Глава 2

Модели конечных автоматов в виде булевых формул и их оптимизация

Кузнецов Борис Павлович, д.т.н. МВАК
МУФО (Великобритания, Испания, Россия, США, Шри-Ланка),
декан ФВТИ

Булевы формулы наиболее часто употребляются в алгоритмах логического управления различными техническими средствами. Булевы формулы, рассматриваемые ниже, будем считать минимизированными и заданными, в общем случае, скобочной формой в базисе «И, ИЛИ, НЕ». Кроме того, без потери общности, будем считать эти формулы бесповторными [19]. Считаем, что булева формула содержит h букв в отличие от общепринятого обозначения n аргументов булевой функции, при этом $n \le h$.

В разделе 2.1. представлены основные модели вычисления булевых формул: - операторные схемы, линейные бинарные графы, ортогональные формы, а также введенные автором нечеткие множества переходов. Там же предложена система алгоритмов взаимно однозначных преобразований указанных четырех моделей вычисления булевых формул.

В разделе 2.2. рассматриваются статические критерии сложности булевых формул, не зависимые от статистических параметров входящих в формулы букв. При этом анализируются число и суммарная длина путей в линейных бинарных графах. Кроме того, предложен нечеткий критерий, позволяющий оценивать сложность булевой формулы посредством лингвистических переменных («идеальная формула», «малая сложность», «средняя сложность», «большая сложность», «очень большая сложность»).

В разделе 2.3. представлены условия оптимальности выбранной (путем перестановок букв и подформул) записи булевой формулы по критерию минимума числа путей в соответствующем линейном бинарном графе, а также алгоритм получения оптимальной записи булевых формул путем целенаправленных перестановок букв и подформул.

Однако, оптимизация булевой формулы по статическому критерию сложности не всегда обеспечивает минимум длительности вычисления ее значения, так как не учитываются статистические параметры букв, в частности вероятность появления и время вычисления как единичного так и нулевого значений входящих в формулу букв. Поэтому в разделе 2.4. исследуется аддитивный критерий сложности булевой формулы и предлагается соответствующий алгоритм ее оптимизации. Аддитивным критерий назван потому, что на каждом пути в линейном бинарном графе соответствующие показатели букв складываются [307].

Так как существует аддитивный критерий сложности булевой формулы, то имеет смысл рассмотреть и мультипликативный критерий. Мультипликативный означает [307], что соответствующие показатели букв, в частности достоверность единичного (нулевого) значения, перемножаются на каждом пути линейного бинарного графа. Поэтому такой критерий анализируется в разделе 2.5. и предлагается соответствующий алгоритм оптимизации по нему.

В связи с тем, что в скором будущем ожидается широкое использование экспертных систем в управлении различными техническими средствами, автором предлагается в разделе 2.6. метод оптимизации продукционной базы знаний по длительности и достоверности вычислений на базе результатов, полученных в разделах 2.4. и 2.5.

2.1. Модели вычисления булевых формул

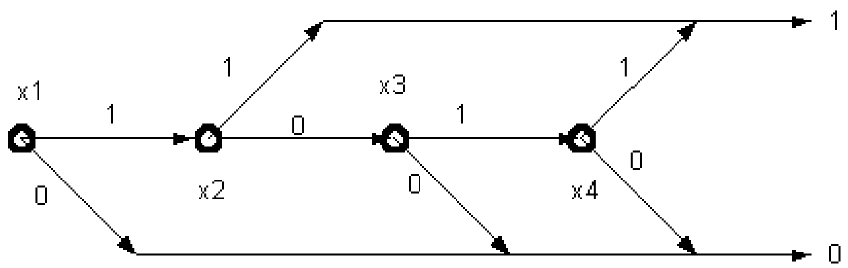
В данном разделе упоминается операторная схема вычисления булевой формулы, подробно рассматриваются линейные бинарные графы, представлена ортогональная дизъюнктивная нормальная форма как совокупность проверяющих тестов, а также предложено нечеткое множество переходов в качестве еще одной аналитической формы представления булевой формулы.

2.1.1. Операторные схемы

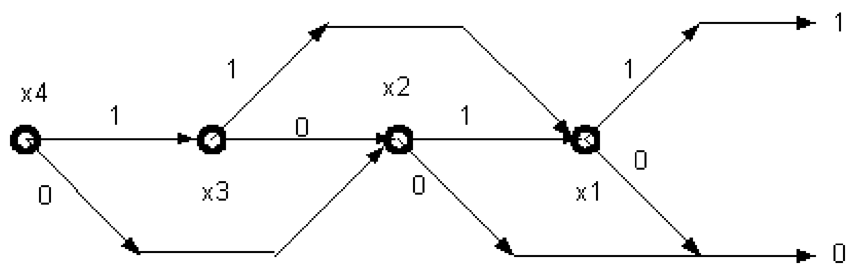
Простейшей моделью вычисления булевой формулы является операторная схема вычислений [183]. При этом вычисления выполняются непосредственно с использованием поразрядных логических операций “&”, “|”,”~” (алгоритмический язык С++ [278] вычислителя). При необходимости применяется стек для вычисления промежуточных значений, например, выражений в скобках. Данная модель вычисляет значение булевой формулы с фиксированным временем, не зависящим от конкретных значений переменных этой формулы, что является ее первым недостатком. Другим недостатком такой схемы вычислений может являться наличие стека, размер которого зависит как от типа формулы, так и от перестановки ее членов. Наиболее существенным недостатком операторной схемы является невозможность выполнять правильные вычисления значений булевых формул, буквы которых являются произвольными целочисленными функциями, что имеет место в алгоритмах контроля и управления электроэнергетических установок. Разновидностью операторной схемы является так называемая польская запись, используемая в большинстве компиляторов. о недостатках которой также указано в главе 6.

2.1.2. Линейные бинарные графы

Основной рассматриваемой вычислительной моделью булевой формулы является бинарная программа [183]. Графическим представлением этой модели вычисления булевой формулы является линейный бинарный граф (ЛБГ) (рис.2.1) [180].



y = x1(x2 v x3 x4)
а)



y = (x4 x3 v x2) x1
б)

Рис. 2.1.

Для дальнейшего понимания перечислим основные свойства ЛБГ, изложенные в работе [176] .

1. Вершина ЛБГ соответствует букве булевой формулы. Порядковый номер буквы соответствует такому же порядковому номеру вершины ЛБГ.

2. Из каждой вершины ЛБГ исходят всегда две дуги, одна из которых соответствует единичному значению буквы формулы, а вторая – нулевому.

3. Две соседние вершины всегда соединены дугой, такие дуги будем называть основными дугами ЛБГ или основной дугой вершины из которой она исходит.

4. Из крайней справа вершины ЛБГ исходят две результирующий дуги: единичная соответствует вычисленному единичному значению булевой функции, соответствующей рассматриваемой формулы, нулевая – нулевому. Эти дуги будем называть единичным и нулевым выходом ЛБГ, соответственно.

5. ЛБГ всегда планарен.

Не основные дуги будем называть переходами. Длиной g_i перехода из i-й вершины в j-ю вершину будем считать число вершин, размещенных в ЛБГ между инцидентными переходу вершинам (число огибаемых дугой перехода вершин). При этом

$$g_i = j - i - 1 .$$

Для ЛБГ, соответствующего произвольной БФ из h букв, имеет место соотношение

$$1 \le g_i \le h - 2.$$

Единичный (нулевой) переход из i-й вершины соответствует маскированию значением «единица» («ноль») i-й буквы формулы выражения, размещенного между i-й и j-й буквами формулы, где j – вершина, в которую направлен переход.

Остовом ЛБГ будем называть суграф [218] ЛБГ, включающий все его вершины, все основные дуги и обе выходных дуги.

Известные алгоритмы синтеза бинарных программ по заданной булевой формуле [52,132,190,206] основаны на суперпозиции ЛБГ, реализующих элементарные логические операции. Эти алгоритмы не обеспечивали планарности графа. В [180] представлен алгоритм синтеза ЛБГ, основанный на так называемых «остатках» булевой формулы. Представим другой алгоритм синтеза ЛБГ по заданной булевой формуле, основанный на ее линейном обходе. Сущность его заключается в том, что по заданной булевой формуле сначала строится остов ЛБГ, затем последовательно строятся переходы и выходы. Изложим этот алгоритм.

1. Заданную булеву формулу преобразуем к нормальному виду [19].

2. Полученную формулу заменим однотипной с ней положительно монотонной формулой [19].

3. Строим остов ЛБГ из h вершин, помечаемых соответствующими буквами формулы по п.2.

4. Основные дуги остова ЛБГ помечаем единицей, если буква в формуле по п.2 предшествует знаку конъюнкции, нулем – в противном случае.

5. Из каждой вершины строим начало перехода, направляемого вверх (вниз), если соответствующая основная дуга из этой вершины помечена нулем (единицей).

6. Каждый переход (из i-й вершины) направляем к j-й вершине, соответствующей j-й букве формулы по п.2, так, что выражение в формуле, размещенное между i-й и j-й буквами маскируется значением i-й буквы, помечающего данный переход.

7. Если i-я буква формулы по п.1 имеет знак инверсии, то дуги, исходящие из i-й вершины ЛБГ, меняют знак на противоположный.

На рис. 2.1 приведен пример построения ЛБГ для двух вариантов перестановки одной и той же булевой формулы. Из этого рисунка следует, что перестановка членов булевой формулы приводит к ЛБГ различной структуры.

2.1.3. Ортогональные формы

Другой важнейшей моделью вычислений булевой формулы является ее ортогональная дизъюнктивная нормальная форма (ОДНФ) [77,252,248] и ОДНФ ее инверсии. Каждая конъюнкция ОДНФ формулы соответствует одному и только одному пути в ЛБГ от начальной вершины к единичному выходу. Каждая конъюнкция ОДНФ инверсии формулы (ОДНФ') соответствует единственному пути от начальной вершины к нулевому выходу. Дизъюнкция ОДНФ и ОДНФ' соответствует полному перечню путей ЛБГ. Буква конъюнкции, входящей в ОДНФ (ОДНФ') без знака инверсии, соответствует единичной дуге, исходящей из одноименной его вершины, со знаком инверсии – нулевой дуге, соответственно.

Запишем ОДНФ и ОДНФ' для формул, приведенных на рис. 2.1.

Для формулы $y = x_1(x_2 \vee x_3x_4)$

ОДНФ: $y = x_1x_2 \vee x_1{\sim}x_2x_3x_4$.

ОДНФ': $y = {\sim}x_1 \vee x_1{\sim}x_2{\sim}x_3 \vee x_1{\sim}x_2x_3{\sim}x_4$.

Для формулы $y = (x_4x_3 \vee x_2)x_1$

ОДНФ: $y = {\sim}x_4x_2x_1 \vee x_4x_3x_1 \vee x_4{\sim}x_3x_2x_1$.

ОДНФ': $y = {\sim}x_4{\sim}x_2 \vee {\sim}x_4x_2{\sim}x_1 \vee x_4x_3{\sim}x_1 \vee x_4{\sim}x_3{\sim}x_2 \vee$

$\vee x_4{\sim}x_3x_2{\sim}x_1$.

Таким образом, мы представили три основных модели вычислений булевой формулы, между которыми существует взаимно однозначное соответствие. Помимо этого, автором предложена [180] полная система алгоритмов взаимно однозначных преобразований:

а) булевой формулы в ЛБГ;

б) ЛБГ в булеву формулу (ретрансляция);

в) булевой формулы в ОДНФ и ОДНФ’;

г) ОДНФ и ОДНФ’ в булеву формулу;

д) ЛБГ в ОДНФ и ОДНФ’ (перечисление путей ЛБГ);

е) ОДНФ и ОДНФ’ в ЛБГ.

Однако перечисленных вычислительных моделей недостаточно для наглядного представления о допустимых перестановках булевых формул.

2.1.4. Нечеткие множества переходов

Построим гистограмму переходов ЛБГ для обоих рассматриваемых на рис.2.1 случаев следующим образом. Слева направо разместим столбики гистограммы, соответствующие переходам из вершин ЛБГ, порядковый номер которых есть порядковый номер как буквы в формуле, так и столбика. Высота столбика равна абсолютной величине длины перехода из соответствующей вершины ЛБГ, а знак перехода указан внутри столбика (рис. 2.2,а,б).

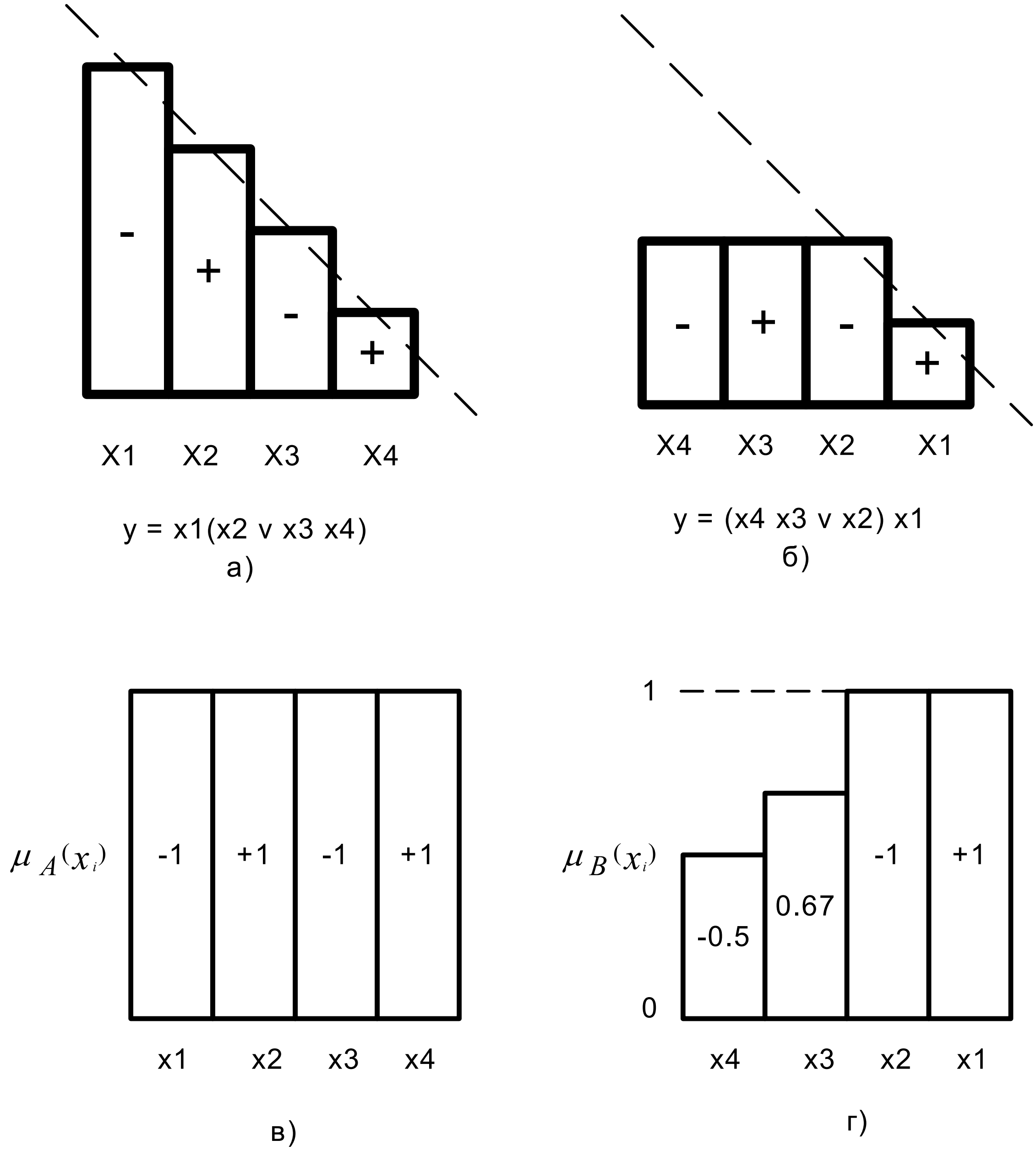


Рис. 2.2

Наклонная линия на рис. 2.2,а соответствует идеальной записи булевой формулы. Эта линия проходит через середину верхнего обреза столбиков гистограммы, соответствующей идеальной записи булевой формулы, например конъюнкции, с тем же числом букв, что и исследуемая булева формула. Наклон этой прямой равен 135°.

Сложность булевой формулы в принятой записи (варианте перестановки) определяется отклонением высоты столбиков гистограммы от указанной наклонной линии. Чем больше отклонение, тем сложнее данная запись булевой формулы. На рис. 2.2,а указанное отклонение отсутствует, и соответствующая запись булевой формулы является наиболее простой. На рис. 2.2,б отклонение максимально, что свидетельствует о наиболее сложной записи данной формулы. Следовательно, надо выбрать первый вариант записи согласно рис. 2.2,а.

