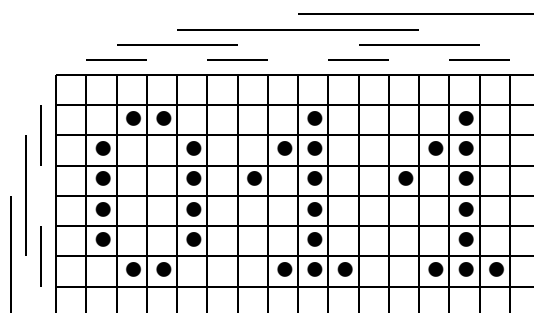


Министерство образования Российской Федерации
Томский государственный университет
Факультет прикладной математики и кибернетики

БУЛЕВЫ ФУНКЦИИ



Учебно-методическое пособие
часть III

Томск
2004

"УТВЕРЖДАЮ"
Декан ФПМК
профессор _____ А.М. Горцев
" ____ " _____ 2004 г.

РАССМОТРЕНО и УТВЕРЖДЕНО методической комиссией
факультета прикладной математики и кибернетики

Председатель комиссии
профессор _____ С.Э. Воробейчиков
Протокол 21 от 10 декабря 2004 г.

В данной части пособия ставится задача минимизации булевых функций и рассматривается двухэтапный метод ее решения, состоящий в построении сокращенной ДНФ и выделении из нее конъюнкций кратчайших и минимальных ДНФ на основе поиска соответствующих покрытий таблицы Квайна. Кроме того, излагается метод построения приближенной кратчайшей ДНФ.

Пособие предназначено для студентов, изучающих теорию булевых функций.

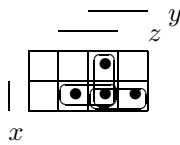
Составители: доцент каф. ЗИК С.В. Быкова
программист каф. ВМиММ Ю.Б. Буркатовская

11. Минимизация булевых функций.

В предыдущем разделе мы познакомились с различными типами ДНФ и научились их находить визуально по матрице Грея. Перейдем к изучению формальных методов построения сокращенной, безызбыточных, минимальных и кратчайших ДНФ булевой функции, заданной совершенной или произвольной ДНФ.

Интерес к кратчайшим и минимальным ДНФ основан на их оптимальности (по длине и рангу соответственно), которая положительно проявляется, по крайней мере, в следующих двух случаях. Во-первых, с оптимальными ДНФ легче оперировать, то есть вычислять значения функции, строить матрицу Грея и подставлять в другие формулы. Во-вторых, оптимальные ДНФ более предпочтительны для построения по ним схем из логических элементов: дизъюнкторов, конъюнкторов и инверторов.

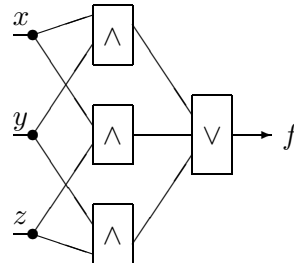
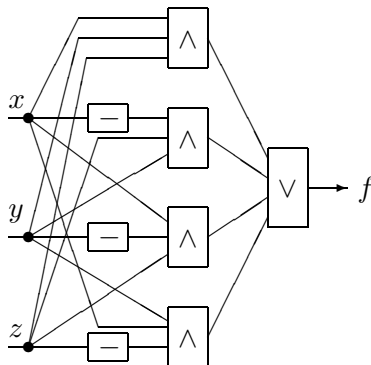
Пример. Рассмотрим мажоритарную функцию.



$$\text{СовДНФ}_f = \bar{x}yz \vee x\bar{y}z \vee xyz \vee xy\bar{z}$$

$$\text{КратДНФ}_f = xy \vee xz \vee yz$$

Нарисуем схемы по совершенной и кратчайшей ДНФ. Логические элементы изобразим в виде соответственно обозначенных прямоугольников.



Схема, построенная по кратчайшей ДНФ (справа), оказалась проще: она содержит меньше элементов. •

Интерес к сокращенной ДНФ вызван тем, что она является промежуточной при построении кратчайших, минимальных и безызбыточных ДНФ.

Безызбыточные ДНФ интересны как сами по себе, так как часто оказываются близкими к оптимальным, так и тем, что среди них находятся все минимальные и простые кратчайшие ДНФ.

Определение. *Минимизировать булеву функцию* это значит построить ее кратчайшую или минимальную ДНФ или, может быть, все кратчайшие или все минимальные ДНФ (конкретная постановка задачи уточняется дополнительно).

Рассмотрим двухэтапный подход к минимизации, основанный на теоремах о кратчайшей и минимальных ДНФ [2.С.21], утверждающих, что любая минимальная и по крайней мере одна из кратчайших ДНФ состоят из простых импликант.

Первый этап: найдем все простые импликанты функции, то есть конъюнкции ее сокращенной ДНФ.

Второй этап: из сокращенной ДНФ выделим конъюнкции искомым ДНФ (кратчайших или минимальных).

11.1. Получение сокращенной ДНФ – первый этап минимизации

Изучим два метода получения сокращенной ДНФ, но прежде вспомним законы поглощения и склеивания конъюнкций и добавим к ним закон неполного склеивания.

Закон поглощения: $K_1 \vee K_1 K_2 = K_1$.

Закон склеивания: $xK \vee \bar{x}K = K$.

Закон неполного склеивания: $xK \vee \bar{x}K = xK \vee \bar{x}K \vee K$.

Закон обобщенного склеивания: $xK_1 \vee \bar{x}K_2 = xK_1 \vee \bar{x}K_2 \vee K_1 K_2$.

Очевидно, что закон неполного склеивания вытекает из закона обобщенного склеивания (при $K_1 = K_2 = K$), а закон склеивания – из закона неполного склеивания (если в последнем выполнить поглощения конъюнкций).

Далее наряду с языком ДНФ будем использовать язык интервалов как более наглядный при визуальном решении и более удобный при компьютерной реализации алгоритмов. Напомним аналогию между рядом понятий (относящихся к одной и той же булевой функции) на упомянутых языках.

Язык ДНФ	Язык интервалов
полная конъюнкция-импликанта	точка
элементарная конъюнкция-импликанта	допустимый интервал
простая импликанта	максимальный интервал
ДНФ	достаточное множество интервалов
совершенная ДНФ	множество всех точек
сокращенная ДНФ	множество всех максимальных интервалов
кратчайшая ДНФ	достаточное множество из минимального числа максимальных интервалов
минимальная ДНФ	достаточное множество максимальных интервалов с минимальной суммой рангов

Переведем на язык интервалов законы поглощения и склеивания, используя интервалы и представляющие их троичные векторы как синонимы.

Закон поглощения говорит о том, что если объединение двух интервалов I и I' совпадает с I , то интервал I' содержится в интервале I (поглощается им). Значит, поглощаемый вектор должен получаться из поглощающего заменой некоторых его внутренних компонент (–) на внешние (0 или 1).

Пример. В паре троичных векторов $\alpha = -1-0-$
 $\beta = 01-01$ вектор β поглощается вектором α . •

Закон склеивания объединяет два соседних интервала в один. Напомним, что соседние интервалы совпадают по номерам внешних компонент, но различаются по значению ровно одной из них (ортогональной). При склеивании получается интервал, который отличается от исходных лишь в ортогональной компоненте – она становится внутренней (операция склеивания соседних интервалов уже рассматривалась в [2.С.13]).

Пример. Результатом склеивания соседних векторов α и β является вектор γ :

$$\begin{aligned}\alpha &= 0 \ 0 \ 1 \ - , \\ \beta &= 0 \ 1 \ 1 \ - , \\ \gamma &= 0 \ - \ 1 \ - . \bullet\end{aligned}$$

Закон неполного склеивания объединяет два соседних интервала и добавляет к результату исходные интервалы.

Пример. Результатом неполного склеивания векторов α и β из предыдущего примера являются три вектора: α , β и γ . \bullet

Закон обобщенного склеивания объединяет соседние части двух смежных интервалов. Смежные интервалы могут не совпадать по номерам внешних компонент, но должны быть ортогональны ровно по одной из них. Результатом обобщенного склеивания являются три интервала: два исходных и третий, который строится следующим образом:

- компоненты, по которым исходные векторы совпадают, сохраняют свои значения;
- компоненты, которые в одном из векторов являются внешними, а в другом – внутренними, принимают значения внешних компонент;
- ортогональная компонента становится внутренней.

