- конституента 26
- конъюнктивная нормальная форма
- совершенная 17
- конъюнктивно-дизъюнктивная форма
- частично определенной функции 125
- конъюнкция
- булева 14, 33
- частичная 36, 66
- кофактор 15
- исключающий отрицательный 77
- исключающий положительный 77
- отрицательный 15, 76
- полный исключающий положительный 78
- полный исключающий отрицательный 78
- положительный 15, 76
- коэффициент распараллеленности 240
- критический путь 235
- куб 14

Литерал 14, 277

- логический вывод 285
- логическое уравнение 321

Матрица распараллеленности 239

- метод
- ветвей и границ 171
- граничных точек 135
- диапазонов функций 135
- сечений 232
- минимизирующее разложение
- по двум операциям 154
- по операции 144
- по операции исключающее или 159
- по характеристической функции 151
- минтерм 14
- множество
- безразличия 19, 60
- включения 19, 59

- выключения 19, 59
- моделирование 238, 246, 256
- мощность 262
- мультиплексор 241
- мутация 219

Неопределенное (безразличное) значение 19, 26, 273

- не полностью определенная функция 19, 59
- не полностью специфицированная диаграмма решений 211
- несущественная переменная 171
- нижняя оценка 169

Обмен соседних переменных 219

- оператор присваивания 300
- частичный векторный 300
- операция
 - двухместная над частично определенными функциями 65
 - идентификации неопределенности 33, 46, 117
 - минимизации 130
 - минимизации диаграммы 228
 - одноместная над частично определенными функциями 62
- опровержение 286
- оптимальный порядок переменных 178, 218
- оптимизация
 - иерархического разложения 164
 - частично определенного алгоритма 321
- ортогональные наборы 26
- ортогональные остаточные функции 172
- ортогональные функции 164
- отрицание
 - булево 14, 31
 - частичное 30

Память алгоритма 298

параллельные вычисления 239

параметр алгоритма 321

340

переменная значения 58

переменная области определенности 58

поглощение 22, 46

покрытие 23

полнота 304

популяция 219

правило логического вывода 285

правило преобразования 303

- функционально эквивалентное 303

предикатный символ 276

предметная переменная 276

преобразование

- диаграммы 266

- алгоритма 302, 321

принцип резолюций Робинсона 273

- обобщенный 286

приоритет операций 39

продукционная система 275

проекция 14

Разложение

- Давио отрицательное 15

- Давио положительное 15

- Шеннона 15

- частично определенной функции 90

раскрутка 312, 313, 317

резольвента 286

- обобщенная 288

рекомбинация 219

рекурсия 311, 318

Рида-Маллера форма 17

- Кронекер 18, 182

- общего вида 191

- отрицательной полярности 17, 181, 190

- положительной полярности 17, 179, 189

- псевдо Кронекер 18

- смешанной полярности 182

- фиксированной полярности 18

- частично определенной функции 126

Семантика 278

сильная форма представления

частичной функции 45
 Сильный закон
 – в смешанной алгебре 83
 – в частичной алгебре 40
 синтаксис 276
 синтез
 – алгоритма 319
 – условного оператора 320
 – циклического оператора 321
 система эквивалентных преобразований 303
 склеиваемые наборы 27
 скрещивание 219
 слабая форма представления частичной функции 51
 слабый закон
 – в смешанной алгебре 84
 – в частичной алгебре 43
 смешанная алгебра 71
 собственные законы смешанной алгебры 87
 совершенная дизъюнктивно-конъюнктивная форма частично определенной функции 116
 совершенная конъюнктивно-дизъюнктивная форма частично определенной функции 121
 совершенная нормальная форма по исключающему или 17
 сокращение глубины диаграммы 232, 234
 соседние наборы 20
 составной параллельный оператор 301
 составной последовательный оператор 301
 список типов декомпозиции 182
 стрелка Пирса
 – булева 33
 – частичная 36, 66
 сумматор 244
 – последовательный 248
 – параллельный 249

Таблица истинности 16
 теорема 286
 терм 277
 трансляция частичных импликаций 292
 троичная диаграмма решений 211

Умножитель 256
 уровень распараллеленности 239
 условный оператор 300
 – частичный 300

Форма значение/область определенности 58
 формула частичной алгебры 39
 фрейм 275
 функциональная диаграмма решений
 – Кронекер 182
 – отрицательная 180
 – положительная 179
 функциональная if-диаграмма решений 202
 – комбинированная 191
 – отрицательная 190
 – положительная 189
 – с инвертированными дугами 205
 – сокращенная 202
 функционально полный базис 55
 функция
 – значения 74
 – Клини 26
 – области определенности 20
 – полезности 219
 – стоимости 130, 166

Хромосома 220

Цель 286
 циклический оператор 301
 – частичный 301

Частичная алгебра 26
 частичная векторная переменная 26
 частичная логика 273
 – предикатов первого порядка 276

- частичная операция 29
 - двухместная 31
 - одноместная 29
 - склеенная 29, 32
 - развернутая 29, 32
 - частичная дизъюнктивно-
 - конъюнктивная форма 51
 - частичная конъюнктивно-дизъюнктивная форма 52
 - частичная минимальная дизъюнктивно-
 - конъюнктивную форма 54
 - частичная скалярная переменная 26, 39
 - частичная совершенная дизъюнктивно-
 - конъюнктивная форма 45
 - частичная совершенная конъюнктивно-
 - дизъюнктивная форма 45
 - частичная сокращенная дизъюнктивно-
 - конъюнктивная форма 52
 - частичная функция 26
 - корректно определенная 26
 - развернутая 27
 - склеенная 27
 - частично определенная функция 60
 - представление 208, 213
 - частичное правило 285
 - частичное разложение 89
 - по двум частичным операциям 103
 - по операции исключающее или 108
 - по частичной операции 90
 - по характеристической функции 96
 - частичный квантор
 - существования 277, 278
 - общности 277, 278
 - частичный кюз 284
 - Хорна 285
 - число частичных операций 26
- Ш**трих Шеффера
 - булев 34
 - частичный 36, 66
- Э**квивалентность
 - булева 33
 - частичная 36, 66
- эквивалентные алгоритмы 303
- эксперимент 221, 224, 269
- Я**дро покрытия 24

Научное издание

ПРИХОЖИЙ Анатолий Алексеевич

**ЧАСТИЧНО ОПРЕДЕЛЕННЫЕ
ЛОГИЧЕСКИЕ СИСТЕМЫ И АЛГОРИТМЫ**

Подписано в печать 01.03.2013. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.

Усл. печ. л. 19,94. Уч.-изд. л. 15,59. Тираж 100. Заказ 128.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет. ЛИ № 02330/0494349 от 16.03.2009. Пр. Независимости, 65. 220013, г. Минск.