Введем новую модель булевой формулы, названную нечетким множеством переходов в соответствующем этой формуле ЛБГ. Нечеткое множество A задается перечислением элементов множества $x_i \in A$ с указанием значения функции принадлежности [147] соответствующего перехода идеальной формуле вида $\mu_A(x_i)$:

(2.1) $$A = \{(x_1/\mu_A(x_1)),\dots,(x_i/\mu_A(x_i)),\dots,(x_h/\mu_A(x_h))\}.$$

Элементами нечеткого множества будем считать вершины ЛБГ, а значениями соответствующей функции принадлежности будем считать величину

(2.2) $$\mu_A(x_i) = g_i / (h - i + 1),$$

где i - порядковый номер буквы в формуле, h - число букв в формуле, g_i - длина перехода из i-й вершины ЛБГ.

Величину g_i можно получить из $\mu_A(x_i)$ посредством вычисления вида

(2.3) $$g_i = (h - i + 1)\mu_A(x_i)$$

В качестве примера рассмотрим нечеткие множества переходов для вариантов ЛБГ, изображенных на рис. 2.1. Обозначим А – множество переходов для варианта «а», В – множество переходов для варианта «б». Следуя представленному определению, имеем:

$$A = \{(x_1 / -4/4), (x_2 / 3/3), (x_3 / -2/2), (x_4 / 1/1)\} =$$
$$= \{(x_1 / -1), (x_2 / 1), (x_3 / -1), (x_4 / 1)\}.$$
$$B = \{(x_4 / -2/4), (x_3 / 2/3), (x_2 / -2/2), (x_1 / 1/1)\} =$$
$$= \{(x_4 / -0.5), (x_3 / 0.67), (x_2 / -1), (x_1 / 1)\}.$$

С целью обеспечения наглядности этих множеств, введем графическое их изображение в виде гистограмм (рис. 2.2,в,г). При этом элементы (столбцы) гистограммы обозначают функцию принадлежности и помечаются согласно выражениям А и В. Такие гистограммы будем именовать гистограммами нечетких множеств переходов.

Из рис. 2.2,в следует, что функция принадлежности неизменна для всех букв формулы и равна единице, что свидетельствует об идеальной записи булевой формулы (рис. 2.1,а). Напротив, из рис. 2.2,г вытекает утверждение о не оптимальности данного варианта булевой формулы (рис. 2.1,б), так как значения функции принадлежности много ниже единицы у первых двух букв.

Представим упрощенные алгоритмы **взаимно однозначных преобразований** линейного бинарного графа, булевой формулы и ортогональной формы в нечеткое множество переходов и обратные преобразования. При этом считаем булеву формулу положительно монотонной.

<u>Преобразование линейного бинарного графа в нечеткое множество переходов.</u>

1) Рассматриваем очередную i-ю по порядку вершину (начиная слева с первой вершины, для которой положим $i = 1$) ЛБГ, и записываем ее обозначение как соответствующий i-й элемент нечеткого множества переходов.

2) Определяем длину g_i перехода из данной вершины (как указано выше) и направление перехода.

3) Записываем x_i и величину (2.2) с соответствующим знаком как значение функции принадлежности i-го элемента нечеткого множества переходов.

4) Если $i < h$, то $i = i + 1$ и возврат к п.1.

Иначе Конец преобразования.

Преобразование булевой формулы в нечеткое множество переходов.

1) Полагаем $i = 1$.

2) Рассматриваем i-ю по порядку слева букву формулы и записываем ее как соответствующий i-й элемент нечеткого множества переходов.

3) Определяем длину g_i перехода и знак значения функции принадлежности следующим образом.

Если справа от буквы следует знак дизъюнкции, то определяем число букв в выражении, маскируемом единичным значением рассматриваемой буквы, добавляем единицу и получаем искомую длину. Искомый знак устанавливаем в плюс.

Если справа от буквы следует знак конъюнкции, то определяем число букв в выражении, маскируемом нулевым значением рассматриваемой буквы, добавляем единицу и получаем искомую длину. Искомый знак устанавливаем в минус.

4) Записываем x_i и величину (2.2) с соответствующим знаком как значение функции принадлежности i-го элемента НМП.

5) Если $i < h$, то $i = i + 1$ и возврат к п.1.

Иначе Конец преобразования.

Преобразование ОДНФ и ОДНФ’ в нечеткое множество переходов.

Напомним, что ОДНФ – это ортогональная дизъюнктивная нормальная форма, полученная из исходной дизъюнктивной нормальной формы по правилам ортогонализации. ОДНФ’ - то же для инверсии исходной формулы.

1) Составляем список конъюнкций из ОДНФ и ОДНФ’.

2) Определяем порядок следования букв в формуле по наибольшей конъюнкции из полученного списка (без возможного знака инверсии).

3) Полагаем $i = 1$.

4) Рассматриваем i-ю по порядку букву, и записываем ее как соответствующий i-й элемент нечеткого множества переходов.

5) Ищем в списке конъюнкцию, в которой рассматриваемая буква соседствует с буквой, имеющей наибольший номер j.

6) Определяем длину перехода, равную $g_i = j – i + 1$, со знаком минус, если в конъюнкции, найденной в п.5, буква содержит знак инверсии.

7) Записываем значение функции принадлежности для i-го элемента нечеткого множества переходов как величину (2.2).

7) Если $i < h$, то $i = i + 1$ и возврат к п.4.

Иначе Конец преобразования.

Преобразование нечеткого множества переходов в линейный бинарный граф.

1) Полагаем $i = 1$.

2) Рассматриваем i-й элемент нечеткого множества переходов, начиная слева. В строящийся линейный бинарный граф включаем i-ю вершину с обозначением буквы, заимствуемой из рассматриваемого элемента нечеткого множества переходов.

3) Если $i = h$ (последний элемент нечеткого множества переходов), то из соответствующей вершины линейного бинарного графа направляем две дуги – выходы (единичный и нулевой) и переходим к п.7.

Иначе перейти к п.4.

4) Строим основную дугу из i-й вершины графа. Помечаем ее единицей, если соответствующий элемент нечеткого множества переходов имеет знак минус при значении функции принадлежности, или нулем – в противном случае.

5) Строим переход из i-й вершины линейного бинарного графа. Переход направляем вверх, если элемент нечеткого множества переходов имеет знак плюс при значении $\mu_A(x_i)$, и вниз – в противном случае. Определяем длину g_i перехода согласно (2.3). Переход направляем в позицию $(i + g_i - 1)$ строящегося линейного бинарного графа.

6) $i = i + 1$. Перейти к п. 2.

7) Конец синтеза линейного бинарного графа.

<u>Преобразование нечеткого множества переходов в булеву формулу.</u>

1) Полагаем $i = h$. (Начинаем просмотр указанного множества с последнего элемента.)

2) Рассматриваем i-й элемент нечеткого множества переходов. Обозначение данного элемента вставляем в синтезируемую формулу в качестве i-ой по порядку буквы.

3) При $i < h$ определяем знак, который следует поставить справа от вставленной в формулу буквы. Если $\mu_A(x_i) < 0$, то это – знак конъюнкции, в противном случае – дизъюнкции.

4) Определяем маскируемое выражение в синтезируемой формуле. Определяем по соотношению (2.3) длину перехода g_i. Определяем наименьший порядковый номер j буквы в формуле, на которую не распространяется маскирование ($j = i + g_i$). Выражение, образованное буквами с номерами $i + 1,\ldots,j - 1$ и знаками операций, их объединяющих, является искомым.

5) Если маскируемое выражение содержит знак дизъюнкции, то оно заключается в скобки. При этом, если закрывающая скобка уже стоит, и парная ей открывающая скобка отсутствует, то она не повторяется.

6) Если перед маскируемым выражением указан знак дизъюнкции, то после него ставится закрывающая скобка.

7) Если $i > 1$, то $i = i - 1$, и переходим к п.2.

8) Если в формуле осталась закрывающая скобка без парной открывающей, то перед первой буквой формулы ставится открывающая скобка.

9) Конец преобразования (синтеза булевой формулы).

Преобразование нечеткого множества переходов в ортогональную форму не рассматриваем ввиду его сложности, так как ортогональную форму легко получить из линейного бинарного графа, синтезируемого из нечеткого множества переходов предложенным алгоритмом.

Отметим, что не всякое нечеткое множество можно считать нечетким множеством переходов. Сформулируем необходимые условия, при которых нечеткое множество вида

$A = \{(x_1/\mu_A(x_1)),\ldots,(x_i/\mu_A(x_i)),\ldots,(x_h/\mu_A(x_h))\}$,

является нечетким множеством переходов, соответствующей некоторой булевой формуле:

(а) $\mu_A(x_h) \equiv 1$.

(б) $|\mu_A(x_{h-1})| \equiv 1$.

(в) $1 \geq |\mu_A(x_i)| > 0,\ i = 1,\ldots,h - 2$.

Наглядно, эти условия можно наблюдать из гистограммы, построенной по нечеткому множеству переходов. При этом все столбики гистограммы должны находиться не выше прямой идеальной функции принадлежности (штриховая линия на рис. 2.2,г), описываемая уравнением $|\mu_A(x_i)| = 1,\ \ i = 1,\ldots,h$.

Таким образом, рассмотрены три вычислительные модели булевой формулы: ЛБГ, ОДНФ и НМП. При этом, ЛБГ задает схему вычислений значений булевой формулы, ОДНФ и ОДНФ' задают минимальное множество тестов, проверяющих правильность программной реализации булевой формулы, а НМП, в частности, его гистограмма – наглядно характеризует отклонение выбранного варианта записи булевой формулы от идеального варианта.

2.2. Статическая сложность булевых формул

Под статическими критериями сложности булевой формулы будем понимать такие, которые не используют для своего вычисления статистические параметры входящих в формулу букв.

Простейшим статическим критерием сложности булевой формулы является число букв в ней. Другим известным критерием статической сложности булевой формулы являются полное число M двоичных наборов длины n, воспроизводящих таблицу истинности, реализуемую булевой формулой, причем $M = 2^n$, где n - число аргументов булевой формулы. Более полным критерием является объем Q таблицы истинности, определяемый как

(2.4) $$Q = nM = n2^n.$$

Таблица истинности показывает полный набор тестов, проверяющих программу, реализующую булеву формулу.

Однако существует меньший по объему минимальный набор проверяющих тестов – ортогональная форма (ОДНФ и ОДНФ' – см. выше, раздел 2.1.3.). Число конъюнкций в этих двух формах представления булевой функции соответствует <u>числу маршрутов (путей)</u> в линейном бинарном графе, программно реализующем заданную булеву формулу, так как это есть минимальный тест, проверяющий программу [196]. При этом каждая конъюнкция ОДНФ показывает единичный путь в линейном бинарном графе, а в ОДНФ' – нулевой путь. В [176] показано, что перечисление всех путей линейного бинарного графа является покрытием полной таблицы истинности.

2.2.1. Вычисление числа путей в линейном бинарном графе (ЛБГ)

Подсчет числа S путей без их непосредственного получения в виде ОДНФ и ОДНФ' осуществляется различными способами. Известный метод Флойда [250] состоит в возведении в степень матрицы смежности графа, что является весьма трудоемким. Другим известным алгоритмом является алгоритм Дейкстры [250]. Но и этот алгоритм, не учитывающий специфику бинарных программ и, тем более, ЛБГ – также трудоемок. Поэтому в [176] предложены два более эффективных алгоритма подсчета путей в ЛБГ без их непосредственного построения. Первый из них базируется на известном подходе Акерса [290], а второй – основан на поэтапном подсчете числа путей, начиная с выходов ЛБГ. Данный подход обладает тем недостатком, что подсчитывается только общее число путей в ЛБГ без разделения на число единичных и нулевых путей. Предложим новый простой алгоритм, устраняющий данный недостаток.

Алгоритм подсчета числа путей в ЛБГ.

1) Единичный выход ЛБГ пометим дробью 1/0, нулевой – 0/1.

2) Полагаем i = h.

3) i-ю вершину помечаем дробью a/b, где a=a0+a1, b=b0+b1, a1 – числитель дроби, помечающей единичную дугу, исходящую из данной вершины, a0 – числитель дроби, помечающий нулевую дугу, исходящую из данной вершины, b1 и b0 – знаменатели указанных дробей, соответственно.

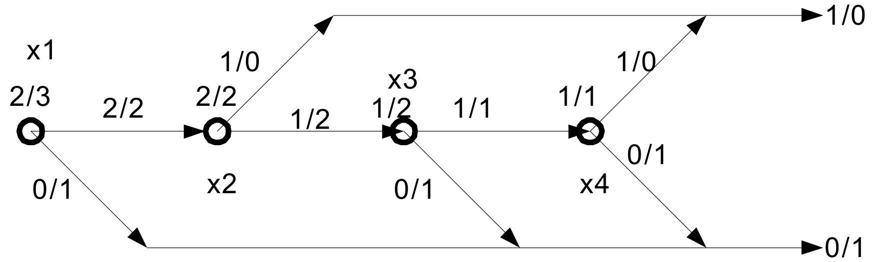
4) Если i > 1, то каждую дугу, заходящую в i-ю вершину помечаем полученной в предыдущем пункте дробью a/b и переходим к п.5, иначе – к п. 6.

5) i = i – 1. Переходим к п.3.

6) Дробь, помечающая первую вершину, отображает число единичных (числитель) и нулевых (знаменатель) путей, сложив эти числа, получим общее число путей в ЛБГ.

7) Конец алгоритма.

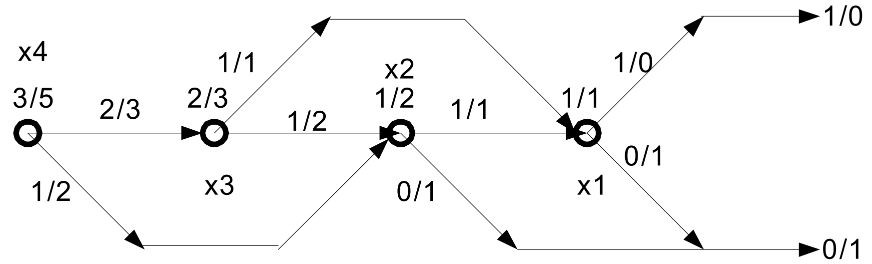
Пример подсчета числа путей по данному алгоритму представлен на рис. 2.3. для тех же ЛБГ, что и на рис.2.1.



y = x1(x2 v x3 x4)

S1=2, S0 = 3, S = 5

а)



y = (x4 x3 v x2) x1

S1 = 3, S0 = 5, S = 8

б)

Рис. 2.3

Для случая, когда буквам булевой формулы соответствуют вложенные булевы формулы (формуле соответствует блочный ЛБГ [274]), предложим модификацию выше приведенного алгоритма. При этом полагаем, что каждой букве формулы (вершине ЛБГ) соответствует дробь d1/d0, задающая число единичных и нулевых путей, соответственно, в ЛБГ, реализующем соответствующую подформулу заданной формулы.

Алгоритм подсчета числа путей в блочном ЛБГ.

1) Единичный выход ЛБГ пометим дробью 1/0, нулевой – 0/1.

2) Полагаем i = h.

3) i-ю вершину помечаем дробью a/b, где a=(a0+a1)d1, b=(b0+b1)d0, a1 – числитель дроби, помечающей единичную дугу, исходящую из данной вершины, a0 – числитель дроби, помечающий нулевую дугу, исходящую из данной вершины, b1 и b0 – знаменатели указанных дробей, соответственно; d1 и d0 – числитель и знаменатель дроби, соответствующей вложенному в i-ю вершину ЛБГ.

4) Если i > 1, то каждую дугу, заходящую в i-ю вершину помечаем полученной в предыдущем пункте дробью a/b и переходим к п.5, иначе – к п. 6.

5) i = i – 1. Переходим к п.3.

6) Дробь, помечающая первую вершину, отображает число единичных (числитель) и нулевых (знаменатель) путей, сложив эти числа, получим общее число путец в ЛБГ.

Конец алгоритма.

2.2.2. Оценки числа путей в ЛБГ

Рассмотрим полученные автором оценки числа путей в ЛБГ, реализующих различные классы булевых формул [177].

Для элементарной дизъюнкции число S1 единичных, нулевых S0 путей и общее число S путей оцениваются однозначно :

(2.5,а) S1 = h,

(2.5,б) S0 = 1,

(2.5,в) S = h + 1.

Для элементарной конъюнкции значения S1, S0 и S также оцениваются однозначно :

(2.6,а) $S1 = 1$,

(2.6,б) $S0 = h$,

(2.6,в) $S = h + 1$.

Для знакопеременных пороговых формул, записанных в виде $x_1(x_2 \vee x_3(x_4 \vee x_5(x_6 \vee \ldots(x_{h-1} \vee x_h)\ldots))))$ – (пример на рис. 2.1,а) и подобных - ЛБГ имеет вид «ели». Значения S1, S0, S оцениваются однозначно:

(2.7,а) $S1\ (S0) = \left\lfloor \frac{h}{2} \right\rfloor + 1$,

(2.7,б) $S0\ (S1) = \left\lfloor \frac{h}{2} \right\rfloor + 1$ – при нечетном h,

$= \frac{h}{2}$ - при четном h.