Пример. Результатом обобщенного склеивания смежных векторов α и β являются векторы α , β и γ :

$$\begin{aligned}\alpha &= 0 \ 0 \ 1 \ - \ 0 \ - \ 1 \ - , \\ \beta &= 1 \ 0 \ 1 \ - \ - \ 0 \ - \ 1 \ , \\ \gamma &= - \ 0 \ 1 \ - \ 0 \ 0 \ 1 \ 1 \ . \bullet\end{aligned}$$

11.1.1. Теорема Квайна и алгоритм Квайна–МакКласки

Первый метод построения сокращенной ДНФ булевой функции основан на следующей теореме.

Теорема (Квайна). Чтобы получить сокращенную ДНФ булевой функции из совершенной ДНФ, надо выполнить всевозможные неполные склеивания соседних конъюнкций, а затем всевозможные поглощения конъюнкций.

Опираясь на теорему Квайна, МакКласки сформулировал алгоритм, который организует построение сокращенной ДНФ более эффективно, чем это предложено в теореме.

Алгоритм Квайна–МакКласки.

Начало. Задана совершенная ДНФ булевой функции.

Шаг 1. Построим список всех точек функции (булевых векторов) и упорядочим их по неубыванию числа единиц – веса.

Шаг 2. Разобьем список на подмножества (классы) векторов одинакового веса. Обозначим через C_i класс векторов веса i .

Шаг 3. Выполним неполные склеивания всех соседних векторов классов C_i и C_{i+1} , $i = 0, 1, \dots, n - 1$. Векторы, участвующие в склеивании (α и β), отметим, а полученные векторы (γ) занесем в новый список и приведем в нем подобные.

Шаг 4. Если новый список векторов не пуст, возвратимся с ним на шаг 2 (заметим, что троичные векторы списка оказываются уже упорядоченными по числу единиц).

Конец. Выписав из всех списков неотмеченные векторы, получим множество всех максимальных интервалов, оно задает сокращенную ДНФ исходной функции.

Пример. Продемонстрируем выполнение алгоритма, для наглядности сопровождая его шаги матрицами Грея.

Начало. Задана совершенная ДНФ булевой функции:

$$\begin{aligned} \text{СовДНФ} = & \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee ab\bar{c}d \vee abcd \vee \\ & \vee abc\bar{d} \vee a\bar{b}cd. \end{aligned}$$

Шаги 1, 2. Совершенную ДНФ представляем списком точек, упорядочиваем их по весу и разбиваем на классы.

Список 0	Список 0 (упорядоченный)	
0 0 0 0	C_0 1) 0 0 0 0	
0 1 0 1	C_2 2) 0 1 0 1	
0 1 1 1	C_3 3) 0 1 1 1	
1 1 0 1	4) 1 0 1 1	
1 1 1 1	5) 1 1 0 1	
1 1 1 0	6) 1 1 1 0	
1 0 1 1	C_4 7) 1 1 1 1	

Шаг 3. Выполняя неполные склеивания векторов из классов C_2 и C_3 , а затем C_3 и C_4 , и отмечая в упорядоченном списке 0 склеиваемые векторы символом *, получаем список интервалов, состоящих из двух точек (список 1). Справа в новом списке указаны номера векторов – участников склеивания. Интервалы списка изображены для наглядности на двух матрицах Грея.

Список 0	Список 1	
C_0 1) 0 0 0 0	1) 0 1 – 1 (2,3)	
C_2 2) 0 1 0 1 *	2) – 1 0 1 (2,5)	
C_3 3) 0 1 1 1 *	3) – 1 1 1 (3,7)	
4) 1 0 1 1 *	4) 1 – 1 1 (4,7)	
5) 1 1 0 1 *	5) 1 1 – 1 (5,7)	
6) 1 1 1 0 *	6) 1 1 1 – (6,7)	
C_4 7) 1 1 1 1 *		

Шаг 4. Список 1 не пуст, поэтому идем на шаг 2 (так как склеивания производились в строгом порядке "сверху вниз", векторы в новом списке уже упорядочены по весу).

Шаги 2, 3. Разбиваем полученный список на классы C_2 , C_3 . Выполняя склеивания векторов из классов C_2 и C_3 , получаем список интервалов, состоящих из четырех точек (список 2). Приводим в нем подобные.

	Список 1	Список 2	
C_2	1) 0 1 – 1 *	1) – 1 – 1 (1,5)	
	2) – 1 0 1 *	2) – 1 – 1 (2,3)	
C_3	3) – 1 1 1 *		
	4) 1 – 1 1		
	5) 1 1 – 1 *		
	6) 1 1 1 –		

Шаг 4. Список 2 не пуст, но дальнейшее склеивание невозможно, поэтому список 3 окажется пустым, идем на конец.

Конец. Выписываем из всех списков неотмеченные векторы. Они задают сокращенную ДНФ:

0	0	0	0
1	–	1	1
1	1	1	–
–	1	–	1

$$\text{СокрДНФ} = \bar{a}\bar{b}\bar{c}\bar{d} \vee acd \vee abc \vee bd. \bullet$$

Сравнивая алгоритм Квайна-МакКласки с теоремой Квайна, оценим предложения МакКласки.

Во-первых, представление конъюнкций булевыми и троичными векторами делает вычисления более простыми и более приспособленными для компьютерной реализации.

Во-вторых, разбиение на классы позволяет не искать соседей в тех парах классов, в которых их не может быть, то есть в классах, номера которых отличаются более чем на единицу.

В-третьих, отметка склеиваемых соседей (но не вычеркивание их, так как один и тот же вектор может быть соседом нескольким векторам) позволяет свести поглощение к выписыванию неотмеченных векторов.

11.1.2. Теорема Блейка и алгоритм Блейка–Порецкого

Второй метод построения сокращенной ДНФ булевой функции основан на следующей теореме.

Теорема (Блейка). Чтобы получить сокращенную ДНФ булевой функции из произвольной ДНФ, надо выполнить всевозможные обобщенные склеивания смежных конъюнкций, а затем всевозможные поглощения конъюнкций.

На основе теоремы Блейка сформулирован алгоритм, который организует построение сокращенной ДНФ более эффективно, чем это предложено в теореме.

Алгоритм Блейка–Порецкого.

Начало. Задана произвольная ДНФ булевой функции.

Шаг 1. Построим список троичных векторов, представляющих конъюнкции ДНФ. Удалим из списка все векторы, поглощаемые другими. Если останется лишь один вектор, пойдем на конец. Иначе обозначим второй из оставшихся векторов через α .

Шаг 2. Найдем для вектора α очередной смежный вектор β среди векторов, расположенных в списке выше α . Если такого вектора β нет, то идем на шаг 5. Иначе обобщенно склеим α и β и полученный вектор γ припишем к списку последним.

Шаг 3. Если вектор γ поглощается хотя бы одним вектором из списка, то удалим γ (в частном случае γ может совпадать с одним из векторов списка, тогда удалим именно приписанный вектор γ , иначе произойдет "зацикливание" алгоритма, так как вектор γ будет вновь появляться в списке) и идем на шаг 2. Если вектор γ не поглощается, то удалим все векторы, поглощаемые им.

Шаг 4. Если вектор α не удален, то идем на шаг 2.

Шаг 5. Если вектор α удален, но был не последним в списке, то выберем в качестве нового α следующий по списку вектор и вернемся на шаг 2.

Конец. Векторы, оставшиеся в списке, задают сокращенную ДНФ исходной функции.

Алгоритм Блейка-Порецкого отличается от алгоритма Квайна-МакКласки прежде всего тем, что оперирует с произвольной ДНФ, в то время как алгоритм Квайна-МакКласки требует на входе совершенную ДНФ. Конечно, можно перейти от произвольной ДНФ к совершенной, как показано в [2.С.11], но она может оказаться очень длинной, и применение алгоритма Квайна-МакКласки станет неэффективным.