(2.7,в) $S = h + 1$,

где выражение в скобках $\lfloor a \rfloor$ означает целую часть числа a.

Для тех же формул, записанных в обратном порядке, имеют место другие оценки. Запись таких формул имеет следующий вид:

(2.8,а) $(\ldots((x_1 \vee x_2)x_3 \vee x_4)x_5 \vee \ldots \vee x_{h-1})x_h$,

(2.8,б) $(\ldots((x_1 \vee x_2)x_3 \vee x_4)x_5 \vee \ldots \vee x_{h-2})x_{h-1} \vee x_h$,

(2.8,в) $(\ldots((x_1 x_2 \vee x_3)x_4 \vee x_5)x_6 \vee \ldots \vee x_{h-1})x_h$,

(2.8,г) $(\ldots((x_1 x_2 \vee x_3)x_4 \vee x_5)x_6 \vee \ldots \vee x_{h-2})x_{h-1} \vee x_h$.

ЛБГ для таких формул обладают той особенностью, что все переходы имеют длину, тождественно равную двум. Получим значение числа единичных S1 путей в ЛБГ для формул вида (2.8,б, 2.8,г) или нулевых S0 путей в ЛБГ для формул (2.8,а, 2.8,в). Другими словами, получим значение $S_{max}(h) = \max(S1,S0)$ для ЛБГ, реализующих формулы вида (2.8) из h букв.

В обеспечение вывода введем понятие i-подграфа ЛБГ, под которым будем понимать ЛБГ, полученный удалением всех вершин исходного ЛБГ, размещенных левее i-й вершины, и удалением всех дуг, исходящих из удаленных вершин. Обозначим $S^{i}_{\max}$ - значение S_{max} для i-подграфа. Для рассматриваемых ЛБГ имеем

(2.9,а) $S^{h}_{\max} = 1$

(2.9,б) $S^{h-1}_{\max} = 2$

(2.9,в) $S^{i}_{\max} = S^{i+1}_{\max} + S^{i+2}_{\max}$ при $i < h-1$

Заменим $S^{i}_{\max}$ на эквивалентное обозначение $S_{max}(h- i + 1)$. Тогда выражение (2.9,в) принимает вид

$S_{max}(h- i + 1) = S_{max}(h- i) + S_{max}(h- i - 1)$.

Полагая i = 1, получим

(2.10) $S_{max}(h) = S_{max}(h - 1) + S_{max}(h - 2)$.

Выражение (2.10) представляет собой определение числа Фибоначчи [73]. Учитывая, что ряд Фибоначчи имеет вид

$\Phi = \{\Phi_1,\Phi_2,\Phi_3,\Phi_4,\Phi_5,\Phi_6,\ldots\} = \{1,1,2,3,5,8,\ldots\}$,

а также соотношения (2.9,а) и (2.9,б), получим

(2.11) $S_{max}(h) = \Phi_{h+1}$.

Таким образом, для знакопеременных пороговых формул, учитывая соотношения (2.8), справедливо соотношение

(2.11) $\left\lfloor \frac{h}{2} \right\rfloor + 1 \le S_{max}(h) \le \Phi_{h+1}$, Выражение (2.11) применимо и для произвольной булевой формулы.

Автором в работе [177] выведены оценки для общего числа S(h) путей в ЛБГ, реализующих произвольные булевы формулы из h букв:

(2.12) $$h + 1 \le S(h) \le \Phi_{h+2} \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2} -\right)^{h+2}.$$

При этом верхняя оценка достижима для формул вида (2.8), а нижняя – для обратной перестановки этих же формул и для элементарных конъюнкций и дизъюнкций.

Отметим, что для формул вида (2.8) число S1 единичных и S0 нулевых путей, а также общее число S путей в ЛБГ, исходя из правой части двойных неравенств (2.11) и (2.12), связаны следующим соотношением:

(2.13) если $S1\ (S0) = S_{max}(h) = \Phi_{h+1}$,

то $S0\ (S1) = S(h) - S_{max}(h) = \Phi_{h+2} - \Phi_{h+1} = \Phi_h$

Для примера, изображенного на рис. 2.1,а имеем: число единичных путей S1 = 2, число нулевых путей S0 = 3, общее число путей S(4) = 5, что согласуется с нижней оценкой (2.12) . Для той же булевой формулы, но записанной наоборот, (рис. 2.1,б) получим: число единичных путей $S1 = 3 = \Phi_4$, число нулевых путей $S0 = 5 = \Phi_5$, общее число путей $S(4) = 8 = \Phi_6$, что согласуется с верхней оценкой (2.12).

Укажем, что верхняя оценка для числа путей, то есть наименьшего множества тестов для проверки программы, реализующей булеву формулу бинарной программой, описываемой ЛБГ, меньше числа двоичных наборов из h букв

(2.14) $$S(h) \le \Phi_{h+2} < 2^h.$$

2.2.3. Суммарная длина путей в линейном бинарном графе

Другим критерием статической сложности булевой формулы является суммарная длина L путей в соответствующем ЛБГ. Значению L соответствует также общее число букв в ортогональной форме (ОДНФ и ОДНФ' — см. раздел 2.1.3.), то есть L определяет объем наименьшей тестовой последовательности.

В работе [177] получены следующие оценки для суммарной длины L(h) путей в ЛБГ, реализующих произвольные булевы формулы:

$$\frac{1}{2} h(h + 3) \le L(h) \le 2h\Phi_h. \tag{2.15}$$

Как следует из последнего соотношения верхняя оценка для суммарной длины путей также определяется через числа Фибоначчи. При этом как нижние, так и верхние оценки числа и суммарной длины путей достижимы для различных перестановок булевых формул вида (2.8).

Еще одним критерием статической сложности булевых формул будем считать производную величину от суммарной длины путей и числа путей в ЛБГ, а именно – среднюю длину $l_{ср}(h)$ пути

$$\frac{h(h+3)}{2(h+1)} \le l_{ср}(h) = \frac{L(h)}{S(h)} \le \frac{2h\Phi_h}{\Phi_{h+2}}. \tag{2.16}$$

Отметим, что минимальная длина l_{mih} пути в произвольном ЛБГ определяется соотношением

$$(2.17) \qquad 1 \le l_{mih} \le \left\lfloor \frac{h+1}{2} \right\rfloor.$$

Представим также верхнюю оценку числа s_l^h путей заданной длины l в ЛБГ из h вершин, реализующих формулы (2.8):

$$(2.18) \qquad s_l^h = 2b_l^h + b_l^{h-1},$$

где b_l^h - коэффициенты из треугольника Паскаля [97,264], изображенного в виде следующей матрицы:

l \ h	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0	0
3	0	0	1	2	1	0	0	0	0	0
4	0	0	0	1	3	3	1	0	0	0
5	0	0	0	0	1	4	6	4	1	0
6	0	0	0	0	0	1	5	10	10	5

При этом $b_1^0 = 0$.

Для ЛБГ (рис. 2.1,б) получим: $s_4^4 = 2b_4^4 + b_4^3 = 2\cdot1+0 = 2$, $s_3^4 = 2b_3^4 + b_3^3 = 2\cdot2+1 = 5$, $s_2^4 = 2b_2^4 + b_2^3 = 2\cdot0+1 = 1$.

2.2.4. Инвариантные булевы формулы

Дополнительно представим результаты по анализу числа путей в так называемых инвариантных булевых формулах, линейный бинарный граф которых не изменяет свою структуру (за исключением пометок буквами вершин) при произвольной перестановке членов формулы. Вначале рассмотрим формулу из h букв, представляющую собой дизъюнкцию произвольных подформул из k букв, линейный бинарный граф каждой из которых содержит p≥1 единичных и q≥2 нулевых путей. Для такой формулы имеет место

$$(2.19,а) \qquad S0 = q^{\frac{h}{k}},$$

$$(2.19,б) \qquad S1 = p\,\frac{q^{\frac{h}{k}}-1}{q-1},$$

$$(2.19,в) \qquad S = \frac{q^{\frac{h}{k}}(p+q-1)-p}{q-1}.$$

Рассмотрим инвариантную дизъюнктивную нормальную форму (ДНФ), где p = 1, q = k ≥ 2. Для такой ДНФ имеют место следующие показатели числа путей:

$$(2.20,а) \qquad S0 = k^{\frac{h}{k}},$$

$$(2.20,б) \qquad S1 = \frac{k^{\frac{h}{k}}-1}{k-1},$$

$$(2.20,в) \qquad S = \frac{k^{\frac{h}{k}+1}-1}{k-1} < \frac{k}{k-1}k^{\frac{h}{k}} < 2k^{\frac{h}{k}} < 2e^{\frac{h}{e}},$$

где e – основание натурального логарифма.

Самый правый член неравенства (2.20,в) [177] показывает, что максимальное число путей в линейном бинарном графе соответствует ДНФ с идеальной длиной конъюнкций, равной e, что практически определяется конъюнкциями длины 2 и 3. Приведем формулы для подсчета числа путей в линейном бинарном графе применительно к ДНФ, в которых $h = 2\alpha_2 + 3\alpha_3$, где α_2 и α_3 – число конъюнкций с числом букв, соответственно два и три:

(2.21) $S = 2^{\frac{h}{2}+1} - 1$ при $h = 2r$, $r = 1,2,\ldots,8$;

$S = \frac{1}{2}\left(3^{\frac{h}{3}+1} - 1\right)$ при $h \neq 6,12$, $\mathrm{mod}_3(h) = 0$;

$S = 3^{\frac{h+1}{3}}$ при $h \neq 2,8,14$, $\mathrm{mod}_3(h) = 2$;

$S = 2\cdot 3^{\frac{h-1}{3}}$ при $h \neq 4,10,16$, $\mathrm{mod}_3(h) = 1$.

Докажем **теорему**: для произвольных инвариантных булевых формул из h букв справедливо соотношение:

(2.22) $S \leq 2e^{\frac{h}{e}}$.

Каждая из рассматриваемых инвариантных булевых формул соответствует многоярусной логической схеме, обладающей следующими свойствами:

- каждая дизъюнкция (конъюнкция) формулы реализуется одним элементом ИЛИ (И);

- все входные переменные подключены только ко входам первого яруса;

- каждый вход элемента i-го ($i \geq 2$) яруса подключен к выходу соответствующего элемента ($i - 1$)-го яруса;

- все элементы одного яруса одинаковы;

- элементы соседних ярусов разнотипны (либо ИЛИ, либо И).

Если в схеме указанного класса первый ярус состоит из элементов И (ИЛИ), то $S_{max} = S0$ ($S_{max} = S1$).

Если элемент первого яруса содержит k входов, и число элементов этого яруса равно $m = h / k$, то

$$(2.23) \qquad S_{max} \leq (S^1_{max})^m = k^{\frac{h}{k}},$$

где S^1_{max} - наибольшее из числа единичных и нулевых путей подформулы, соответствующей элементу первого яруса.

Так как для любых положительных чисел a и b, где $a \geq b$, выполняется соотношение $a + b \leq 2a$, то с учетом (2.23) получим

$$(2.24) \qquad S = S0 + S1 \leq 2S_{max} = 2k^{\frac{h}{k}}.$$

Для определения максимального значения S вычислим значение k, при котором производная правой части выражения (2.24) равна нулю

$$(2.25) \qquad (2k^{\frac{h}{k}})' = 2h(1 - \ln(k))\, k^{\frac{h}{k-2}} = 0.$$

Решением этого уравнения является $k = e$ – основанию натурального логарифма. Заменяя k на e в (2.24) получим доказываемое в теореме соотношение.

Данная теорема имеет другое прикладное значение, а именно: при реализации нейросемантических структур [54] оптимальным значением числа букв в слоеобразующем элементе равна е.

2.2.5. Нечеткий критерий сложности булевых формул

Предложим еще один критерий Н статической сложности, названный нечетким. В его основу закладывается рассмотренное в разделе 2.1.4 нечеткое множество переходов.

Рассмотрим общий случай функции принадлежности $|\mu_A(X)|$ при произвольном h. Эта функция размещается в области, ограниченной прямой $\mu = 0$ и прямой $\mu_д = 1$. Последняя представляет собой функцию принадлежности нечеткого множества переходов элементарной дизъюнкции из h букв. Как было указано выше, линейный бинарный граф элементарной дизъюнкции имеет наименьшее число путей - имеет наименьшую сложность.

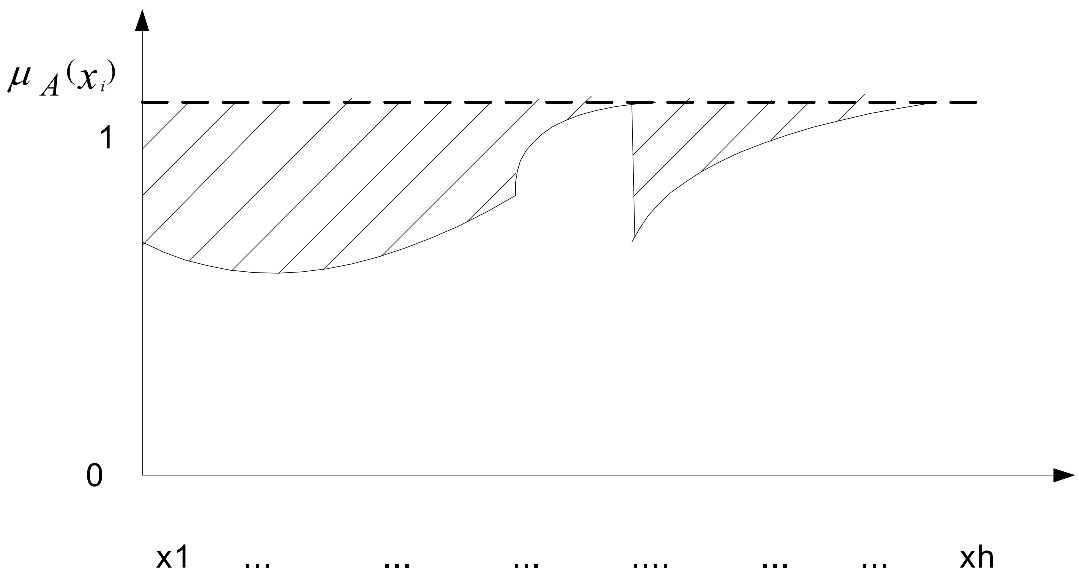


Рис. 2. 4

На рис. 2.4. область, ограниченная прямой $\mu_д$ и кривой $\mu_A(X)$, (заштрихована) показывает отклонение от идеала - характеризует сложность булевой формулы. В частности, если $\mu_A(X) \approx 0$, то булевой формуле соответствует наивысшая сложность. Заштрихованная область на рис. 2.4. нечетко характеризует сложность формулы. Соответствующий критерий H предложим вычислять на основе следующего соотношения:

(2.26)
$$H = \frac{2\sum_{i=1}^{h}\left(h-i+1\right)(1-\left|\ \mu_A(x_i)\ \right|)}{h^2}.$$

Значение этого критерия меняется в диапазоне $0 \le H < 1$. Так как одно и то же значение H могут иметь место для различных формул, в том числе при различных h, то следует ввести лингвистическую переменную вида:

(2.27) “Сложность” = {“Идеал”, “Малая”, “Средняя”, “Большая”, “Очень Большая”}.

Значения этой лингвистической переменной определим треугольными функциями принадлежности [4]:

(2.28)
$$trimf_c(H, a_c, b_c, d_c) = \max\left(\min\left(\frac{H-a_c}{b_c-a_c}, \frac{d_c-H}{d_c-b_c}\right), 0\right),$$

где c – значение лингвистической переменной (7), a, b, d – абсциссы левой, верхней и правой точки треугольной функции принадлежности данной переменной. Положим:

для значения “Идеал”: a = b = d = 0;

для значения “Малая”: a = 0, b = 0.1, d = 0.2;

для значения “Средняя”: a = 0.1, b = 0.25, d = 0.4;

для значения “Большая”: a = 0.35, b = 0.7, d = 0.9;

для значения “Очень Большая”: a = 0.8, b = d = 1.

Использование нечеткого множества переходов и нечеткого критерия позволяет определить относительную сложность булевой формулы для каждого варианта ее записи безотносительно числа букв в формуле.

Рассмотрим вычисление нечеткого критерия сложности формулы, представленной на рис. 2.1,б. Соответствующее этой формуле нечеткое множество B переходов (см. раздел 2.1.4.) имеет вид: $B = \{(x_4 / -0.5), (x_3 / 0.67), (x_2 / -1), (x_1 / 1)\}$. Вычислим значение нечеткого критерия согласно выражению (2.26):

$$H = 2((4-1+1)(1-0.5) + (4-2+1)(1-0.67) + (4-3+1)(1-1) + \\ +(4-4+1)(1-1))/4^2 = 2(2+0.99+0+0)/16 \approx 6/16 = 0.375.$$

Сравнительно малая величина полученного значения критерия свидетельствует о высокой сложности булевой формулы. Соответствующее значение лингвистической переменной (2.27): «Сложность» = «Большая».