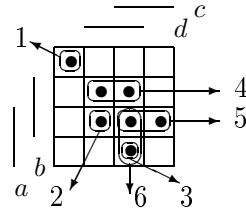
Пример. Продемонстрируем алгоритм Блейка-Порецкого, сопровождая его выполнение матрицами Грея.

Начало. Задана ДНФ булевой функции, рассмотренной в примере к алгоритму Квайна-МакКласки:

$$\text{ДНФ} = \bar{a}\bar{b}\bar{c}\bar{d} \vee ab\bar{c}d \vee a\bar{b}cd \vee \bar{a}bd \vee abc \vee acd.$$

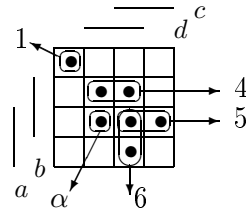
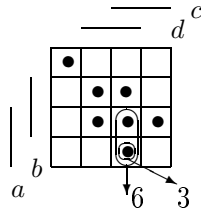
Шаг 1. ДНФ представляем списком векторов и нумеруем их.

- 1) 0 0 0 0
- 2) 1 1 0 1
- 3) 1 0 1 1
- 4) 0 1 - 1
- 5) 1 1 1 -
- 6) 1 - 1 1



Вычеркиваем из списка третий вектор, поглощаемый шестым (левая из нижних матриц демонстрирует это поглощение). Остальные векторы не поглощаются (правая матрица демонстрирует состояние списка). Выбираем в качестве α второй вектор.

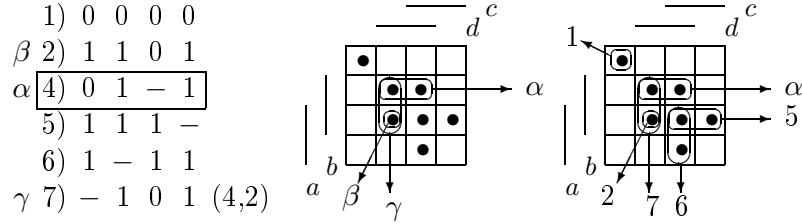
- 1) 0 0 0 0
- α 2) 1 1 0 1
- ~~3) 1 0 1 1~~
- 4) 0 1 - 1
- 5) 1 1 1 -
- 6) 1 - 1 1



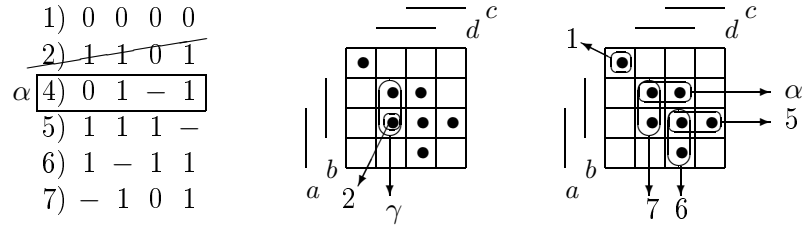
Шаг 2. Так как для α нет смежного вектора среди предыдущих, то идем на шаг 5.

Шаг 5. Так как вектор α был не последним в списке, то в качестве нового α рассматриваем следующий по списку (четвертый) вектор (см. ниже) и возвращаемся на шаг 2.

Шаг 2. Вектор α не смежен первому вектору, но смежен второму (по первой компоненте). Обозначаем его через β . Обобщенно склеиваем α и β и записываем результат – вектор γ в список седьмым. Отмечаем, что он получен из четвертого и второго (левая матрица демонстрирует склеивание).



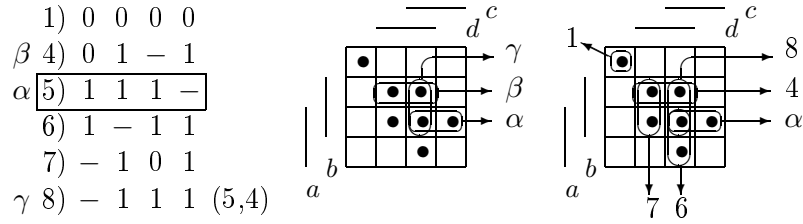
Шаг 3. Вектор γ не поглощается ни одним из предыдущих, поэтому остается в списке, но удаляется второй вектор как поглощаемый вектором γ (см. ниже).



Шаг 4. Так как вектор α не был удален, идем на шаг 2.

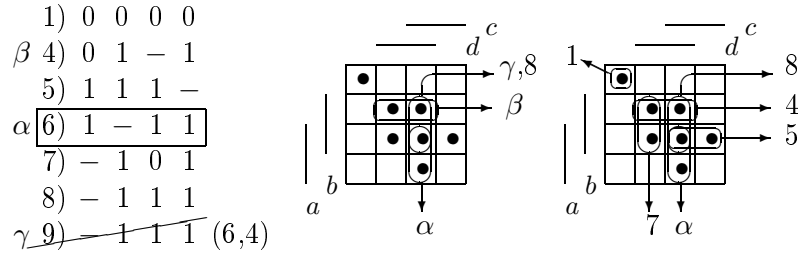
Шаги 2, 5. Так как для вектора α перебраны все предыдущие, то идем на шаг 5. Так как α был не последним в списке, то в качестве нового α рассматриваем пятый вектор (см. следующую страницу) и возвращаемся на шаг 2.

Шаги 2-4. Вектор α не смежен первому, но смежен четвертому вектору β . Обобщенно склеиваем α и β и записываем результат γ в список восьмым. Вектор γ не поглощается ни одним вектором из списка, поэтому остается в нем. Вектор γ не поглощает ни один вектор из списка, в том числе α . Идем на шаг 2.



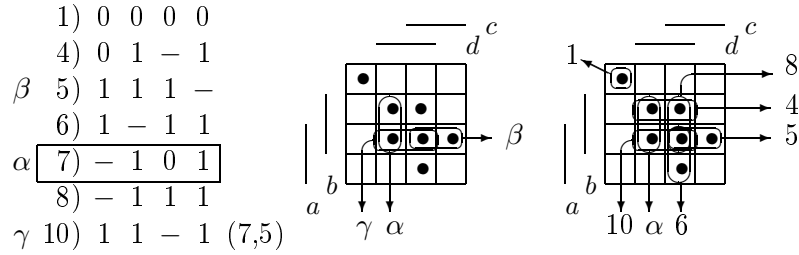
Шаги 2, 5. Для вектора α перебраны все предыдущие, поэтому новым α становится шестой вектор, идем на шаг 2.

Шаги 2-3. Вектор α смежен четвертому вектору β (см. ниже). Обобщенно склеивам α и β и записываем результат γ в список девятым. Вектор γ совпадает с восьмым вектором, то есть поглощается им, поэтому тут же удаляем вектор γ из списка. Идем на шаг 2.

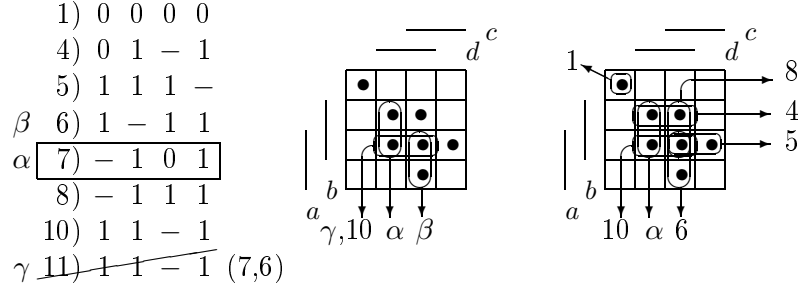


Шаги 2, 5. Для вектора α больше нет смежных среди предыдущих, поэтому новым α будет седьмой вектор, идем на шаг 2.

Шаги 2-4. Вектор α смежен из предыдущих пятому вектору β (см. ниже). Обобщенно склеивам их и записываем результат γ в список десятым. Вектор γ не поглощается векторами списка и не поглощает ни один вектор, поэтому идем на шаг 2.

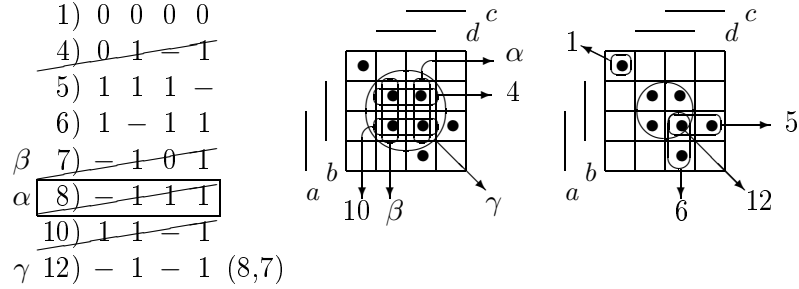


Шаги 2-3. Вектор α смежен шестому вектору β . Обобщенно склеиваем их и записываем результат γ в список одиннадцатым. Вектор γ совпадает с десятым вектором, поэтому тут же удаляем γ из списка. Идем на шаг 2.



Шаги 2, 5. Так как для α перебраны все предыдущие, то в качестве нового α рассматриваем восьмой вектор и возвращаемся на шаг 2.

Шаги 2-4. Вектор α смежен только седьмому вектору β из предыдущих (см. ниже). Обобщенно склеиваем их и записываем результат γ в список двенадцатым. Вектор γ поглощает четвертый, седьмой, восьмой и десятый векторы, поэтому удаляем их из списка. Так как вектор α оказался вычеркнутым, идем на шаг 5.



Шаг 5. Выбираем в качестве нового α двенадцатый вектор и идем на шаг 2.

Шаги 2, 5. Так как для двенадцатого вектора нет смежных среди предыдущих, то идем на шаг 5. Вектор был последним в списке, поэтому идем на конец.

Конец. Невычеркнутые из последнего списка векторы задают сокращенную ДНФ:

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & - \\ 1 & - & 1 & 1 \\ - & 1 & - & 1 \end{array} \quad \text{СокрДНФ} = \bar{a}\bar{b}\bar{c}\bar{d} \vee abc \vee acd \vee bd.$$

Результат совпадает с сокращенной ДНФ, полученной алгоритмом Квайна-МакКласки на странице 9, и это естественно, так как сокращенная ДНФ булевой функции единственна. [2.С.6]

Демонстрируя алгоритм на примере, мы многократно переписывали списки, незначительно изменяя их на каждом шаге. Покажем, как можно ограничиться одним списком, вычеркивая поглощаемые векторы и добавляя новые. Договоримся при вычеркивании вектора сообщать номер поглощающего его вектора.

$$\begin{array}{ll} 1) & 0 \ 0 \ 0 \ 0 \\ 2) & \cancel{1 \ 1 \ 0 \ 1} \quad 7 \\ 3) & \cancel{1 \ 0 \ 1 \ 1} \quad 6 \\ 4) & \cancel{0 \ 1 \ - \ 1} \quad 12 \\ 5) & 1 \ 1 \ 1 \ - \\ 6) & 1 \ - \ 1 \ 1 \\ 7) & \cancel{- \ 1 \ 0 \ 1} \ (4,2) \quad 12 \\ 8) & \cancel{- \ 1 \ 1 \ 1} \ (5,4) \quad 12 \\ 9) & \cancel{- \ 1 \ 1 \ 1} \ (6,4) \quad 8 \\ 10) & \cancel{1 \ 1 \ - \ 1} \ (7,5) \quad 12 \\ 11) & \cancel{1 \ 1 \ - \ 1} \ (7,6) \quad 10 \\ 12) & - \ 1 \ - \ 1 \ (8,7) \end{array}$$

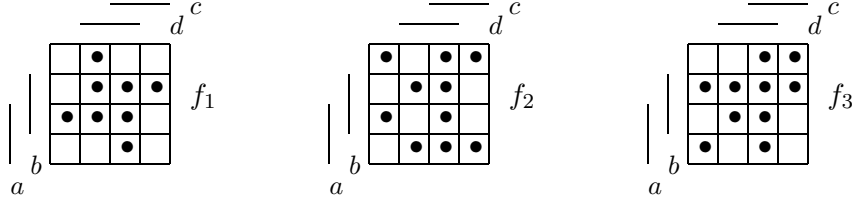
Оставшиеся в списке векторы образуют решение. •

Итак, нами изучен первый этап минимизации булевых функций – построение сокращенной ДНФ:

- из совершенной ДНФ (алгоритм Квайна-МакКласки);
- из произвольной ДНФ (алгоритм Блейка-Порецкого).

11.1.3. Упражнения

Упр. 1. Получить сокращенные ДНФ булевых функций алгоритмом Квайна-МакКласки и сравнить их с сокращенными ДНФ, найденными визуально.



$$f_4(a, b, c, d) = \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}cd \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee \bar{a}b\bar{c}d;$$

$$f_5(a, b, c, d) = \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d;$$

$$f_6(a, b, c, d) = \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}b\bar{c}d;$$

Упр. 2. Получить сокращенные ДНФ булевых функций алгоритмом Блейка-Порецкого и сравнить их с сокращенными ДНФ, найденными визуально.

$$f_1(a, b, c, d) = \bar{a}b\bar{c} \vee ac\bar{d} \vee a\bar{b}\bar{c}d \vee \bar{a}bcd \vee \bar{a}cd \vee bcd;$$

$$f_2(a, b, c, d) = \bar{a}\bar{c}d \vee \bar{a}bcd \vee \bar{a}\bar{b}cd \vee ac\bar{d} \vee a\bar{b}d \vee \bar{a}\bar{b}c;$$

$$f_3(a, b, c, d) = \bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{c}d \vee \bar{a}bcd \vee \bar{a}\bar{b}cd \vee a\bar{b}c \vee \bar{a}b\bar{c}.$$

Упр. 3. Получить сокращенные ДНФ булевых функций алгоритмами Квайна-МакКласки и Блейка-Порецкого и сравнить результаты (ДНФ для второго алгоритма найти любым известным способом). Проверить, представляют ли полученные сокращенные ДНФ исходные функции, построением таблиц истинности.

$$f_1(x, y, z, t) = x \oplus y\bar{t} \rightarrow (x \leftrightarrow z) \rightarrow \bar{x}\bar{t} \oplus \bar{x}\bar{y}zt;$$

$$f_2(x, y, z, t) = xyz \vee (y \rightarrow xt \rightarrow (y \oplus x) \rightarrow y\bar{t});$$

$$f_3(x, y, z, t) = xt \downarrow y \rightarrow (xt \sim \bar{z}) \rightarrow x\bar{t} \vee xyz.$$

11.2. Построение таблицы Квайна и поиск ее покрытий – второй этап минимизации

Решая задачу минимизации булевых функций и научившись на первом этапе строить сокращенную ДНФ (множество всех максимальных интервалов), перейдем к рассмотрению второго этапа. На этом этапе из сокращенной ДНФ будем получать кратчайшие или минимальные ДНФ (из множества всех максимальных интервалов будем выбирать достаточные подмножества либо из минимального числа интервалов, либо из интервалов с минимальной суммой рангов).

11.2.1. Таблица Квайна

Определение. *Булевой матрицей* назовем матрицу, элементами которой являются булевы константы (другими словами, строками и столбцами которой являются булевы векторы).

Прежде всего рассмотрим, как в явном виде показать, используя булеву матрицу, из каких точек функции состоят ее максимальные интервалы.

Определение. *Таблицей Квайна* булевой функции $f(x_1, \dots, x_n)$ назовем булеву матрицу, строкам которой поставим в соответствие конъюнкции сокращенной ДНФ f (все максимальные интервалы функции), а столбцам – конъюнкции совершенной ДНФ f (все точки). Элемент матрицы, стоящий на пересечении i -й строки и j -го столбца, положим равным единице тогда и только тогда, когда i -я конъюнкция сокращенной ДНФ поглощает j -ю конъюнкцию совершенной ДНФ (i -й максимальный интервал функции содержит ее j -ю точку).