Рассмотренные нечеткие множества переходов и нечеткий критерий предложены автором в [312]. Данные результаты рекомендуется использовать в перспективных транспортных средствах и космических аппаратах дальнего и ближнего космоса.

2.3. Оптимизация булевых формул по статическому критерию сложности

Изменение порядка записи булевой формулы путем перестановки ее членов приводит к новым значениям рассмотренных критериев статической сложности, в том числе к изменению положения и/или формы кривой $\mu_A(X)$) (рис.2.4). Задача выбора оптимальной записи булевой формулы очевидно представляется полным перебором всех возможных перестановок ее членов и выбором перестановки, обеспечивающей наименьшее значение числа путей S в соответствующем линейном бинарном графе или нечеткого критерия H согласно (2.26).

Однако, выбрать оптимальный по статическому критерию сложности вариант записи булевой формулы можно не только полным перебором и построением гистограмм, а целенаправленным алгоритмом ее поиска.

В обеспечение такого алгоритма введем понятия веса члена конъюнкции и дизъюнкции.

Вес c_i i-го члена конъюнкции определяется выражением

(2.29) $$c_i = (p_i - 1) / q_i,$$

где p_i и q_i – число единичных и нулевых путей, соответственно, в линейном бинарном графе, реализующем i-й член конъюнкции, i = 1,…,h, h – число букв конъюнкции.

Имеет место следующая лемма:

Если задана конъюнкция h булевых формул и выполняется соотношение

(2.30) $$(p_1 - 1) / q_1 \le (p_2 - 1) / q_2 \le \ldots \le (p_h - 1) / q_h,$$

где i – порядковый номер формулы, входящей в данную конъюнкцию, то справедливо неравенство

(2.31) $$\frac{p_i - 1}{q_i} \le \frac{S1 - 1}{S0} \le \frac{p_h - 1}{q_h},$$

где S1, S0 – число единичных и нулевых путей, соответственно, в линейном бинарном графе рассматриваемой конъюнкции булевых формул.

Доказательство леммы представлено в [178].

Там же на базе утверждения (2.31) и (2.30) доказана следующая теорема:

если справедливо соотношение (2.30), то оптимальной записью конъюнкции булевых формул является

(2.32) $y = y_1 y_2 \ldots y_h$,

где y_i – i-я по порядку булева формула, входящая в конъюнкцию.

Вес d_i i-го члена дизъюнкции определяется величиной

(2.33) $d_i = (q_i - 1) / p_i$,

где p_i и q_i – число единичных и нулевых путей, соответсвенно, в линейном бинарном графе, реализующем i-й член дизъюнкции.

Имеет место аналогичная предыдущей лемма:

если задана дизъюнкция из h булевых формул и выполняется соотношение

(2.34) $(q_1 - 1) / p_1 \le (q_2 - 1) / p_2 \le \ldots \le (q_h - 1) / p_h$,

где i – порядковый номер формулы, входящей в данную дизъюнкцию, то справедливо неравенство

(2.35) $$\frac{q_i - 1}{p_i} \le \frac{S0 - 1}{S1} \le \frac{q_h - 1}{p_h},$$

где S1, S0 – число единичных и нулевых путей, соответственно, в линейном бинарном графе рассматриваемой дизъюнкции булевых формул.

Доказательство леммы представлено в [178].

Там же на базе утверждения (2.34) и (2.35) доказана следующая теорема:

если справедливо соотношение (2.34), то оптимальной записью дизъюнкции булевых формул является

(2.36) $y = y_1 \vee y_2 \vee \ldots \vee y_h$,

где y_i – i-я по порядку булева формула, входящая в рассматриваемую дизъюнкцию.

Перечислим следствия из приведенных двух теорем.

Следствие 1. Если задана конъюнкция (дизъюнкция) некоторых формул, число единичных (нулевых) путей в соответствующем линейном бинарном графе не зависит от взаимного расположения этих формул, а оптимизация по статическим критериям сложности достигается за счет оптимизации только нулевых (единичных) путей.

Следствие 2. Соотношение (2.30, 2.34) является также критерием для определения минимального числа нулевых (единичных) путей в линейном бинарном графе, реализующем конъюнкции (дизъюнкции).

Следствие 3. Одиночной букве x_i, являющейся членом конъюнкции (дизъюнкции) формул соответствует $p_i = q_i = 1$, и ее вес (2.29, 2.33) равен нулю. Поэтому она должна размещаться первой в формуле.

Число путей в конъюнкции h формул вычисляется согласно выражениям

(2.37,а) $S1 = p_1 p_2 \ldots p_h$,

(2.37,б) $S0 = q_1 + p_1(q_2 + p_2(q_3 + \ldots + p_{h-1} q_h)\ldots))$.

Число путей в дизъюнкции h формул вычисляется согласно выражениям

(2.38,а) $S0 = q_1 q_2 \ldots q_h$,

(2.38,б) $S1 = p_1 + q_1(p_2 + q_2(p_3 + \ldots + q_{h-1} p_h)\ldots))$.

Приведем алгоритм оптимизации булевой формулы по статическому критерию сложности.

1) Булева формула представляется в виде суперпозиции формул до глубины 1 [185] (до элементарных конъюнкций или дизъюнкций). Все получаемые подформулы при этом нумеруются (номер формулы обозначим k).

2) Для формул глубины j, начиная с j = 1, определяется c_k (d_k). В формулах глубины j + 1 выполнить перестановку формул глубины j в порядке неубывания c_k (d_k), а при равенстве весов – в порядке неубывания числа путей в линейных бинарных графах, реализующих эти формулы.

3) Указанные в п.2 действия выполняются при j = j + 1 до тех пор, пока не осуществится перестановка во внешней формуле (с наибольшей глубиной).

Пример. Оптимизировать по числу путей в линейном бинарном графе фрагмент программы, реализующий формулу
$y = x_1x_2 \vee x_3 \vee x_4$, где x_1 и x_2 – признаки отсутствия напряжения, x_3 – сигнал аварии работающей панели, x_4 – сигнал короткого замыкания, а y – сигнал на ввод резервного генератора энергетической установки.

Выполняем приведенный выше алгоритм. Обозначим
$y_5 = x_1x_2$ и $y_6 = y = y_5 \vee x_3 \vee x_4$. Определим веса букв формулы y_5 согласно выражению (2.29): $c_1 = c_2 = 0$. Веса равны, поэтому формулу y_5 оставляем без изменения. Определим веса букв формулы y_6 согласно выражению (2.33): $d_3 = d_4 = 0$, $d_5 = (q_5 - 1)/p_5 =$
$= (2\text{-}1)/1 = 1$. Вес $d_5 > d_3 = d_4$. Поэтому формулу y_6 упорядочим в следующей записи $y_6 = x_3 \vee x_4 \vee y_5$. Окончательно получим искомую оптимальную запись формулы $y = x_3 \vee x_4 \vee x_1x_2$. При этом число

путей в соответствующем линейном бинарном графе уменьшилось с семи до пяти.

2.4. Оптимизация булевых формул по динамическому аддитивному критерию сложности.

В реальном вычислительном процессе управления различными техническими средствами большую роль, например для быстродействия алгоритмов, играет вероятность пребывания входных переменных в состояниях ноль и единица. Кроме того, динамика вычисления булевых формул с учетом вероятности пребывания их аргументов в том или ином состоянии, имеет большое значение в ряде приложений, например в базах знаний.

Поэтому рассмотрим динамические критерии сложности булевых формул, зависимые как от порядка записи, так и от статистических параметров аргументов.

В частности, предлагаются два критерия динамической сложности булевых формул: аддитивный и мультипликативный. Понятия аддитивной и мультипликативной величин представлены, например, в [91].

2.4.1. Понятие аддитивного критерия сложности

Аддитивный критерий динамической сложности в орграфе [41], имеющем начальную и конечные вершины, определяется следующим образом [36]:

$$(2.39) \qquad Q = \sum_{j \in M} \left(P_j t_j\right),$$

где M – множество путей , j – номер пути в орграфе, P_j – вероятность j-го пути, t_j – сумма значений некоторого параметра, характеризующего вершину, например, время вычисления на j-м пути.

Отметим, что оптимизация булевой формулы путем перестановки ее членов по быстродействию, но без учета статистических характеристик букв, рассмотрена в [184].

Соотношение (2.39) применим к линейным бинарным графам. При этом будем учитывать, что i-й букве соответствующей булевой формуле из h букв свойственны:

а) вероятность p_i единичного значения;

б) аддитивный параметр t_{i1}, присущий единичному значению буквы;

в) аддитивный параметр t_{i0}, присущий нулевому значению буквы.

В качестве такого аддитивного параметра можно считать время вычисления единичного и нулевого значений буквы, соответственно.

2.4.2. Определение аддитивного критерия сложности для элементарных дизъюнкций и конъюнкций

Рассмотрим линейный бинарный граф, реализующий дизъюнкцию из h букв (2.36), обладающую статистическими параметрами, перечисленными в п. 2.4.1. При этом аддитивный критерий Q принимает значение

$$Q = pQ_1 + (1 - p)Q_0 = R_1 + (1 - p)Q_0, \tag{2.40}$$

где Q_1 и Q_0 – время вычисления единичного и нулевого значений булевой формулы, соответственно, R_1 – единичная компонента критерия Q, p – вероятность вычисления единичного значения формулы, причем

$$p = 1 - (1 - p_1)(1 - p_2)\ldots(1 - p_h) = \tag{2.41}$$
$$= p_1 + (1 - p_1)p_2 + (1 - p_1)(1 - p_2)p_3 + \ldots$$

$$\ldots + (1-p_1)(1-p_2)\ldots(1-p_{h-1})p_h,$$

(2.42) $Q_0 = t_{10} + t_{20} + \ldots + t_{h0}$.

Величину R_1, будем вычислять следующим образом:

(2.43) $R_1 = p_1 t_{11} + (1-p_1)p_2(t_{10} + t_{21}) + \ldots + (1-p_1)(1-p_2)\ldots$

$$\ldots(1-p_{h-1})p_h(t_{10} + t_{20} + \ldots + t_{h-1,0} + t_{h1}).$$

При этом

(2.44) $Q_1 = R_1 / p$.

Аналогичным образом представим формулы для вычисления аддитивного критерия Q для элементарной конъюнкции из h букв, используя те же обозначения, за исключением R_0 – нулевой компоненты критерия:

(2.45) $Q = pQ_1 + (1-p)Q_0 = pQ_1 + R_0$,

(2.46) $p = p_1 p_2 \ldots p_h$,

(2.47) $Q_1 = t_{11} + t_{21} + \ldots + t_{h1}$,

(2.48) $R_0 = (1-p_1)t_{10} + p_1(1-p_2)(t_{11} + t_{20}) + \ldots$

$$p_1 p_2 \ldots p_{h-1}(1-p_h)(t_{11} + t_{21} + \ldots + t_{h-1,1} + t_{h0}),$$

(2.49) $Q_0 = R_0 / (1-p)$.

Отметим, что при произвольной допустимой перестановке букв в конъюнкции (дизъюнкции) соответственно изменяется линейный бинарный граф, что приведет к изменению результатов вычислений по формулам (2.40) – (2.49). Поэтому введем понятие оптимальной записи булевой формулы, то есть такой ее перестановке, при которой обеспечивается наименьшее значение Q.

2.4.3. Оптимальная запись двухместной дизъюнкции (конъюнкции)

Рассмотрим двухместную дизъюнкцию в записи $y_a = x_1 \vee x_2$ и в другой записи $y_b = x_2 \vee x_1$, которым соответствуют аддитивные критерии Q_a и Q_b при заданных p_1, p_2, t_{11}, t_{10}, t_{21}, t_{20}. Определим условие, при котором $Q_a \le Q_b$, то есть запись y_a является оптимальной записью этой дизъюнкции. Согласно (2.40) – (2.44) имеем:

(2.50,а) $Q_a = p_1t_{11} + (1 - p_1)p_2(t_{10} + t_{21}) + z$,

(2.50,б) $Q_a = p_2t_{21} + (1 - p_2)p_1(t_{20} + t_{11}) + z$,

где

$z = (1 - p_1)(1 - p_2)(t_{10} + t_{20})$.

Получим условие, при котором $Q_a \le Q_b$, подставляя (2.50) в это неравенство:

$$p_1t_{11} + (1 - p_1)p_2(t_{10} + t_{21}) + z \le p_2t_{21} + (1 - p_2)p_1(t_{20} + t_{11}) + z.$$

Вычтем z и раскроем скобки:

$$p_1t_{11} + p_2t_{10} - p_1p_2t_{10} + p_2t_{21} - p_1p_2t_{21} \le$$

$$\le \quad p_2t_{21} + p_1t_{20} + p_1t_{11} - p_2p_1t_{20} - p_2p_1t_{11}.$$

После приведения подобных и деления на $p_1p_2 > 0$, получим

$t_{10} / p_1 - t_{10} - t_{21} \le t_{20} / p_2 - t_{20} - t_{11}$, или

$t_{11} + t_{10}(1 / p_1 - 1) \le t_{21} + t_{20}(1 / p_2 - 1)$.

Обозначим

(2.51) $q_i = (1 - p_i) / p_i$

и получим искомое условие, при котором $Q_a \le Q_b$:

(2.52) $T_1 \le T_2$,

где T_i назовем аддитивным рангом i-й буквы дизъюнкции:

(2.53) $T_i = t_{i1} + q_it_{i0} = t_{i1} + t_{i0}(1 - p_i) / p_i$.

Условие (2.52) доказывает следующую лемму.

Лемма 2.4.1. Запись $y = x_1 \vee x_2$ ($y = x_1x_2$) двухместной дизъюнкции (конъюнкции) является оптимальной (по минимуму Q), если имеет место соотношение $T_1 \le T_2$, где T_1 и T_2 вычислены согласно (2.53) для дизъюнкции ((2.54) – для конъюнкции (см. ниже)).

При этом доказательство леммы для конъюнкции проводится аналогично на основании соотношения (2.45), а аддитивный ранг буквы конъюнкции определяется выражением

(2.54) $T_i = t_{i0} + s_i t_{i1} = t_{i0} + t_{i1} p_i / (1 - p_i)$,

где

(2.55) $s_i = p_i / (1 - p_i) = 1 / q_i$.

Далее обозначения (2.51), (2.53)-(2.55) распространим на произвольное $i = 1,2,\ldots,h$, $h \geq 2$.

2.4.4. Основные теоремы

Введем понятие аддитивного ранга T дизъюнкции и конъюнкции, определяемого, соответственно, выражениями:

(2.56) $T = Q_1 + Q_0(1 - p) / p$,

(2.57) $T = Q_0 + Q_1 p / (1 - p)$,

где Q_0 и Q_1 определяются согласно (2.42), (2.44) - для дизъюнкции, и (2.47), (249) – для конъюнкции, а p – соответственно по (2.41), (2.46).

Теорема 2.4.1. Если h-местная дизъюнкция (конъюнкция) представлена такой записью $y = x_1 \vee x_2 \vee \ldots \vee x_h$ ($y = x_1 x_2 \ldots x_h$), которой соответствует соотношение

(2.58) $T_1 \leq T_2 \leq \ldots \leq T_h$,

то имеет место двойное неравенство вида

(2.59) $T_1 \leq T \leq T_h$.

Доказательство. Пусть выполнены условия теоремы для дизъюнкции. При этом, вероятность p единичного значения булевой формулы можно вычислить следующим образом:

$p = p_1+(1-p_1)(p_2+(1-p_2)(p_3+\ldots+(1-p_{h-1})p_h)\ldots)) =$

$= q_0p_1(1+q_1p_2(1+q_2p_3(1+\ldots+q_{h-1}p_h)\ldots))),$

где $q_0=1$. Обозначив

$$A_i=1+q_ip_{i+1}A_{i+1},\ (i=1,\ldots,h-1;\ A_h=1),$$

получим $p=q_0p_1A_1$. Аддитивный критерий

$$Q = Q_1p+Q_0(1-p)=p(Q_1+qQ_0)=pT,$$

где

$$q = (1-p)/p.$$

Тогда

$$T=Q_1+qQ_0 = Q/p.$$

Теперь аддитивный критерий можно выразить следующим образом:

$Q=p_1t_{11}+(1-p_1)(p_2(t_{10}+t_{21})+(1-p_2)(p_3(t_{10}+t_{20}+t_{31})+\ldots+$

$+(1-p_{h-1})(p_h(t_{10}+t_{20}+\ldots+t_{h-1,0}+t_{h1})+(1-p_h)(t_{10}+t_{20}+\ldots+t_{h0})\ldots)))=$

$=p_1t_{11}+(1-p_1)t_{10}+(1-p_1)p_2t_{21}+(1-p_1)(1-p_2)t_{20}+\ldots+$

$+(1-p_1)\ldots(1-p_{h-1})p_ht_{h1}+(1-p_1)\ldots(1-p_h)t_{h0}=$

$=p_1(t_{11}+q_1t_{10}+q_1p_2(t_{21}+q_2t_{20}+q_2p_3(t_{31}+\ldots q_{h-1}p_h(t_{h1}+q_ht_{h0})\ldots)))=$

$=p_1(T_1+q_1p_2(T_2+q_2p_3(T_3+\ldots+q_{h-1}p_hT_h)\ldots)))=q_0p_1B_1,$

где

$$B_i=T_i+q_ip_{i+1}B_{i+1},\quad (I=1,\ldots,h-1;\quad B_h=T_h).$$

Тогда

$$T=Q/p=q_0p_1B_1/(q_0p_1A_1)=B_1/A_1.$$

Докажем теперь, что $T_1 \le T = B_1/A_1$. Так как $A_1>0$, то докажем, что $A_1T_1 \le B_1$, то есть $B_1-A_1T_1 \ge 0$. Выполним подстановку A_1 и B_1 и докажем, что

$T_1+q_1p_2B_2-(1+q_1p_2A_2)T_1\geq 0$,

откуда следует необходимость доказать соотношение

$q_1p_2B_2-q_1p_2A_2T_1\geq 0$,

а так как $q_1p_2>0$, то необходимо доказать соотношение

$B_2-A_2T_1 \geq 0$.