Пример. Рассмотрим булеву функцию $f(a, b, c, d)$, для которой в [2.С.22–24] визуальным образом были найдены различные ДНФ. Построим для данной функции таблицу Квайна (используя ее в дальнейшем для минимизации, получим кратчайшие и минимальные ДНФ на странице 26 и сравним их с найденными ранее). Для построения таблицы Квайна точки функции пронумеруем на левой матрице, а все максимальные интервалы изобразим на остальных двух матрицах.

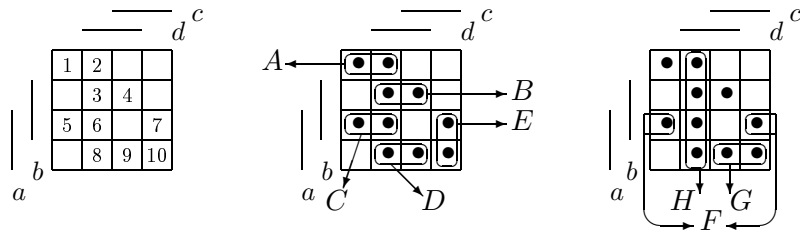


Таблица Квайна данной функции имеет следующий вид (нули в ней для наглядности опущены):

	0000	0001	0101	0111	1100	1101	1110	1001	1011	1010	
000 –	1	1									A
01 – 1			1	1							B
110 –					1	1					C
10 – 1								1	1		D
1 – 10							1			1	E
11 – 0					1		1				F
101 –									1	1	G
– – 01		1	1			1		1			H
	1	2	3	4	5	6	7	8	9	10	

Обозначим полученную таблицу через Q , и будем демонстрировать на ней материал следующих разделов. •

11.2.2. Покрывтия таблицы Квайна и ДНФ

Определение. Будем говорить, что строка таблицы Квайна *покрывает* столбец, если она содержит единицу в этом столбце.

Пример. Строка H покрывает второй, третий, шестой и восьмой столбцы таблицы Q . •

Определение. *Покровтием* таблицы Квайна назовем подмножество таких ее строк, которые в совокупности покрывают все столбцы таблицы.

Примеры. Покровтиями таблицы Q являются: все восемь ее строк; семь строк: A, B, C, D, F, G, H ; первые шесть строк; первые пять строк; однако не удастся найти ни одного четырехстрочного покрытия. •

Любое покрытие таблицы Квайна функции $f(x_1, \dots, x_n)$ задает подмножество максимальных интервалов, достаточное для функции, так как каждая точка функции принадлежит хотя бы одному из ее максимальных интервалов. Другими словами, *любое покрытие таблицы Квайна задает ДНФ f* . В частности, покрытие, состоящее из всех строк таблицы, задает сокращенную ДНФ.

Определение. *Длиной покрытия* назовем количество строк, образующих покрытие.

Примеры. Длины покрытий из предыдущего примера равны соответственно 8, 7, 6 и 5. •

Определение. Покрытие таблицы Квайна называется *безызбыточным*, если при удалении из него хотя бы одной строки оно перестает быть покрытием.

Примеры. Множество из первых шести строк таблицы Квайна Q не является безызбыточным покрытием, поскольку при удалении шестой строки оно остается покрытием. Первые пять строк образуют безызбыточное покрытие, и оно не единственное – шесть строк A, B, C, D, F, G тоже образуют безызбыточное покрытие. •

В общем случае таблица Квайна функции $f(x_1, \dots, x_n)$ имеет несколько безызбыточных покрытий, и *любое безызбыточное покрытие задает безызбыточную ДНФ f* , так как из нее нельзя выбросить ни одной конъюнкции (из безызбыточного покрытия нельзя выбросить ни одной строки) и ни одной буквы из какой-либо конъюнкции (строкам приписаны простые импликанты) так, чтобы полученная ДНФ оставалась равносильной исходной.

Определение. *Кратчайшим* покрытием таблицы Квайна назовем покрытие минимальной длины.

Примеры. Первые пять строк таблицы Квайна Q , а также строки A, B, F, G, H образуют ее кратчайшие покрытия. •

Очевидно, что *каждое кратчайшее покрытие таблицы Квайна функции $f(x_1, \dots, x_n)$ задает кратчайшую ДНФ f* .

Определение. *Рангом строки* таблицы Квайна назовем ранг приписанной ей простой импликанты.

Пример. Ранг строки H (конъюнкции $\bar{c}d$) таблицы Квайна Q равен двум, ранги остальных строк равны трем. •

Определение. Минимальным покрытием таблицы Квайна назовем покрытие с минимальной суммой рангов строк.

Пример. Строки A, B, F, G, H таблицы Квайна Q образуют минимальное покрытие, так как сумма их рангов равна 14, а все остальные покрытия имеют больший суммарный ранг строк. •

Нетрудно видеть, что минимальное покрытие таблицы Квайна функции $f(x_1, \dots, x_n)$ задает минимальную ДНФ f .

Таким образом, задача минимизации булевой функции решается построением таблицы Квайна и поиском ее кратчайших или минимальных покрытий (в зависимости от постановки задачи).

Отвлечемся на время от задачи минимизации и покажем, что совсем иные задачи сводятся к поиску тех или иных покрытий булевых матриц.

Задача 1. Пусть в некотором устройстве возможны 10 неисправностей, и пусть имеется 8 тестов, каждый из которых обнаруживает свое подмножество неисправностей:

Тесты:	A	B	C	D	E	F	G	H
Неисправности:	1, 2	3, 4	5, 6	8, 9	7, 10	5, 7	9, 10	2, 3, 6, 8

Требуется подобрать кратчайший полный тест, то есть набор из минимального числа тестов, обнаруживающих все неисправности.

Построим матрицу, строкам которой припишем тесты, а столбцам – неисправности. Элемент матрицы, стоящий на пересечении i -й строки и j -го столбца, положим равным единице тогда и только тогда, когда i -й тест обнаруживает j -ю неисправность.

1	1									A
		1	1							B
				1	1					C
							1	1		D
						1			1	E
				1		1				F
								1	1	G
	1	1			1		1			H
1	2	3	4	5	6	7	8	9	10	

Матрица полностью совпадает с таблицей Q , а ее кратчайшее покрытие является решением задачи о кратчайшем полном тесте.

Пусть теперь проверка каждого теста требует определенных временных затрат, например, тест H требует двух единиц времени, а остальные – трех. Тогда имеет смысл искать минимальный полный тест, то есть набор тестов, который обнаруживает все неисправности за минимальное время. Решение задачи сводится к поиску минимального покрытия той же матрицы, если рангом строки считать временные затраты на проверку теста.

Задача 2. Имеется группа переводчиков с иностранных языков на русский. О каждом переводчике известно, какими иностранными языками он владеет. Требуется выбрать минимальное число переводчиков, знающих в совокупности все языки. И снова мы приходим к мысли о построении булевой матрицы и поиске ее кратчайшего покрытия.

Учитывая это, будем абстрагироваться от задач, породивших матрицы, и говорить лишь об их строках, столбцах и элементах.

11.2.3. Поиск всех безызбыточных покрытий

Очевидно, что кратчайшие и минимальные покрытия булевой матрицы являются безызбыточными. Поэтому поиск всех кратчайших и минимальных покрытий можно свести к построению всех безызбыточных покрытий и выделению из них кратчайших и минимальных. Тривиальный метод поиска всех безызбыточных покрытий состоит в переборе и анализе всех подмножеств строк матрицы. Однако, количество перебираемых подмножеств велико (равно 2^m , где m – число строк матрицы), поэтому остановимся на другом, более эффективном методе.

Будем предполагать, что матрица имеет покрытие, то есть не содержит столбцов, целиком состоящих из нулей (для таблицы Квайна это условие всегда выполняется), и прежде всего попытаемся упростить матрицу.

Определение. Столбец булевой матрицы назовем *ядерным*, если он содержит ровно одну единицу. Строку булевой матрицы назовем *ядерной*, если она покрывает ядерный столбец.

Пример. В матрице Q на странице 18 ядерными являются столбцы 1, 4 и строки A, B . •

Очевидно, что ядерные строки входят в любое покрытие, что дает возможность сформулировать следующее правило упрощения матрицы.

Правило ядерной строки. Если в матрице есть ядерная строка, то она включается в любое искомое покрытие и удаляется из матрицы вместе со всеми столбцами, которые ядерная строка покрывает. Удаляются также пустые строки (целиком состоящие из нулей), которые при этом могут появиться.