Поступая с последним неравенством аналогично приведенному выше процессу сужения доказываемых соотношений, придем к необходимости доказать, что $B_3-A_3T_1\geq 0$, и так далее, до тех пор, пока не потребуется доказать неравенство

$B_h - A_hT_1 \geq 0$.

После соответствующей подстановки B_h и A_h получим $T_h - T_1 \geq 0$, что истинно, так как $T_1 \leq T_h$ по условию теоремы, и тем самым доказана левая часть доказываемого неравенства $T_1 \leq T$.

Перейдем к доказательству правой части двойного неравенства $T \leq T_h$, или после подстановки $B_1/A_1\leq T_h$, или $T_hA_1-B_1\geq 0$. Раскрыв компоненты последнего неравенства, придем к необходимости доказать следующее неравенство:

$T_h(1+q_1p_2A_2) – (T_1+q_1p_2B_2) \geq 0$.

Раскрыв скобки в последнем неравенстве, докажем следующее неравенство:

$T_h+T_hq_1p_2A_2 – T_1 - q_1p_2B_2 \geq 0$.

Усилим последнее неравенство, исключая $T_h - T_1 \geq 0$, и поделив оставшиеся члены на $q_1p_2 > 0$. После этого докажем, что $T_hA_h – B_2 \geq 0$. Действуя аналогично, придем к необходимости доказать $T_hA_h – B_h \geq 0$. После подстановки $A_h = 1$ и $B_h = T_h$ придем к тождеству $T_h=T_h$, то есть полученный результат цепочки представленного вывода доказывает исходное неравенство.

Таким образом, доказана теорема с выводом (2.59) для дизъюнкции. Доказательство этой теоремы для конъюнкции аналогично, поэтому не приводится.

Теорема 2.4.2. Запись h-местной дизъюнкции (конъюнкции) $y = x_1 \vee x_2 \vee \ldots \vee x_h$ ($y = x_1x_2\ldots x_h$), является оптимальной по аддитивному критерию, если имеет место соотношение (2.58)
($T_1 \le T_2 \le \ldots \le T_h$).

Доказательство выполним по индукции.

Пусть доказано, что фрагмент рассматриваемой булевой формулы вида $y_i = x_1 \vee x_2 \vee \ldots \vee x_i$ есть оптимальная запись для $2 \le i < h$. Докажем, что другой фрагмент рассматриваемой дизъюнкции вида $y_{i+1} = y_i \vee x_{i+1}$, также есть оптимальная запись для того же значения i. Исходя из теоремы 2.4.1, аддитивный ранг T фрагмента y_i не больше ранга T_{i+1} буквы x_{i+1}, поэтому приведенная запись y_{i+1} не противоречит лемме, если y_i рассматривать как букву дизъюнкции. Запись вида $y_{i+1} = x_{i+1} \vee y_i$ противоречит лемме и не допустима. Покажем, что букву x_{i+1} нельзя вставить внутри фрагмента y_i. Представим y_i дизъюнкцией $y_i = y' \vee y''$, где $y' = x_1 \vee x_2 \vee \ldots \vee x_k$, $y'' = x_{k+1} \vee \ldots \vee x_i$, где $1 \le k < i$. Образуем дизъюнкцию вида $y' \vee x_{i+1} \vee y''$. При этом, хотя фрагмент вида $y' \vee x_{i+1}$ является его оптимальной записью, но зато фрагмент $x_{i+1} \vee y''$ не является оптимальной записью. При этом значение аддитивного критерия последнего фрагмента больше значения этого критерия для фрагмента $y'' \vee x_{i+1}$. Поэтому рассматриваемый фрагмент $y' \vee (x_{i+1} \vee y'')$ обладает большим значением критерия по сравнению с записью $y' \vee y'' \vee x_{i+1}$. Следовательно, запись вида $y_{i+1} = y_i \vee x_{i+1}$ есть единственный вариант оптимальной записи данного фрагмента.

Доказательство для конъюнкции идентично и поэтому не рассматривается.

2.4.5. Оптимизация скобочных булевых формул по аддитивному критерию сложности

Представим алгоритм оптимизации булевой формулы по аддитивному критерию сложности.

1) Булева формула представляется в виде суперпозиции формул до глубины 1 [185] (до элементарных конъюнкций или дизъюнкций). Все получаемые подформулы при этом нумеруются (номер подформулы обозначим k).

2) Для подформул глубины j, начиная с j = 1, определяется вероятность единичного значения, аддитивные параметры единичного и нулевого значений и аддитивный ранг каждой буквы (подформулы глубины j – 1). Во всех таких подформулах выполняется перестановка букв (подформул глубины j-1) в порядке неубывания аддитивного ранга.

3) Указанные в п.2 действия выполняются при j = j + 1 до тех пор, пока не осуществится перестановка во внешней формуле (с наибольшей глубиной).

Как указывалось выше, аддитивным критерием сложности может служить среднестатистическое время вычисления значения булевой формулы. При этом аддитивные параметры буквы (величины t_{i1} и t_{i0}) должны указывать длительность вычисления, соответственно, единичного и нулевого значения i-й буквы формулы.

Оптимизация булевых формул по быстродействию является важным этапом в синтезе алгоритмов программного управления различными техническими средствами. Более подробно данный вопрос изложен в работе автора [152].

Пример. Оптимизировать по минимуму длительности вычислений формулу $y = x_1 \vee x_2 \vee x_3$, где x_1 – признак готовности измерения значения тока перемычки (привилегированного измеряемого в резервном приборе параметра), x_2 – признак готовности результатов тестирования оборудования космической станции, x_3 – данная станция выбрана в качестве основной (не в резерве), y - разрешение выдачи данных оператору системы управления установкой (представлен фрагмент формулы). При этом тестирование производится, например, один раз в 10 сек, измерение тока перемычки – один раз в 2 сек, выбирается произвольно в качестве основной одна из двух станций, выдача данных оператору выполняется три раза в секунду. На основе изложенного представим статистические параметры аргументов формулы y:

$p_1 = 0.3 / 2 = 0.15$; $p_2 = 0.3 / 10 = 0.03$; $p_3 = 0.5$;

$t_{11} = t_{10} = 0.4$ мксек; $t_{21} = t_{20} = 0.5$ мксек; $t_{31} = t_{30} = 0.1$ мксек.

Вычислим значения аддитивного ранга каждой буквы дизъюнкции y согласно выражению (2.53):

$T_1 = t_{11} + t_{10} (1-p_1) / p_1 = 0.4+0.4(1-0.15)/0.15 = 3.8$;

$T_2 = t_{21} + t_{20} (1-p_2) / p_2 = 0.5+0.5(1-0.03)/0.03 = 16.67$;

$T_3 = t_{31} + t_{30} (1-p_3) / p_3 = 0.1+0.1(1-0.5)/0.5 = 0.2$.

В соответствии с теоремой 2.4.2 оптимально записью исследуемой формулы будет запись $y = x_3 \vee x_1 \vee x_2$, так как $T_3 < T_1 < T_2$.

Определим длительность вычисления значения данной формулы до и после оптимизации. Для исходной формулы

$y = x_1 \vee x_2 \vee x_3$ согласно выражению (2.39) имеем:

$Q = p_1t_{11}+(1-p_1)p_2(t_{10}+t_{21})+(1-p_1)(1-p_2)p_3(t_{10}+t_{20}+t_{31})+$

$+(1-p_1)(1-p_2)(1-p_3)(t_{10}+t_{20}+t_{30})=0.15\cdot0.4+(1-0.15)\cdot0.03(0.4+0.5)+$
$+(1-0.15)(1-0.03)\cdot0.5(0.4+0.5+0.1)+ (1-0.15)(1-0.03)(1-0.5)(0.4+$
$+0.5+0.1)=0.90.$

Для оптимизированной формулы $y = x_3 \vee x_1 \vee x_2$ аналогично получим:

$Q = p_3t_{31}+(1-p_3)p_1(t_{30}+t_{11})+(1-p_3)(1-p_1)p_2(t_{30}+t_{10}+t_{21})+$
$+(1-p_3)(1-p_1)(1-p_2)(t_{30}+t_{10}+t_{20})=0.5\cdot0.1+(1-0.5)\cdot0.15\cdot(0.1+0.4)+$
$+(1-0.5)(1-0.15)\cdot0.03\cdot(0.1+0.4+0.5)+(1-0.5)(1-0.15)(1-0.03)(0.1+$
$+0.4+0.5)=0.51.$

Таким образом, длительность вычисления булевой формулы снизилась почти вдвое только лишь за счет перестановки одной буквы в ней. Рассмотрим также наихудший случай, то есть запись формулы в порядке, обратному оптимальной записи этой формулы, а именно: $y = x_2 \vee x_1 \vee x_3$. Получим для этой формулы среднестатистическую длительность вычислений:

$Q = p_2t_{21}+(1-p_2)p_1(t_{20}+t_{11})+(1-p_2)(1-p_1)p_3(t_{20}+t_{10}+t_{31})+$
$+(1-p_2)(1-p_1)(1-p_3)(t_{20}+t_{10}+t_{30})=0.03\cdot0.5+(1-0.03)\cdot0.15\cdot(0.5+0.4)+$
$+(1-0.03)(1-0.15)\cdot0.5\cdot(0.5+0.4+0.1)+(1-0.03)(1-0.15)(1-0.5)(0.5+$
$+0.4+0.1)=0.97.$

По сравнению с оптимальной записью в этом случае проигрыш вдвое по длительности вычислений еще более заметен.

Таким образом, можно сделать вывод о необходимости применять предложенный метод оптимизации булевых формул по длительности вычислений. Практическое использование метода дает снижение длительности вычислений булевых формул в полтора – пять раз.

2.5. Оптимизация булевых формул по динамическому мультипликативному критерию сложности

При программной реализации логического управления решаются проблемы надежности как аппаратуры, так и программ [18]. При этом важным аспектом надежности является повышение достоверности как управляющих сигналов с выхода ЭВМ, так и, в особенности, рекомендаций (советов) оператору. Вычисление достоверности будем ассоциировать с вычислением мультипликативного критерия сложности булевой формулы.

2.5.1. Мультипликативный критерий

В работе [199] получена формула для вычисления достоверности исполнения некоторого оператора программы, которую примем за основу вычисления мультипликативного критерия:

$$D = \frac{\sum_{i \in M} \prod_{j=1}^{Li} Dij \cdot Pij}{\sum_{i \in M} \prod_{j=1}^{Li} Pij} \qquad (2.60)$$

где M – множество единичных (нулевых) путей в управляющем графе программы, Li – длина i-го пути, j – порядковый номер вершины на i-м пути, Dij –мультипликативный показатель, например, достоверность единичного (нулевого) значения переменной булевой формулы, Pij – вероятность единичного (нулевого) значения этой переменной.

Применительно к булевым формулам и линейным бинарным графам мультипликативный критерий D будем вычислять следующим образом:

(2.61) $D = D_1 p + D_0(1 - p)$,

где D_1 и D_0 – единичная и нулевая компоненты критерия, соответствующие совокупностям единичных и нулевых путей в линейном бинарном графе.

Соответствующее выражению (2.61) определение достоверности назовем совокупной достоверностью вычислений значения булевой формулы (определение автора; дано в работе [153]).

2.5.2. Мультипликативный критерий для элементарных двухместных дизъюнкций (конъюнкций)

Рассмотрим дизъюнкцию вида $y = x_1 \vee x_2$. Обозначим d_{i1} и d_{i0} – мультипликативные показатели, соответственно, единичного и нулевого значения i-й буквы, p_i – вероятность единичного значения этой буквы. Нулевая компонента D_0 мультипликативного критерия данной дизъюнкции вычисляемая по единственному нулевому пути соответствующего линейного бинарного графа не зависит от перестановки букв:

$D_0 = d_{10}(1\text{-}p_1)d_{20}(1\text{-}p_2) \;/\; (1\text{-}p_1)(1\text{-}p_2) = d_{10}d_{20} = \text{const}.$

Поэтому значение мультипликативного критерия может изменяться за счет его единичной компоненты D_1, зависимой от перестановки букв.

Рассмотрим два варианта записи дизъюнкции. Выше приведенную запись условимся считать вариантом «a», а запись вида $y = x_2 \vee x_1$ – вариантом «b». Соответственно единичные компоненты мультипликативного критерия обозначим D_{a1} и D_{b1}. Последние вычисляются по совокупности единичных путей в соответствующих ЛБГ следующим образом:

$D_{a1} = (d_{11}p_1+d_{10}(1-p_1)d_{21}p_2) / (p_1+(1-p_1)p_2) = (a_1+b_1a_2)/(p_1+p_2-p_1p_2)$,

$D_{b1} = (d_{21}p_2+d_{20}(1-p_2)d_{11}p_1) / (p_2+(1-p_2)p_1) = (a_2+b_2a_1)/(p_1+p_2-p_1p_2)$,

где $a_i = d_{i1}p_i$, $b_i = d_{i0}(1 - p_i)$, $i = 1,2$.

Найдем условие, при котором $D_{a1} \geq D_{b1}$. Положим

$D_{a1} - D_{b1} = (a_1+b_1a_2- a_2-b_2a_1) / (p_1+p_2-p_1p_2) \geq 0$.

Так как $p_1,p_2 < 1$, то $p_1 + p_2 - p_1p_2 > 0$, поэтому рассмотрим неравенство $a_1+b_1a_2- a_2-b_2a_1 \geq 0$. При этом $a_1(1-b_2) \geq a_2(1-b_2)$. Так как $0 < b_i < 1$, то $1 - b_i > 0$. Следовательно, условие для выполнения $D_{a1} \geq D_{b1}$ будет иметь следующий вид: $a_1/(1-b_1) \geq a_2/(1-b_2)$. Величину

(2.62) $r_i = a_i/(1-b_i) = d_{i1}p_i/(1- d_{i0}(1 - p_i))$

назовем мультипликативным рангом i-й буквы дизъюнкции.

Повторив аналогичные приведенным рассуждения для элементарной двухместной конъюнкции в вариантах записи вида а) $y = x_1x_2$ и b) $y = x_2x_1$, а также учитывая, что перестановка влияет только на нулевую компоненту D, получим, что $D_{a0} \geq D_{b0}$ в том случае, когда выполняется неравенство вида

$b_1/(1-a_1) \geq b_2/(1-a_2)$. При этом величину

(2.63) $r_i = b_i/(1-a_i) = d_{i0}(1 - p_i)/ d_{i1}p_i$

назовем мультипликативным рангом i-й буквы конъюнкции.

Таким образом, доказана следующая лемма.

<u>Лемма 2.5.1</u>. Максимальное значение мультипликативного критерия сложности элементарной двухместной дизъюнкции (конъюнкции) достигается такой перестановкой ее букв, при которой соответствующие мультипликативные ранги букв не возрастают при просмотре формулы слева направо.

2.5.2. Мультипликативный критерий для элементарных дизъюнкций (конъюнкций) с произвольным числом букв

Введем понятие собственного мультипликативного ранга R элементарной дизъюнкции (конъюнкции), определяемый выражением

(2.64) $R = D_1 p \,/\, (1 - D_0(1 - p))$ – для дизъюнкции,

$R = D_0(1 - p) \,/\, (1 - D_1 p)$ – для конъюнкции,

где p – вероятность единичного значения формулы. Докажем следующую лемму.

<u>Лемма 2.5.2</u>. Если элементарная дизъюнкция $y = x_1 \vee \ldots \vee x_h$ (конъюнкция) из h букв упорядочена по не возрастанию мультипликативных рангов букв согласно $r_1 \geq r_2 \geq \ldots \geq r_h$, то имеет место соотношение $r_1 \geq R \geq r_h$.

Доказательство.

Вначале докажем, что $r_1 \geq R$ для $h = 2$:

$r_1 = a_1/(1-b_1) \geq R = D_1 p \,/\, (1 - D_0(1 - p)) = (a_1+b_1a_2)/(1-b_1b_2)$.

Умножив обе части неравенства на оба неотрицательных знаменателя, докажем, что $a_1(1-b_1b_2) \geq (a_1+b_1a_2)(1-b_1)$, или $a_1-a_1b_1b_2 \geq a_1-a_1b_1+a_2b_1(1-b_1)$, или $a_1b_1(1-b_2) \geq a_2b_1(1-b_1)$, или $a_1(1-b_2) \geq a_2(1-b_1)$. Таким образом $r_1 \geq r_2$, что задано по условию. Следовательно, утверждение $r_1 \geq R$ справедливо при $h = 2$.

На основе последнего утверждения для любой двухместной дизъюнкции вида $y_i = x_i \vee y_{i+1}$ $(i = 1,\ldots,h-1)$, рассматриваемой как часть исходной дизъюнкции, всегда справедливо соотношение $r_i \geq R_i$, где R_i – собственный мультипликативный ранг для подформулы y_i, что по индукции приводит к доказываемому утверждению $r_1 \geq R$ для $y = y_1$ при произвольном h.

Теперь докажем, что $R \geq r_h$. Вначале докажем это для дизъюнкции $y' = x_{h-1} \vee x_h$ с учетом $r_{h-1} \geq r_h$:
$R' = (a_{h-1}+b_{h+1}a_h)/(1-b_{h-1}b_h) \geq a_h/(1-b_h)$.
Умножив неравенство на оба знаменателя докажем, что $a_{h-1}(1-b_h)+b_{h-1}a_h-b_{h-1}b_ha_h \geq a_h-b_{h-1}b_ha_h$, или $a_{h-1}(1-b_h) \geq a_h(1-b_{h-1})$, то есть $r_{h-1} \geq r_h$, что доказывает утверждение $R \geq r_h$ для двухместной дизъюнкции. На этом основании для любой двухместной дизъюнкции вида $y_i = y_{i-1} \vee x_i$ $(i=2,\ldots,h)$ как части дизъюнкции y всегда справедливо соотношение $R_i \geq r_i$, где R_i – собственный мультипликативный ранг для y_i, что по индукции приводит к доказываемому утверждению $R \geq r_h$ для $y = y_h$ при произвольном h.

Таким образом, лемма доказана для элементарной дизъюнкции. Доказательство для конъюнкции аналогично, и поэтому не представляется.

Отметим, что мультипликативный критерий, мультипликативный ранг дизъюнкции (конъюнкции) и единичная (нулевая) компонента данного критерия находятся попарно в прямо пропорциональной зависимости.

Теорема 2.5.1. Максимум значения мультипликативного критерия элементарной дизъюнкции (конъюнкции) достигается такой перестановкой ее букв, при рассмотрении которой слева направо соответствующие мультипликативные ранги букв не возрастают.

Доказательство.

Пусть задана дизъюнкция из h букв, и известно, что $r_1 \geq r_2 \geq \ldots \geq r_h$. Доказательство того, что запись (оптимальная по мультипликативному критерию) вида $y = x_1 \vee x_2 \vee \ldots \vee x_h$

обеспечивает максимум значения этого критерия среди всех возможных перестановок букв в y выполним по индукции. Пусть на очередном шаге доказано, что перестановка вида
$y_i = x_1 \vee x_2 \vee \ldots \vee x_i$ обеспечивает максимум мультипликативного критерия для подформулы y_i. При этом собственный мультипликативный ранг R_i такой перестановки не меньше величины r_{i+1} (из леммы 2.5.2). Согласно лемме 2.5.1 значение мультипликативного критерия для перестановки вида $y_{i+1} = y_i \vee x_{i+1}$ выше значения этого критерия для перестановки вида $y_{i+1} = x_{i+1} \vee y_i$. Теперь покажем, что букву x_{i+1} нельзя вставить между буквами подформулы y_i. Представим последнюю как дизъюнкцию вида
$y_i = y_{i1} \vee y_{i2}$, где $y_{i1} = x_1 \vee x_2 \vee \ldots \vee x_j$, $y_{i2} = x_{j+1} \vee x_{j+2} \vee \ldots \vee x_i$, $(1 \le j < i)$. При этом единичная компонента мультипликативного критерия дизъюнкции $y_{i2} \vee x_{i+1}$ выше такой же компоненты дизъюнкции $x_{i+1} \vee y_{i2}$ в силу того, что собственный мультипликативный ранг для y_{i2} не меньше величины r_{i+1}. Поэтому вставка буквы x_{i+1} в середину дизъюнкции y_i недопустима, а перестановка вида $y_{i+1} = x_1 \vee x_2 \vee \ldots \vee x_i \vee x_{i+1}$ обеспечивает максимум значения мультипликативного критерия для дизъюнкции из i+1 букв при $r_1 \ge r_2 \ge \ldots \ge r_h \ge r_{h+1}$ $(i=1,\ldots,h-1)$.

Таким образом, теорема доказана для дизъюнкции. Доказательство для конъюнкции аналогично, и поэтому не приводится.

Из данной теоремы вытекает следующее. Пусть элементарная дизъюнкция (конъюнкция) из h букв упорядочена согласно теореме. Тогда максимум значения мультипликативного критерия формулы с добавленной новой буквой x_{h+1} обеспечивается следующим образом:

а) при $r_{h+1} \geq r_1$ – записью вида $x_{h+1} \vee x_1 \vee x_2 \vee \ldots \vee x_h$;

б) при $r_h \geq r_{h+1}$ – записью вида $x_1 \vee x_2 \vee \ldots \vee x_h \vee x_{h+1}$;

в) при $r_i \geq r_{h+1} \geq r_{i+1}$ – записью вида

$$x_1 \vee x_2 \vee \ldots \vee x_i \vee x_{h+1} \vee x_{i+1} \vee \ldots \vee x_h.$$

2.5.3. Мультипликативный критерий для произвольных скобочных булевых формул

Для произвольной скобочной булевой формулы в базисе И, ИЛИ, НЕ также необходимо искать оптимальную по рассматриваемому критерию перестановку. Предлагается следующий алгоритм поиска такой перестановки.

1) Булева формула приводится к нормальному виду. При этом знак инверсии ставится только над отдельной буквой [19].

2) Нормальная булева формула представляется суперпозицией элементарных дизъюнкций и конъюнкций, обозначаемых каждая новой буквой z_i ($i=h+1,h+2,\ldots$).

3) Для каждой элементарной дизъюнкции (конъюнкции), включающей только независимые переменные, вычисляется мультипликативный ранг каждой буквы и эта дизъюнкция (конъюнкцию) упорядочивается по не возрастанию ранга. При этом, если буква имеет знак инверсии, то для вычисления ее ранга используются выражения $a_i = d_{i0}(1-p_i)$, $b_i = d_{i1}p_i$.

4) Для каждой упорядоченной на предыдущем шаге подформулы, обозначенной z_i, вычисляются p_i, d_{i1}, d_{i0}. Так как эта подформула, в свою очередь, является буквой конъюнкции (дизъюнкции) следующего уровня вложенности, то для нее вычисляется мультипликативный ранг буквы r_i (не путать с собственным мультипликативным рангом подформулы R_i).

5) Для каждой дизъюнкции (конъюнкции) следующего уровня вложенности выполняется упорядочивание входящих в нее букв (подформул) в соответствии с вычисленным для них рангом согласно теореме 2.5.1.

6) Если упорядочена внешняя [185] дизъюнкция (конъюнкция) первого уровня вложенности, то перейти к п.7, иначе – к п.4.

7) Конец работы алгоритма: формула оптимизирована (упорядочена) по мультипликативному критерию.

Прикладное значение оптимизации по мультипликативному критерию сложности булевой формулы состоит, например, в повышении достоверности вычислений только путем перестановок в булевых формулах.

Пример. Для измерения электрических величин электроэнергетической установки используется канал аналоговых измерений, в который включается аналогово-цифровой преобразователь (АЦП) с несколькими входами и нормализатор (устройство для понижения контролируемого напряжения 27В в границы диапазона 0 - 5В, воспринимаемого АЦП). Для проверки исправности этого канала нормализатор формирует постоянное напряжение 2.5В. Для выдачи совета оператору «Проверить предохранитель нормализатора» применим дизъюнкцию вида $y = x_2 \vee x_1$, где x_1 - сигнал отсутствия контрольного напряжения, x_2 - сигнал отсутствия напряжения на рабочих входах АЦП. Вероятность p_1 единичного значения сигнала x_1 определяется исходя из того, что каждую тысячную секунду драйвер АЦП дает единичный сбой и опрос входа контрольного напряжения выполняется в течение 30 сек каждые 10000 сек., а также возможности кратковременного снижения напряжения питания нормализатора и выгорания его предохранителя: $p_1 = 0.95$. Вероятность p_2 единичного значения сигнала x_2 определяется исходя из упомянутого сбоя драйвера, а также фактического отсутствия ненулевых напряжений и токов в контролируемых точках электроэнергетической установки: $p_2 = 0.60$. Достоверность d_{11} единичного значения сигнала x_1 определяется только упомянутым сбоем драйвера и имеет значение, равное 0.997. Достоверность d_{10} нулевого значения x_1 примем за единицу. Достоверность d_{21} единичного значения x_2 определяется сбоем драйвера с учетом того, что измерения выполняются каждые две секунды: $d_{21} = 0.985$. Достоверность d_{20} нулевого значения x_2 из-за возможных помех примем равной 0.99.

В соответствии с теоремой 2.5.1 и по выражению (2.62) вычислим мультипликативный ранг исследуемой формулы
$y = x_2 \vee x_1$:
$r_1 = d_{11}p_1/(1-d_{10}(1-p_1))=0.997\cdot 0.95/(1-1(1-0.95))=0.997;$
$r_2 = d_{21}p_2/(1-d_{20}(1-p_2))=0.985\cdot 0.6/(1-0.99(1-0.6))=0.978.$
Согласно полученному результату, а именно $r_1 > r_2$ формулу следует переписать в виде $y = x_1 \vee x_2$. Подтвердим это расчетами достоверности обоих вариантов согласно выражению (2.60). Для исходного варианта записи формулы получим:

$$D_{21} = p_2d_{21}+(1-p_2)p_1d_{20}d_{11}+(1-p_2)(1-p_1)d_{20}d_{10}=0.6\cdot 0.985+$$

$$+(1-0.6)\cdot 0.95\cdot 0.99\cdot 0.997+(1-0.6)(1-0.95)\cdot 0.99\cdot 1.0=0.986.$$

Для оптимизированного варианта получим:

$$D_{12} = p_1d_{11}+(1-p_1)p_2d_{10}d_{21}+(1-p_1)(1-p_2)d_{10}d_{20}=0.95\cdot 0.997+$$

$$+(1-0.95)\cdot 0.6\cdot 1.0\cdot 0.985+(1-0.95)(1-0.6)\cdot 1.0\cdot 0.997=0.997.$$

Так как $D_{12} > D_{21}$, то приведенный расчет достоверности подтверждает правильность выполненной в примере оптимизации.

2.6. Оптимизация систем булевых формул и продукционных баз знаний по длительности и достоверности вычислений.

В предыдущих разделах было установлено, что перестановка членов булевых формул влияет не только на статическую, но и на динамическую сложность этих формул.

В сложных алгоритмах, как правило, используются множество булевых формул, которые применяются в том числе и для вычисления промежуточных переменных. Для таких систем оптимизация, например по быстродействию (аддитивный критерий) или по достоверности (мультипликативный критерий), начинается с присваивания каждой формуле веса: вес 1 – для булевых формул, включающих только независимые, например, входные переменные; вес 2 – для формул, включающих независимые переменные и буквы, соответствующие результатам вычислений формул с весом 1, и так далее. Оптимизация записей булевых формул выполняется в порядке возрастания их веса, начиная с единичного.

Перспективным направлением развития систем управления, например электроэнергетическими установками межпланетных кораблей, является дополнение традиционных функций контроля и управления новыми функциями на базе теории искусственного интеллекта.

Одним из компонент этой теории служит так называемое логическое программирование на базе языка ПРОЛОГ [215,288, 268]. В основу данного языка положены так называемые продукции, использующие булевы формулы. Однако, логический вывод в программной системе на базе языка ПРОЛОГ выполняется сравнительно долго, что дает основание применить рассмотренный метод оптимизации по аддитивному критерию (по длительности вычислений).

С другой стороны, булевы формулы применяются в различного рода системах искусственного интеллекта, в том числе в экспертных системах [78,205,224,236,12], где явно вычисляется достоверность результатов логического вывода. Однако для повышения достоверности результатов вычислений до сих пор использовались чисто технические приемы [232,110], например дублирование информации. Поэтому, рассмотрим вопрос об оптимизации по мультипликативному критерию (достоверности) для систем искусственного интеллекта предложенным в предыдущем разделе методом.

Таким образом, для продукционных баз знаний требуется оптимизация по динамическим критериям сложности булевых формул. Поэтому предложим общий подход [158] к такой оптимизации. При этом оптимизация возможна либо по аддитивному, либо по мультипликативному критерию.

Предполагаются известными статистические данные: вероятности наличия фактов (факт по терминологии языка ПРОЛОГ означает единичное значение переменной), а также достоверность (длительность вычисления) как истинности (наличия) фактов, так и ложности (отсутствия) фактов.

<u>Оптимизация системы бесповторных утверждений.</u>

Пусть задана база знаний в виде системы утверждений, правые части которых представляют собой дизъюнкции, конъюнкции или дизъюнкции конъюнкций. Группа утверждений с одинаковой головой (левой частью утверждения по терминологии языка ПРОЛОГ) рассматривается как одно утверждение, правая часть которого есть дизъюнкция конъюнкций. Такая база знаний изображается И-ИЛИ графом [57] с одним или несколькими корнями, И-вершинами, ИЛИ-вершинами и листьями (утверждениями, требующими сопоставления с фактами). При этом полагаем, что утверждения в правой части бесповторны. Кроме того, считается возможным выполнить произвольную перестановку членов дизъюнкций и конъюнкций.

Пусть заданы статистические характеристики целей, соответствующих листьям И-ИЛИ графа. Предложим следующий алгоритм оптимизации по динамическому критерию сложности булевой формулы.

1) Отмечаем все листья И-ИЛИ графа.

2) Находим неотмеченную вершину графа, все вершины-приемники которой уже отмечены (отмеченной вершине k соответствует цель, для которой вычислены p_k, t_{k1}, t_{k0} (d_{k1}, d_{k0})).

3) Вычисляем мультипликативный (аддитивный) ранг каждой буквы – цели данного утверждения.

4) Выполняем оптимальную запись этого утверждения путем перечисления его целей согласно соответствующей теореме об оптимальной записи (разделы 2.4.4., 2.5.2).

5) Вычисляем значение вероятности истинности утверждения, достоверности (длительности вычисления) истинного и ложного значений.

6) Отмечаем рассмотренную вершину.

7) Если есть неотмеченные вершины, то п.2, иначе – п.8.

8) Конец работы алгоритма. Записи всех утверждений оптимизированы по заданному критерию.

Оптимизация записи утверждений с повторяющимися целями имеет некоторые особенности, рассмотренные автором в работе [158].

Рассмотрим упрощенный пример. Пусть в базе знаний имеется следующая система утверждений: $y_1 = y_2 \vee y_3$; $y_2 = x_1 \vee x_2$; $y_3 = x_2 \vee x_3$. Здесь обозначены: y_1 – утверждение «Переключи АВХ» (АВХ – активный венероход) при условии y_2 – «Возможен отказ АВХ№1» или при условии y_3 – «Возможен отказ АВХ№2»; y_2 – утверждение «Возможен отказ АВХ№1» при условии x_1 – «индикатор связи с АВХ№1 неподвижен» или при условии x_2 – «информация на экране не обновляется в течении последних нескольких часов»; y_3 – утверждение «Возможен отказ АВХ№2» при условии x_2 или x_3 – «индикатор связи с АВХ№2 неподвижен». Пусть вероятности и длительности вычисления значений независимых переменных x_1, x_2, x_3 имеют следующие значения:

$p_1 = p_3 = 0.002$, $p_2 = 0.004$, $t_{11} = t_{31} = 30$сек., $t_{10} = t_{30} = 0.3$сек,
$t_{21} = 300$мин, $t_{20} = 2$мин.

Для оптимизации данной базы знаний по длительности вычислений реализуем представленный выше алгоритм. И-ИЛИ-граф представляет собой корневую ИЛИ-вершину y_1, ее потомки ИЛИ-вершины y_2 и y_3, и связанные с последними листья x_1, x_2, x_3. Отмечаем листья. Из неотмеченных вершин выбираем y_2. Оптимизируем утверждение $y_2 = x_1 \vee x_2$ по длительности вычислений. Согласно выражению (2.53) вычислим аддитивные ранги букв x_1 и x_2:

$T_1 = t_{11}+t_{10}(1-p_1)/p_1=30+0.3(1-0.002)/0.002=179.7;$

$T_2 = t_{21}+t_{20}(1-p_2)/p_2=300+2(1-0.004)/0.004=828.$

Так как $T_1 < T_2$, то согласно теореме 2.4.2 запись утверждения $y_2 = x_1 \vee x_2$ является оптимальной и не требует перестановки. Отмечаем вершину графа y_2. Рассмотрим следующую неотмеченную вершину графа, а именно y_3. Оптимизируем утверждение $y_3 = x_2 \vee x_3$. Исходя из равенства значений соответствующих статистических параметров букв x_1 и x_3, получим $T_3 = T_1$. Так как $T_3 < T_2$, то оптимальной записью рассматриваемого утверждения будет следующая: $y_3 = x_3 \vee x_2$. Теперь статистические параметры целей y_2 и y_3 стали одинаковыми, поэтому оптимизировать утверждение $y_1 = y_2 \vee y_3$ не имеет смысла. То есть оптимизации записи утверждения y_3 оказалось достаточно для оптимизации всей базы знаний.