Пример. Применим правило ядерной строки дважды к знаковой нам матрице Q . В результате строки A и B будут включены в покрытие, а матрица упростится.

$$Q' = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 1 & & & & \\ \hline & & & 1 & 1 & \\ \hline & & 1 & & & 1 \\ \hline 1 & & 1 & & & \\ \hline & & & & 1 & 1 \\ \hline & 1 & & 1 & & \\ \hline \end{array} \begin{array}{l} C \\ D \\ E \\ F \\ G \\ H \end{array} \bullet$$

$$\begin{array}{cccccc} 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

В частности, если все столбцы матрицы покрываются ядерными строками, то эти строки образуют единственное безызбыточное покрытие матрицы.

Упростив матрицу по правилу ядерной строки, и оставив по одному из равных столбцов (что, очевидно, не повлияет на решение), продолжим поиск всех безызбыточных покрытий упрощенной матрицы.

Припишем ее строкам булевы переменные s_1, \dots, s_m , обозначим через S произвольное подмножество строк и положим $s_i = 1$, если и только если i -я строка принадлежит множеству S , то есть будем задавать подмножество S строк матрицы набором σ значений переменных s_1, \dots, s_m .

Определение. Булеву функцию $p(s_1, \dots, s_m)$, принимающую значение единицы на тех и только тех наборах σ , которые задают покрытия матрицы, назовем *функцией покрытия матрицы*.

Пример. Зададим формулой функцию покрытия предыдущей матрицы Q' , исходя из следующих рассуждений.

Первый столбец покрывается подмножеством строк S , если и только если в S входят:

- либо строка C (переменная $C = 1$),
 - либо строка F (переменная $F = 1$),
 - либо обе строки одновременно ($C = 1$ и $F = 1$),
- то есть если и только если $C \vee F = 1$.

Аналогично второй столбец покрывается подмножеством S , если и только если $C \vee H = 1$, и так далее для всех столбцов.

Все столбцы покрываются подмножеством строк S , то есть $p(\sigma) = 1$, если и только если одновременно выполняются все перечисленные условия, то есть $(C \vee F)(C \vee H) \dots = 1$. Это означает, что функцию покрытия матрицы Q' можно задать формулой:

$$p(C, \dots, H) = (C \vee F)(C \vee H)(E \vee F)(D \vee H)(D \vee G)(E \vee G). \bullet$$

В общем случае для матрицы с k столбцами функция покрытия задается конъюнкцией вида $D_1 \cdot \dots \cdot D_k$, где D_j – дизъюнкция всех переменных, приписанных строкам с единицей в j -м столбце. Будем называть полученную формулу *конъюнктивной нормальной формой (КНФ)* функции покрытия:

$$p(s_1, \dots, s_m) = D_1 \cdot \dots \cdot D_k = \text{КНФ}_p.$$

Так как каждое слагаемое дизъюнкции D_j задает одно из всех безызбыточных покрытий j -го столбца, то, перемножив пару дизъюнкций D_i и D_j , получим все безызбыточные покрытия пары столбцов i и j (но не только безызбыточные).

Пример. Перемножим первые две дизъюнкции из предыдущего примера:

$$(C \vee F)(C \vee H) = C \vee CH \vee CF \vee FH.$$

Получены не только все безызбыточные покрытия первых двух столбцов: $\{C\}$ и $\{F, H\}$, но и покрытия $\{C, H\}$ и $\{C, F\}$, которые содержат строку C , поэтому не являются безызбыточными (это эквивалентно тому, что конъюнкции CH и CF поглощаются конъюнкцией C). •

Обобщая приведенные рассуждения на все столбцы булевой матрицы, приходим к выводу, что для построения всех безызбыточных покрытий матрицы надо перемножать все дизъюнкции КНФ функции покрытия, выполняя при этом все возможные поглощения. В результате будет получена ДНФ, конъюнкции которой задают все безызбыточные покрытия матрицы (покрытие состоит из строк, которым приписаны переменные, образующие конъюнкцию).

Алгоритм поиска всех безызбыточных покрытий булевой матрицы (основан на построении КНФ функции покрытия и преобразовании ее в ДНФ с выполнением поглощений).

Начало. Задана булева матрица.

Шаг 1. Применим к матрице правило ядерной строки многократно, пока это возможно. Если матрица стала пустой, то идем на конец. Иначе из равных столбцов оставим по одному, пронумеруем столбцы упрощенной матрицы и припишем строкам булевы переменные s_1, \dots, s_m .

Шаг 2. Выберем очередной (j -й) столбец матрицы. Если столбцы исчерпаны, идем на шаг 3. Иначе построим дизъюнкцию D_j переменных, приписанных строкам, содержащим единицы в j -м столбце, и возвратимся на шаг 2.

Шаг 3. Из полученных дизъюнкций построим конъюнктивную нормальную форму $D_1 \cdot \dots \cdot D_k$.

Шаг 4. Раскроем скобки в КНФ, выполняя поглощения конъюнкций. Получим ДНФ $= K_1 \vee \dots \vee K_t$.

Конец. Конъюнкции K_1, \dots, K_t задают все безызбыточные покрытия упрощенной на первом шаге матрицы. Добавив к каждому из них ядерные строки (выделенные на первом шаге), получим все безызбыточные покрытия исходной матрицы.

Пример. Продемонстрируем алгоритм на знакомой нам булевой матрице Q со страницы 18. Для нее уже выполнен шаг 1 (страница 22): строки A и B выделены как ядерные и удалены вместе с первыми четырьмя столбцами. В результате (страница 23) получена упрощенная матрица и построена КНФ функции ее покрытия (шаг 2 повторен 6 раз и выполнен шаг 3):

1	1					C
			1	1		D
		1			1	E
1		1				F
				1	1	G
	1		1			H
5	6	7	8	9	10	

$$\text{КНФ} = (C \vee F)(C \vee H)(E \vee F)(D \vee H)(D \vee G)(E \vee G).$$

Шаг 4. Перемножив скобки в КНФ (удобнее перемножать первую на вторую, третью на шестую и четвертую на пятую скобки) и выполнив поглощения, получим ДНФ.

$$\begin{aligned} & (C \vee FH)(E \vee FG)(D \vee GH) = \\ & = (CE \vee CFG \vee EFH \vee FGH)(D \vee GH) = \\ & = CDE \vee CEGH \vee CDFG \vee CFGH \vee \\ & \quad \vee DEFH \vee EFGH \vee DFGH \vee FGH = \\ & = CDE \vee CEGH \vee CDFG \vee DEFH \vee FGH = \text{ДНФ}. \end{aligned}$$

Конец. Конъюнкции ДНФ задают все безызбыточные покрытия упрощенной матрицы: $\{C, D, E\}$, $\{C, E, G, H\}$, $\{D, E, F, H\}$, $\{C, D, F, G\}$ и $\{F, G, H\}$.

Добавив к каждому из них ядерные строки A и B , выделенные на первом шаге, получим все безызбыточные покрытия исходной матрицы (два покрытия длины 5 и три покрытия длины 6):

$$\begin{aligned} & \{A, B, C, D, E\}, \{A, B, F, G, H\} \\ & \{A, B, C, E, G, H\}, \{A, B, D, E, F, H\}, \{A, B, C, D, F, G\}. \bullet \end{aligned}$$

11.2.4. Поиск минимальных и кратчайших покрытий

Зная все безызбыточные покрытия матрицы, выбрать из них кратчайшие или минимальные не составляет труда.

Пример. Из пяти найденных в предыдущем примере безызбыточных покрытий матрицы Q только два являются кратчайшими: $\{A, B, C, D, E\}$ и $\{A, B, F, G, H\}$, а минимальным лишь одно – $\{A, B, F, G, H\}$, так как ранг строки H равен двум, а ранги остальных строк – трем. Эти покрытия задают те же две кратчайшие и одну минимальную ДНФ функции $f(a, b, c, d)$ со страницы 18, которые были получены в [2.С.22] визуально. •

Однако поиск всех безызбыточных покрытий булевой матрицы становится излишне трудоемким, если требуется найти одно кратчайшее покрытие. В этом случае можно ограничиться поиском только некоторых безызбыточных покрытий, лишь бы среди них содержалось хотя бы одно кратчайшее. И тогда применимы следующие два правила упрощения булевой матрицы.