В будущем, когда начнут внедряться системы искусственного интеллекта в программном обеспечении систем управления, например, электроэнергетическими установками межпланетных кораблей, оптимизация баз знаний по быстродействию или достоверности вычислений согласно изложенным в данной главе методам станет актуальной задачей. В настоящее время оптимизация булевых формул по длительности вычислений использована автором при разработке алгоритмов управления автономными электроэнергетическими установками.

2.7. Выводы

Получены следующие результаты.

1. Рассмотрены четыре модели вычисления бесповторных булевых формул: операторные схемы, линейные бинарные графы, ортогональные формы и предложенные автором нечеткие множества

переходов. При этом установлены свойства линейных бинарных графов и взаимно однозначное соответствие бесповторных булевых формул, линейных бинарных графов, ортогональных форм и нечетких множеств переходов. Разработана полная система алгоритмов взаимно однозначного преобразования четырех форм представления бесповторных булевых формул.

2. Исследованы статические критерии сложности булевых формул: число и длина путей в линейном бинарном графе, а также нечеткий критерий сложности. При этом:

- предложен новый алгоритм подсчета числа единичных и нулевых путей в линейных бинарных графах;

- получены оценки числа и длины путей в линейных бинарных графах, реализующих произвольные скобочные булевы формулы, в том числе знакопеременные пороговые, дизъюнктивные нормальные формы, инвариантные булевы формулы;

- предложены формулы для вычисления нечеткого критерия сложности в виде лингвистической переменной;

- введено понятие веса буквы конъюнкции (дизъюнкции), сформулирована и доказана теорема об оптимальной записи конъюнкции (дизъюнкции) при неубывании весов букв формулы;

- предложен алгоритм оптимизации произвольной булевой формулы по числу путей в соответствующем линейном бинарном графе путем перестановки членов в булевой формуле;

- рассмотрен пример оптимизации записи булевой формулы по числу путей в соответствующем линейном бинарном графе, подтверждающий правильность предложенного алгоритма оптимизации и применимость его к алгоритмам управления, например автономными электроэнергетическими установками.

3. Исследован динамический аддитивный критерий сложности булевых формул, определяющий, например длительность их вычисления, и учитывающий статистические параметры независимых переменных формулы. При этом:

- введено понятие аддитивного ранга дизъюнкции (конъюнкции), сформулирована и доказана теорема об оптимальной по данному критерию сложности записи дизъюнкции (конъюнкции);

- предложен алгоритм оптимизации произвольной булевой формулы по данному критерию путем перестановок ее членов, что позволяет уменьшить, например длительность вычисления значения формулы.

- рассмотрен пример оптимизации булевой формулы по длительности вычисления, из которого следует, что предложенный алгоритм позволяет существенно снизить временные затраты на вычисление значения булевых формул, например в программах управления электроэнергетическими установками кораблей.

4. Исследован динамический мультипликативный критерий сложности булевых формул, определяющий, например достоверность их вычисления, и учитывающий статистические параметры независимых переменных формулы. При этом:

- введено понятие мультипликативного ранга дизъюнкции (конъюнкции), сформулирована и доказана теорема об оптимальной по данному критерию сложности записи дизъюнкции (конъюнкции);

- предложен алгоритм оптимизации произвольной булевой формулы по данному критерию путем перестановок ее членов, что позволяет повысить, например достоверность вычисления значения формулы.

- Рассмотрен пример, из которого следует, что достоверность вычислений, например в советующей системе для электроэнергетической системы, зависит от перестановок членов булевых формул и подвергается оптимизации согласно предложенному алгоритму.

5. Предложен алгоритм оптимизации продукционных баз знаний по длительности или достоверности вычислений на базе указанных в п.п. 3 и 4 настоящих выводов и приведен пример такой оптимизации.

6. Данные методики оптимизации использованы в конкретных системах управления различными установками.

ЛИТЕРАТУРА к главе 2

3. Абрамов С.А Элементы анализа программ. М.: Наука. 1986. 128 с.

4. Аверкин А.Н., Батыршин И.З., Блишун А.Ф. и др. Нечеткие множества в моделях управления и искусственного интеллекта. М.: Наука. 1986. 312 с.

5. Автоматизация проектирования цифровых устройств / С.И. Баранов, С.А.Майоров, Ю.П.Сахаров, В.А.Селютин. Л.: Судостроение. 1979. 264 с.

6. Автоматизированное проектирование цифровых устройств / С.С.Бадулин, Ю.М.Барнаулов, В.А.Бердышев и др. М.: Радио и связь. 1981. 240 с.

7. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под ред. В.И.Варшавского. – М.: Наука, 1986. 400 с.

10. Алагич С., Арбиб М. Проектирование корректных структурированных программ. М.: Радио и связь. 1984. 264 с.

11. Алексенко А.Г. Основы микросхемотехники. Элементы морфологии микроэлектронной аппаратуры. М.: "Советское радио". 1977. 408 с.

12. Алиев Р.А., Абдикеев Н.М., Шахназаров М.М. Производственные системы с искусственным интеллектом. М.: Радио и связь. 1990. 264 с.

13. *Алферова З.В. Теория алгоритмов. М.: Статистика. 1973. 164 с.*

14. *Андерсон Р. Доказательство правильности программ. М.: Мир. 1982. 168 с.*

15. *Арженко А.Ю., Чугаев Б.Н. Оптимальные бинарные вопросники. М.: Энергоатомиздат. 1989. 128 с.*

16. *Арженко А.Ю., Чугаев Б.Н. Оптимизация транзитивных бинарных вопросников // Автоматика и телемеханика. 1985. № 2. С. 159 - 164.*

18. *Арсеньев Ю.Н., Журавлев В.М. Проектирование систем логического управления на микропроцессорных средствах. М.: Высшая школа. 1991. 343 с.*

19. *Артюхов В.Л., Копейкин Г.А., Шалыто А.А. Настраиваемые модули для управляющих логических устройств. Л.: Энергоиздат.1981. 168с.*

24. *Артюхов В.Л., Кузнецов Б.П., Шалыто А.А. Настраиваемые логические устройства для судовых управляющих систем Конспект лекций. Л.: Институт повышения квалификации судостроительной промышденности. 1986. 44с.*

28. *Архангельский А.Я. Программирование в C++Builder 5. М.: Бином. 2000. 1152 с.*

29. *Асаи К., Ватада Д., Иваи С. и др. Прикладные нечеткие системы. М.: Мир. 1993. 368 с.*

30. *Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы, матроиды, алгоритмы. Ижевск: НИЦ "Регулярная и хаотическая динамика". 2001. 288 с.*

32. *Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1. Синтаксический анализ. М.: Мир. 1978. 614 с.*

33. *Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 2. Компиляция. М.: Мир. 1978. 488 с.*

35. *Бадулин С.С. Автоматизированное проектирование цифровых устройств. М.: Радио и связь. 1981. 240 с.*

36. *Байцер Б. Архитектура вычислительных комплексов. Том I. М.: Мир. 1974. 498 с.*

38. Баранов С.И. Синтез микропрограммных автоматов. Л.: Энергия, 1979. 232 с.

41. Басакер Р., Саати Т. Конечные графы и сети. М.: Наука. 1973. 368 с.

42. Батанов Л.А. Автоматизация проектирования цифровых вычислительных систем. М.: Энергия. 1978. 80 с.

43. Безбородов Ю.М. Индивидуальная отладка программ. М.: Наука. 1982. 192 с.

44. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем. СПб.: БХВ-Петербург. 2002. 464 с.

45. Берлин А.Н. Универсальная программа и принципы ее применения. СПб.: "Петеркон". 2001. 228 с.

47. Бесекерский В.А., Изранцев В.В. Системы автоматического управления с микроЭВМ. М.: Наука. 1987. 320 с.

49. Блох А.Ш. Граф-схемы и их применение. Минск: Вышейшая школа. 1975. 304 с.

50. Блох А.Ш. Синтез переключательных схем. Минск: Наука и техника. 1966. 200 с.

51. Боглаев Ю.П. Вычислительная математика и программирование. М.: Высш. школа. 1990. 544 с.

52. Богомолов А.М., Твердохлебов В.А. Целенаправленное поведение автоматов. Киев: Наукова думка. 1975. 124 с.

54. Бодякин В.И. Куда идешь, Человек? Основы эволюциологии. Информационный подход. М.: СИНТЕГ. 1998. 332 с.

55. Борисов А.Н., Алексеев А.В., Меркурьева Г.В. и др. Обработка нечеткой информации в системах принятия решений. М.: Радио и связь. 1989. 304 с.

56. Боэм Б., Браун Дж., Каспар Х. и др. Характеричтики качества программного обеспечения. М.: Мир. 1981. 208 с.

57. Братко И. Программирование на языке Пролог для искусственного интеллекта. М.: Мир. 1990. 560 с.

58. Брукшир Дж. Гленн. Введение в компьютерные науки. Общий обзор. М.: "Вильямс". 2001. 688 с.

59. Бузовский О.В., Муниф Хасан. Критерии оценки скобочных форм булевых функций. // Вестник КПИ. Автоматика и электроприборостроение. Вып. 20. Киев: Киевский политех. ин-т. 1983. С.70 - 73.

60. Бутаков Е.А. Методы создания качественного программного обеспечения. М.: Энергоатомиздат. 1984. 232 с.

61. Бутин Ю.Н., Золотаревская М.Я., Кириллов А.П., Юнг В.Н. О реализации алгоритмов логического управления в специализированных программируемых устройствах // Автоматика и телемеханика. 1983. № 6. С. 131-140.

63. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. М.: "Бином", СПб.: "Невский диалект". 1998. 560 с.

66. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир. 1981. 320 с.

67. Варжапетян А.Г., Глущенко В.В. Системы управления: исследование и компьютерное проектирование. М.: Вузовская книга. 2000. 328 с.

68. Варфоломеева Е.П., Дудкин В.С. Некоторые вопросы реализации устройств управления с помощью бинарных программ. // Изв. ЛЭТИ. 1981. № 285. С.13 - 17.

70. Вельбицкий И.В. и др. Технологический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6. М.: Статистика, 1980. 263 с.

72. Воклер И.Э. Об эффективности использования бинарных программ в системах моделирования дискретных устройств. // Моделирование дискретных управляющих и вычислительных систем. Тезисы докладов IV Всесоюзного семинара. Свердловск. Институт математики и механики УНЦ АН СССР. 1984. С.17.

73. Воробьев Н.Н. Числа Фибоначчи. М.: Наука. 1983. 164 с.

76. Гаврилов М.А. Избранные труды. Теория релейных и конечных автоматов. М.: Наука. 1983. 280 с.

77. Гаврилов М.А., Девятков В.В., Пупырев Е.И. Логическое проектирование дискретных автоматов. М.: Наука. 1977. 352с.

78. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер. 2000. 384 с.

80. Глазунов Л.П., Грабовецкий В.П., Щербаков О.В. Основы теории надежности автоматических систем управления. Л.: Энергоатомиздат. 1984. 208 с.

81. Гласс Р. Руководство по надежному программированию. М.: Финансы и статистика. 1982. 256 с.

83. Глушков В.М. О применении абстрактной теории автоматов для минимизации микропрограмм // Изв. АН СССР. Техническая кибернетика. 1964. № 1. С. 3 - 8.

84. Глушков В.М. Синтез цифровых автоматов. М.: Физматгиз. 1962. 476 с.

85. Глушков В.М. Теория автоматов и формальные преобразования микропрограмм // Кибернетика. 1965. № 5. С. 1 - 9.

86. Гольденберг Л.М. Импульсные и цифровые устройства. – М.: Связь. 1973. – 496 с.

90. Горбатов В.А. Семантическая теория проектирования автоматов. М.: Энергия. 1979. 264 с.

91. Горбатов В.А. Фундаментальные основы дискретной математики. Информационная математика. М.: Наука. 1999. 544 с.

94. Грин Д., Кнут Д. Математические методы анализа алгоритмов. М.: Мир. 1987. 120 с.

95. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. М.: Мир. 1975. 544 с.

96. Грис Д. Наука программирования. М.: Мир. 1984. 416 с.

97. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. М.: Мир. 1998. 703 с.

104. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов. М.: Мир. 1981. 368 с.

105. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир. 1975. 247 с.

106. Девятков В.В. Программная реализация управляющих алгоритмов. // Автоматизированное проектирование дискретных управляющих устройств. М.: Наука. 1980. С 30 - 62.

108. Джамп Д. AutoCAD. Программирование. М.: Радио и связь. 1992. 336 с.
110. Дружинин Г.В., Сергеева И.В. Качество информации. М.: Радио и связь. 1990. 172 с.
112. Дьяченко В.Ф. Лазарев В.Г., Савин Г.Г. Управление на сетях связи. М.: Наука. 1967. 166 с.
113. Дьяченко В.Ф. Минимизация логических схем алгоритмов // Кибернетика. 1978. № 6. С. 46 - 52.
114. Дьяченко В.Ф. О методе перехода от логических схем алгоритмов к булевым функциям // Техническая кибернетика. 1963. № 6. С. 84 - 86.
115. Евстигнеев В.А. Применение теории графов в программировании. М.:Наука. 1985. 352 с.
117. Ершов А.П. Введение в теоретическое программирование. М.: Наука. 1977. 288 с.
118. Ершов А.П. Об операторных схемах Янова. // Проблемы кибернетики. Вып. 20. М.: Физматгиз. 1968. С.181 - 200.
119. Закревский А.Д. Синтез асинхронных автоматов на ЭВМ. Минск: Наука и техника. 1975. 184 с.
120. Захаров В.Н., Поспелов Д.А., Хазацкий В.Е. Системы управления. М.: Энергия. 1977. 424 с.
122. Змитрович А.И. Интеллектуальные информационные системы. Минск: ТетраСистемс. 1997. 368 с.
123. Золотаревская М.Я. О реализации систем булевых функций в программируемых логических устройствах. // Проблемы управления в технике, экономике, биологии. М.: Наука. 1980. С. 24 - 31.
124. Зыков А.А. Основы теории графов. М.: Наука. 1987. 384с.
125 Ивани А., Смелянский Р.Л. Элементы теоретического программирования. М.: Из-во МГУ. 1985. 192 с.
126. Иванов Б.Н. Дискретная математика. Алгоритмы и программы. М.: Лаборатория базовых знаний. 2001. 288 с.

127. Иванов Г.И. О применении частичных подстановок для временного разложения булевых функций в обобщенные граф-схемы алгоритмов // Автоматика и телемеханика. 1983. № 11. С.120 - 131.

129. Игнатьев М.Б., Фильчаков В.В., Осовецкий Л.Г. Активные методы обеспечения надежности алгоритмов и программ. СПб.: Политехника. 1992. 288 с.

130. Йодан Э. Структурное проектирование и конструирование программ. М.: Мир. 1979. 416 с.

131. Калужнин Л.А. Об алгоритмизации математических задач. // Проблемы кибернетики. Вып. 2. М.: Физматгиз. 1959. С. 52 -68.

132. Камынин С.С., Любимский Э.З., Шура-Бура М.Р. Об автоматизации программирования при помощи программирующей программы. // Проблемы кибернетики. Вып. 1. М.: Физматгиз. 1956. С.135-171.

134. Касьянов В.Н. Оптимизирующие преобразования программ. М.: Наука. 1988. 336 с.

135. Касьянов В.Н., Поттосин И.В. Методы построения трансляторов. Новосибирск: Наука. 1986 344 с.

136. Керниган Б., Ритчи Д. Язык программирования Си. М.: Финансы и статистика. 1985. 279 с.

139. Кнут Д. Искусство программирования для ЭВМ. Том 3. Сортировка и поиск. М.: Мир. 1978. 844 с.

140. Кобринский Н.Е., Трахтенброт Б.А. Введение в теорию конечных автоматов. М.: Физматгиз. 1962. 404 с.

145. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦМНО. 1999. 960 с.

146. Котов В.Е., Сабельфельд В.К. Теория схем программ. М.: Наука. 1991. 248 с.

147. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь. 1982. 432с.

148. Кристофидес Н. Теория графов: Алгоритмический подход. М.: Мир. 1978. 432 с.

149. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия - Телеком. 2001. 382 с.

152. Кузнецов Б.П. Длительность вычислений на линейных бинарных графах // Автоматика и телемеханика. 1994. №9.

С.166-172 .