Правило строки-предшественницы. Если в булевой матрице есть такие строки α и β , что $\alpha \preceq \beta$, то строка-предшественница α удаляется из матрицы (при этом кратчайшее покрытие не теряется, так как в любом покрытии строку α можно заменить строкой β , и оно останется покрытием).

Правило столбца-последователя. Если в булевой матрице есть такие столбцы γ и δ , что $\gamma \preceq \delta$, то столбец-последователь δ удаляется из матрицы (так как любое покрытие столбца γ одновременно покрывает столбец δ).

Заметим, что удаление столбцов-последователей может привести к появлению строк-предшественниц, и наоборот, поэтому эти два правила стоит выполнять "циклически" друг за другом.

Определение. Булеву матрицу, не содержащую ни ядерных строк, ни строк-предшественниц, ни столбцов-последователей, назовем *циклическим остатком*.

Примеры к перечисленным правилам и к циклическому остатку будут приведены чуть позже, на страницах 27–29. •

Алгоритм поиска одного кратчайшего покрытия булевой матрицы (основан на построении циклического остатка и поиске всех его безызбыточных покрытий).

Начало. Задана булева матрица.

Шаг 1. Если в матрице есть однострочное покрытие, то включим его в покрытие, идем на конец.

Шаг 2. Применим правило ядерной строки. Если ядерная строка обнаружилась, идем на шаг 1.

Шаг 3. Применим правило строки-предшественницы. Если такая строка обнаружилась, идем на шаг 2.

Шаг 4. Применим правило столбца-последователя. Если такой столбец обнаружился, идем на шаг 3.

Шаг 5. Построим КНФ функции покрытия полученного циклического остатка и преобразуем КНФ в ДНФ, раскрывая скобки и выполняя поглощения.

Шаг 6. Конъюнкция минимального ранга полученной ДНФ задает кратчайшее покрытие циклического остатка. Добавим к этому покрытию ядерные строки, выделенные на шаге 2.

Конец. Получено кратчайшее покрытие исходной матрицы.

Примеры. Все та же матрица Q со страницы 18 сократилась до циклического остатка сразу после удаления ядерных строк (страница 25), что не дает возможности продемонстрировать шаги 3 и 4 алгоритма. Рассмотрим в связи с этим другую матрицу – Q'' .

$$Q'' = \begin{array}{|c|c|c|c|c|c|c|c|} \hline & 1 & & 1 & & 1 & & \\ \hline & 1 & 1 & & & & & \\ \hline 1 & & & 1 & & 1 & & \\ \hline 1 & & & & & & 1 & 1 \\ \hline 1 & & & 1 & & & 1 & \\ \hline & & & & 1 & 1 & & 1 \\ \hline & & & & 1 & & 1 & 1 \\ \hline \end{array} \begin{array}{l} A \\ B \\ C \\ D \\ E \\ F \\ G \end{array}$$

1 2 3 4 5 6 7 8

Шаги 1, 2. Матрица Q'' не имеет однострочного покрытия, но в ней есть ядерная строка B . Включаем ее в покрытие и удаляем

из матрицы вместе со вторым и третьим столбцами, которые она покрывает. Получаем матрицу Q_1'' и идем на шаг 1.

$$Q_1'' = \begin{array}{|c|c|c|c|c|c|} \hline & 1 & & 1 & & \\ \hline 1 & 1 & & 1 & & \\ \hline 1 & & & & 1 & 1 \\ \hline 1 & 1 & & & 1 & \\ \hline & & 1 & 1 & & 1 \\ \hline & & 1 & & 1 & 1 \\ \hline \end{array} \begin{array}{l} A \\ C \\ D \\ E \\ F \\ G \end{array}$$

1 4 5 6 7 8

Шаги 1–3. В матрице Q_1'' нет однострочного покрытия и ядерной строки, но есть строка-предшественница: $A \preceq C$. Удаляем строку A , получаем матрицу Q_2'' . Идем на шаг 2.

Шаги 2–4. В матрице Q_2'' нет ядерной строки и строки-предшественницы, но первый столбец следует за четвертым. Удаляем первый столбец, получаем матрицу Q_3'' . Идем на шаг 3.

$$Q_2'' = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 1 & & 1 & & \\ \hline 1 & & & & 1 & 1 \\ \hline 1 & 1 & & & 1 & \\ \hline & & 1 & 1 & & 1 \\ \hline & & 1 & & 1 & 1 \\ \hline \end{array} \begin{array}{l} C \\ D \\ E \\ F \\ G \end{array}$$

1 4 5 6 7 8

$$Q_3'' = \begin{array}{|c|c|c|c|c|} \hline 1 & & 1 & & \\ \hline & & & 1 & 1 \\ \hline 1 & & & 1 & \\ \hline & 1 & 1 & & 1 \\ \hline & 1 & & 1 & 1 \\ \hline \end{array} \begin{array}{l} C \\ D \\ E \\ F \\ G \end{array}$$

4 5 6 7 8

Шаги 3–4. В матрице Q_3'' нет строки-предшественницы, но восьмой столбец следует за пятым. Удаляем восьмой столбец, получаем матрицу Q_4'' . Идем на шаг 3.

Шаг 3. В матрице Q_4'' появилась строка-предшественница: $D \preceq E$. Удалив D , получаем матрицу Q_5'' и идем на шаг 2.

$$Q_4'' = \begin{array}{|c|c|c|c|} \hline 1 & & 1 & \\ \hline & & & 1 \\ \hline 1 & & & 1 \\ \hline & 1 & 1 & \\ \hline & 1 & & 1 \\ \hline \end{array} \begin{array}{l} C \\ D \\ E \\ F \\ G \end{array}$$

4 5 6 7

$$Q_5'' = \begin{array}{|c|c|c|c|} \hline 1 & & 1 & \\ \hline 1 & & & 1 \\ \hline & 1 & 1 & \\ \hline & 1 & & 1 \\ \hline \end{array} \begin{array}{l} C \\ E \\ F \\ G \end{array}$$

4 5 6 7

Шаги 2–5. В матрице Q_5'' нет ни ядерных строк, ни строк-предшественниц, ни столбцов-последователей, значит, получен циклический остаток. Строим КНФ функции его покрытия:

$$\text{КНФ} = (C \vee E)(F \vee G)(C \vee F)(E \vee G) =$$

[преобразуем КНФ в ДНФ: для удобства умножаем первую скобку на третью, вторую на четвертую и выполняем поглощения]

$$= (C \vee EF)(G \vee EF) = CG \vee EF = \text{ДНФ}.$$

Шаг 6. Обе конъюнкции имеют минимальный ранг, поэтому выбираем любую из них, CG , она задает кратчайшее покрытие циклического остатка. Добавив ядерную строку B , получаем кратчайшее покрытие исходной булевой матрицы: $\{B, C, G\}$. •

Итак, нами изучен второй этап минимизации булевых функций, на котором из сокращенной ДНФ получаются либо все минимальные и кратчайшие, либо одна кратчайшая ДНФ.

11.2.5. Упражнения

Упр. 1. Найти все безызбыточные покрытия матрицы Q'' со страницы 27.

В упражнениях 2–3 применять двухэтапный метод минимизации (сокращенную ДНФ находить любым алгоритмом).