153. Кузнецов Б.П. Достоверность управляющих сигналов в контуре логического управления // Автоматика и телемеханика. 1993. №10. С.160-167.

158. Кузнецов Б.П. Оптимизация продукционной базы знаний по достоверности и длительности вывода // Теория и системы управления. 1996. №5. С. 45 - 50.

159. Кузнецов Б.П. Последовательно-событийные автоматы // Приборы и системы. Управление, Контроль, Диагностика. 2001. № 5. С.15 -21.

163. Кузнецов Б.П. Психология автоматного программирования // BYTE/Россия. 2000. №11. С. 22-29.

164. Кузнецов Б.П. Распределенные конечные автоматы // Приборы и системы. Управление, Контроль, Диагностика. 2000. № 2. С. 9 - 12.

166. Кузнецов Б.П. Система программирования контроллеров // Программист. 2002. № 3. С.17-22.

168. Кузнецов Б.П. Структура и сложность модулей циклических программ. // Автоматика и телемеханика. 1999. № 2.

С.151-165.

176. Кузнецов Б.П., Шалыто А.А. Реализация булевых формул линейными бинарными графами. I. Синтез и анализ // Известия РАН. Техническая кибернетика. 1994. №5. С.132-142.

177. Кузнецов Б.П., Шалыто А.А. Реализация булевых формул линейными бинарными графами. II. Оценки числа и суммарной длины путей // Известия РАН. Теория и системы управления. 1995. №3. С.144 - 153.

178. Кузнецов Б.П., Шалыто А.А. Реализация булевых формул линейными бинарными графами. III. Оптимизация числа и суммарной длины путей // Известия РАН. Теория и системы управления. 1995. №5 С.214 - 223.

180. Кузнецов Б.П., Шалыто А.А. Система преобразований некоторых форм представления булевых функций//Автоматика и телемеханика. 1985. №11. С.120-127.

181. Кузнецов Б.П., Шалыто А.А. Структурный подход к программной реализации булевых формул // Автоматика и вычислительная техника. 1985. № С. 84 - 88

183. Кузнецов О.П. О программной реализации логических функций и автоматов. I. Анализ и синтез бинарных программ // Автоматика и телемеханика. 1977. №7. С. 163-174.

184. Кузнецов О.П. О программной реализации логических функций и автоматов. II. Время вычислений бинарных программ // Автоматика и телемеханика. 1977. №9. С. 137-149.

185. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. М.: Энергоатомиздат. 1988. 480 с.

186. Кузьмин В.А. Оценка сложности реализации функций алгебры логики простейшими видами бинарных программ. // Методы дискретного анализа в теории кодов и схем. Новосибирск. 1976. Вып 29. С.24 - 36.

187. Куликов М.Я., Червенчук В.Д. О средствах оптимизации для таблиц решений // Кибернетика. 1984. № 2. С. 29 - 34.

189. Куприянов М.С., Матюшкин Б.Д. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования. СПб.: Политехника. 1998. 592 с.

190. Курош А.Г. Лекции по общей алгебре. М.: Физматгиз. 1962. 364с.

191. Кээваллик А.Э. Теорема декомпозиции конечных автоматов // Автоматика и вычислительная техника. 1974. № 1. С. 17 - 24.

192. Лаговиер Б.А. Об алгоритмах программной реализации автоматов // Автоматика и телемеханика. 1979. № 1. С.121 - 127.

193. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. М.: Энергоатомиздат, 1989. 328 с.

194. Ландау И.Я. Применение ЦВМ для проектирования ЦВМ. М.: Энергия. 1974. 152 с.

195. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. М.: Мир. 1982. 406 с.
196. Липаев В.В. Качество программного обеспечения. М.: Финансы и статистика. 1983. 263 с.
197. Липаев В.В. Надежность программного обеспечения АСУ. М.: Энергоиздат. 1981. 240 с.
198. Липаев В.В. Надежность программных средств. М.: СИНТЕГ. 1998. 232 с.
199. Липаев В.В. Проектирование программных средств. М.: Высшая школа. 1990. 303 с.
200. Липаев В.В. Тестирование программ. М.: Радио и связь. 1986. 296 с.

203. Литвак Б.Г. Экспертная информация: Методы получения и анализа. М.: Радио и связь. 1982. 184 с.
204. Литвиненко Н.И., Бардин А.А. Минимизация альтернативных графов функцией селекции // Вопросы радиоэлектроники. Сер. Электронная вычислительная техника. 1984. № 3. С. 45 -51.
205. Лорьер Ж.-Л. Системы искусственного интеллекта. М.: Мир. 1991. 568 с.
206. Луховицкая Э.С. Блок обработки логических условий в ПП-2. // Проблемы кибернетики. Вып 1. М.: Физматгиз. 1958. С.172-177.
207. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. М.: Мир. 1979. 656 с.
208. Любченко В.С. Батарея, огонь! Или Задача Майхилла для Microsoft Visual C++: О синхронизации процессов в среде Windows // Мир ПК. 2000. № 2. С. 148 - 155.
209. Любченко В.С. Новые песни о главном (римейк для программистов) // Мир ПК. 1998. № 6. С.114 - 119.
210. Любченко В.С. Стрелки! Нале-во! Из задач для Microsoft Visual C++ // Мип ПК. 2001. № 8. С.122 - 124, № 9. С. 128 - 131.
211. Ляпунов А.А. О логических схемах программ. // Проблемы кибернетики. Вып.1 М.: Физматгиз. 1958. С. 46 - 74.

212. *Майерс Г. Искусство тестирования программ. М.: Финансы и статистика. 1982. 176 с.*

213. *Майерс Г. Надежность программного обеспечения. М.: Мир, 1980. 359 с.*

214. *Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. Л.: Машиностроение. 1974. 432 с.*

215. *Малпас Дж. Реляционный язык Пролог и его применение. М.: Наука. 1990. 464 с.*

216. *Марголин М.С., Потапенко Т.П. Использование неполной исходной информации для упрощения программ // Управляющие системы и машины. 1981 №6. С. 78 - 81.*

217. *Маслов А.Н. Введение в язык программирования С. М.: Память. 1991. 64 с.*

218. *Мелихов А.Н. Ориентированные графы и конечные автоматы. М.: Наука. 1971. 416с.*

219. *Микропроцессорные средства производственных систем / В.Н.Алексеев, А.М.Коновалов, В.Г.Колосов и др. Л.: Машиностроение. 1988. 287 с.*

221. *Мюллер И. Эвристические методы в инженерных разработках. М.: Радио и связь. 1984. 144 с.*

223. *Непомнящий В.А., Рякин О.М. Прикладные методы верификации программ. М.: Радио и связь. 1988. 256 с.*

224. *Нильсон Н. Принципы искусственного интеллекта. М.: Радио и связь. 1985. 376 с.*

225. *Новорусский В.В. Конечноавтоматные системы управления (принципы построения и анализ поведения). Новосибирск: Наука. 1982. 269 с.*

226. *Овчаренко Н.И. Автоматика электрических станций и электроэнергетических систем. М.: Изд-во НЦ ЭНАС. 2000. 504 с.*

227. *Одинцов И.О. Профессиональное программирование. Системный подход. СПб.: БХВ-Петербург. 2002. 512 с.*

228. *Олссон Г., Пиани Д. Цифровые системы автоматизации и управления. СПб.: Невский диалект. 2001. 557 с.*

229. *Оре О. Теория графов. М.: Наука. 1980. 336 с.*

230. *Петер Р. Рекурсивные функции. М.: Из-во иностр. литер. М. 1954. 250 с.*

231. *Петренко А.И. Автоматизированное проектирование СБИС на базовых матричных кристаллах. М.: Радио и связь. 1988. 184с*

232. *Пивоваров А.Н. Методы обеспечения достоверности информации в АСУ. М.: Радио и связь. 1982. 144 с.*

233. *Поваров Г.Н. О логическом синтезе электронных вычислительных и управляющих схем // Логические исследования. М.: Изд-во АН СССР. 1959. С. 39-52.*

234. *Понтрягин Л.С. Знакомство с высшей математикой: Дифференциальные уравнения и их приложения. М.: Наука. 1988. 208 с.*

235. *Поспелов Д.А. Логические методы анализа и синтеза схем. М.: Энергия. 1974. 368 с.*

236. *Построение экспертных систем / Под ред. Ф.Хейеса-Рота и др. М.: Мир. 1987. 441 с.*

237. *Поттосин Ю.В., Шестаков Е.А. Приближенные алгоритмы параллельной декомпозиции автоматов // Автоматика и вычислительная техника. 1981. № 2. С.31 - 38.*

238. *Проверка и утверждение программ реального времени / Под ред. Уи.Дж.Квирка. Киев: Наукова думка. 1990. 216 с.*

240. *Пупырев Е.И. Интерпретирующие программы реализации булевых функций и автоматов // Автоматика и телемеханика. 1982. № 1. С.131 - 139.*

241. *Пупырев Е.И. Перестраиваемые автоматы и микропроцессорные системы. М.: Наука. 1984. 192 с.*

242. *Проектирование программного обеспечения САПР. Разработка САПР. Кн.3. / Б.С.Федоров, Н.Б.Гуляев. М.: Высшая школа. 1990. 159 с.*

243. *Имитационное моделирование. Разработка САПР. Кн.9. / В.М.Черненький. М.: Высшая школа. 1990. 112 с.*

244. *Рейнгольд Э.,Нивергельт Ю., Део Н. Комбинаторные алгоритмы. М.: Мир. 1980. 476 с.*

245. Рогинский В.Н. Нахождение путей и их характеристик в сложных структурах. // Теория дискретных управляющих устройств. М.: Наука. 1982. С.167 - 175.

248. Рябинин И.А., Киреев Ю.Н. Надежность судовых электроэнергетических систем и судового оборудования. Л.: Судостроение. 1974. 264 с.

249. Саркисян А.А. Повышение качества программ на основе автоматизированных методов. М.: Радио и связь. 1991. 160 с.

250. Свами М., Тхуласираман К. Графы, сети и алгоритмы. М.: Мир. 1984. 455 с.

251. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов. М.: Эдиториал УРСС. 2001. 224 с.

253. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. М.: Горячая линия-Телеком. 2001. 630 с.

254. Средства отладки больших систем / Под ред. Р.Растина. М.: Статистика. 1977. 135 с.

255. Танаев В.С. Поварич М.П. Синтез граф-схем алгоритмов выбора решений. Минск: Наука и техника. 1974. 112 с.

256. Тейер Т., Липов М., Нельсон Э. Надежность программного обеспечения. М.: Мир. 1981. 323 с.

257. Теория и методы автоматизации проектирования вычислительных систем / Под ред. М.Брейера. М.: Мир. 1977. 285 с.

258. Фрайтаг Г., Годе В., Якоби Х. и др. Введение в технику работы с таблицами решений. М.: Энергия. 1979. 88 с.

260. Фридл Дж. Регулярные выражения. Библиотека программиста. СПб.: Питер. 2001. 352 с.

261. Фридмен М., Ивенс Л. Проектирование систем с микрокомпьютерами. М.: Мир. 1986. 405 с.

262. Фрир Дж. Построение вычислительных систем на базе перспективных микропроцессоров. М.: Мир. 1990. 413 с.

263. Фритч В. Применение микропроцессоров в системах управления. М.: Мир. 1984. 464 с.
264. Фудзисава Т., Касами Т. Математика для радиоинженеров: Теория дискретных структур. М.: Радио и связь. 1984. 240 с.
265. Хамби Э. Программирование таблиц решений. М.: Мир. 1976. 88 с.
266. Холстед М.Х. Начала науки о программах. М.: Финансы и статистика. 1981. 128 с.
267. Хьюз Дж., Мичтом Дж. Структурный подход к программированию. М.: Мир. 1980. 278 с.
268. Чери С., Готлоб Г., Танка Л. Логическое программирование и базы данных. М.: Мир. 1992. 352 с.
269. Чирков М.К. Основы общей теории конечных автоматов. Л.: Изд-во ЛГУ. 1975. 280 с.
274. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука. 2000. 780 с.
277. Шестаков В.И. Алгебраический метод синтеза многотактных релейных систем // ДАН СССР. 1954. Т.99. № 6. С.987 - 990.
278. Шилд Г. Программирование на BORLAND C++ для профессионалов. Минск: ООО "Попурри". 1999. 800 с.
279. Шинкин В.Н. Метод минимизации бинарных программ, реализующих логические функции // Автоматика и телемеханика. 1983. №1. С. 131-140.
280. Шнейдерман Б. Психология программирования. М.: Радио и связь. 1984. 304 с.
281. Штейн М.Е., Штейн Б.Е. Методы машинного проектирования цифровой аппаратуры. М.: Сов. Радио. 1973. 296 с.
282. Штрик А.А., Осовецкий Л.Г., Мессих И.Г. Структурное проектирование надежных программ встроенных ЭВМ. Л.: Машиностроение. 1989. 296 с.
283. Шураков В.В. Надежность программного обеспечения систем обработки данных. М.: Статистика. 1981. 216 с.
284. Щербаков Н.С. Достоверность работы цифровых устройств. М.: Машиностроение. 1989. 224 с.

285. Энциклопедия кибернетики. Том I. – Киев. Главная редакция УСЭ. – 1974. с.60.
287. Ющенко Е.Л., Цейтлин Г.Е., Грицай В.П. И др. Многоуровневое структурное проектирование программ: Теоретические основы, инструментарий. М.: Финансы и статистика. 1989. 208 с.
288. Язык Пролог в пятом поколении ЭВМ / Ильинский Н.И. М.: Мир. 1998. 501 с.
289. Янов Ю.И. О логических схемах алгоритмов. // Проблемы кибернетики. Вып.1. М.: Физматгиз. 1958. С. 75 - 127.
290. Akers S.B. Binary Decision Diagramms // IEEE Transactions on Computers. 1978. Vol. C-27.№6, pp. 509-516.
291. Bryant R. Graph-based algorithms for boolean function ma-nipulation // IEEE Transactions on Computers. 1986. Vol. C-35, №8, pp. 677- 691.
292. Cioffi G., Constantini E., DeJiulio S. A new approach to the decomposition of sequential systems // Digital Processes. 1977. Vol. 3, pp. 35 - 48.
293. Devadas S., Newton R. Decomposition and factorization of sequential finite state machines // IEEE Transactions on CAD. 1989. Vol. 8. № 11, pp. 1290 - 1300.
294. Hachtel G.D.,Rho J.-K., Somenzi F., Jacoby R. Exact and heuristic algorithms for the minimization of incompletely specified state machines // Proc.of the European Design Automation Conference (EURO-DAC'91). Amsterdam. The Netherlands. 1991.
pp. 184-191.
295. Harel D. Statecharts: A Visual Formalism for Complex Systems // Science of Computer Programming. 1987. Vol. 8,
pp. 231 - 274
296. Heckel P. A Technique for Isolating Differences Between Files // Communications of the ACM. 1978. Vol. 21, № 4, pp. 264-268.
297. Huffman D.A. The synthesis of sequential switching circuits // Journal of the Franklin Inst. 1954. Vol.257. № 3 and 4,
pp. 161 - 190, 275 - 303.

298. Hunt J.W. and Szymanski T.G. A fast algorithm for computing longest common subsequences // Communications ACM. 1977.
Vol. 20 (5) pp. 350 - 353.
299. Kim J., Newborn M. The simplification of sequential machines with input restrictions // Transactions on Computers. 1972.
Vol. C-21, pp. 1440 - 1443.
300. Kleene S.C. Representation of events in nerve sets and finite automata // Automata Studies. Princeton University Press. 1956, pp.3 - 41.
301. Lee C.Y. Representation of switching circuits by binary decision programms. // Bell System Technology Journal. 1959. Vol 38. № 4, pp. 985 - 1000.
302. Mealy G.H.Method for synthesizing sequential circuts. // BellSystem Techn. Journal. 1955. Vol. 34, pp.1045 - 1079.
303. Moore E.F. Gedanken-experiments on sequential machines // Automata Studies. Princeton University Press. 1956, pp. 129 - 153.
304. Myers E.W. An O(ND) difference algorithm and its variations // Algorithmica. 1986. № 1, pp. 251 - 266.
305. Rubey R.J., Hartwick R.D. Quantitative Measurement of Program Quality // Proceedings of ACM National Conference. 1968.
pp. 671 - 677.
306. Tung B., Kleinrock L. Distributed Control Methods. // Proceeding of the 2nd Int. Symp. On High Performance Distributed Computing. 1993, pp.206 - 215.
307. Рабинович М.А. Цифровая обработка информации для задач оперативного управления в электроэнергетике. М.: НЦ ЭНАС. 2001. 344 с.
310. Кузьмин В.Б., Травкин С.И. Теория нечетких множеств в задачах управления и принципах устройства нечетких процессоров. Обзор зарубежной литературы // Автоматика и телемеханика. 1992. № 11. С. 3 - 36.
312. Кузнецов Б.П. Нечеткое представление и нечеткий критерий сложности булевых формул // Системы управления и обработки информации. ФНПЦ «НПО «Аврора». СПб. 2002. – Вып. 4.
С.97 – 103.