Упр. 2. Получить все кратчайшие и минимальные ДНФ функций $f_4 - f_6$ из упр.1 на странице 16 и следующих функций:

$$f_1(a, b, c) = a \oplus b / (b \sim \bar{c});$$

$$f_2(a, b, c, d) = \bar{b}\bar{c}\bar{d} \vee a\bar{c}d \vee b\bar{c} \vee abc \vee \bar{a}\bar{b}c\bar{d} \vee cd \vee a\bar{b}c.$$

$$f_3(a, b, c, d) = \bar{a}\bar{b}\bar{c}\bar{d} \vee b\bar{c}d \vee \bar{a}b\bar{c} \vee abcd \vee a\bar{b}\bar{c} \vee \bar{a}\bar{b}c\bar{d}.$$

Упр. 3. Получить одну кратчайшую ДНФ булевых функций:

$$f_1(a, b, c, d) = \bar{a}\bar{b}\bar{d} \vee \bar{a}b\bar{c}\bar{d} \vee bd \vee a\bar{b}\bar{c}\bar{d} \vee ac\bar{d} \vee \bar{a}\bar{b}c \vee a\bar{c}d;$$

$$f_2(a, b, c, d) = \bar{a}\bar{b}\bar{c}d \vee abc \vee a\bar{b}d \vee \bar{a}\bar{c}\bar{d} \vee c\bar{d} \vee b\bar{c}d;$$

$$f_3(a, b, c, d) = \bar{a}\bar{c}\bar{d} \vee a\bar{b}d \vee \bar{a}bc \vee c\bar{d} \vee b\bar{c}d \vee \bar{a}\bar{b}cd.$$

11.3. Приближенная кратчайшая ДНФ

Рассмотренный в предыдущих разделах двухэтапный метод минимизации булевых функций оказывается довольно трудоемким как на первом, так и на втором этапах. Однако существуют ситуации, в которых имеет смысл согласиться с "приближенным" решением задачи минимизации, то есть искать не обязательно оптимальные решения, но достаточно близкие к ним, лишь бы эти решения находились существенно быстрее, чем оптимальные. Естественно, что при этом приходится идти на компромисс между точностью решения и временем его поиска. Алгоритмы, ориентированные на быстрый поиск приближенных решений, принято называть приближенными.

Определение. Назовем *приближенной кратчайшей ДНФ* (*ПриКратДНФ_f*) булевой функции $f(x_1, \dots, x_n)$ такую ее ДНФ, которая не обязательно является кратчайшей, но достаточно близка к ней по длине.

Заметим, что любую ДНФ (в частности, безызбыточную) можно считать приближенной кратчайшей, так как в определении не уточняется, что значит "достаточно близка" по длине.

Пример. Найдем две приближенные кратчайшие ДНФ одной и той же булевой функции.

$\text{ПриКратДНФ}' = \bar{a}cd \vee \bar{b}c\bar{d} \vee bd \vee abc,$

$\text{ПриКратДНФ}'' = \bar{a}\bar{b}c \vee bd \vee ac\bar{d}.$

Вторая из найденных приближенных кратчайших ДНФ оказывается к тому же кратчайшей. •

11.3.1. Алгоритм Закревского

Существуют различные методы поиска приближенных кратчайших ДНФ, но мы остановимся лишь на одном из них. Как показало его тестирование на функциях небольшого числа аргументов, метод находит чаще всего либо кратчайшую ДНФ, либо отличающуюся от кратчайшей на одну конъюнкцию.

Алгоритм Закревского (ориентирован на матричное представление функции и визуальный способ решения).

Начало. Задана матрица Грея булевой функции.

Шаг 1. Для каждой точки вычислим цену – количество соседних ей точек. Все точки будем считать неотмеченными.

Шаг 2. Среди неотмеченных точек рассмотрим точку минимальной цены и найдем все максимальные интервалы, которым она принадлежит. Выберем из них тот максимальный интервал, который содержит наибольшее число неотмеченных точек. Если таких интервалов несколько, то выберем из них интервал максимальной мощности.

Шаг 3. Включим выбранный интервал в решение и отметим на матрице его точки. Если не все точки отмечены, то идем на шаг 2.

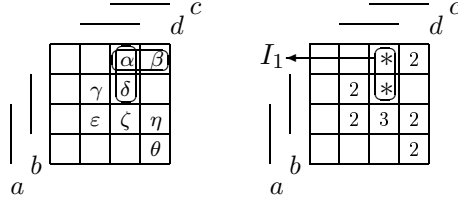
Конец. Включенные в решение интервалы задают приближенную кратчайшую ДНФ.

Пример. Продемонстрируем выполнение алгоритма на матрице Грея из предыдущего примера.

Начало. Точки булевой функции обозначаем греческими буквами (левая матрица).

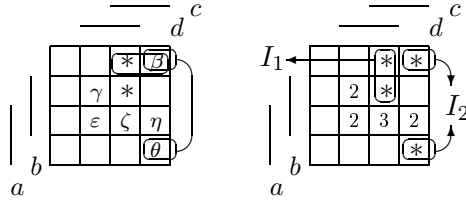
Шаг 1. Вычисляем цены точек (правая матрица).

Шаги 2, 3. Выбираем точку α (минимальной цены 2). Она входит в два максимальных интервала (левая матрица). Оба интервала содержат по две неотмеченные точки и имеют одинаковые мощности, поэтому включаем в решение любой из них и отмечаем его точки (правая матрица).



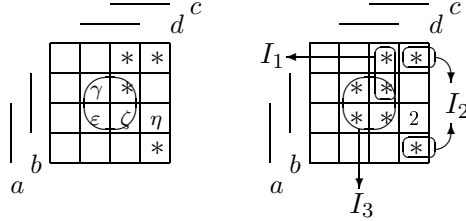
Поскольку не все точки отмечены, идем на шаг 2.

Шаги 2, 3. Выбираем неотмеченную точку β (цены 2). Она входит в два максимальных интервала (с одной и двумя неотмеченными точками), поэтому включаем в решение интервал I_2 с двумя неотмеченными точками и отмечаем их.



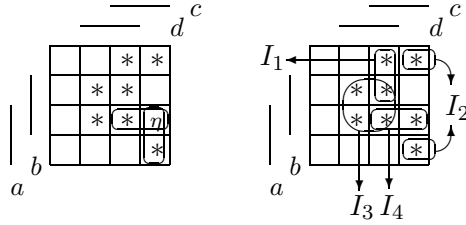
Поскольку не все точки отмечены, идем на шаг 2.

Шаги 2, 3. Выбираем неотмеченную точку γ (цены 2). Она входит в один максимальный интервал, включаем его в решение и отмечаем точки.



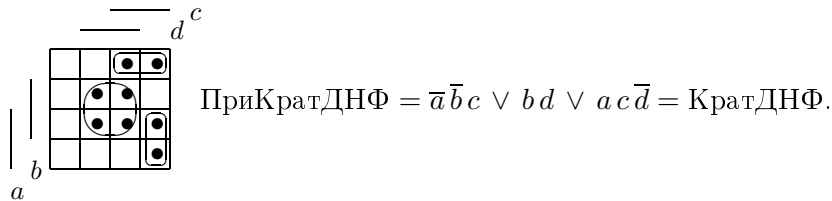
Поскольку не все точки отмечены, идем на шаг 2.

Шаги 2, 3. Выбираем неотмеченную точку η (цены 2). Она входит в два максимальных интервала, содержащих по одной неотмеченной точке. Мощности интервалов равны, поэтому включаем в решение любой интервал и отмечаем его точки. Поскольку все точки функции отмечены, идем на конец.



Конец. ПриКратДНФ = $\bar{a}cd \vee \bar{b}c\bar{d} \vee bd \vee abc$.

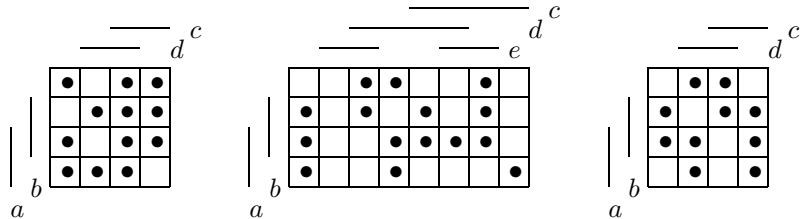
Полученная ДНФ не является кратчайшей (см. страницу 30), но кратчайшая ДНФ могла быть получена данным алгоритмом, если бы при первом выполнении шага 2 в решение был включен другой максимальный интервал:



Рассмотренный пример еще раз подчеркивает приближенный характер алгоритма Закревского. •

11.3.2. Упражнения

Найти приближенные кратчайшие ДНФ булевых функций.



11.4. Контрольная работа

Тема контрольной работы: минимизация булевых функций.

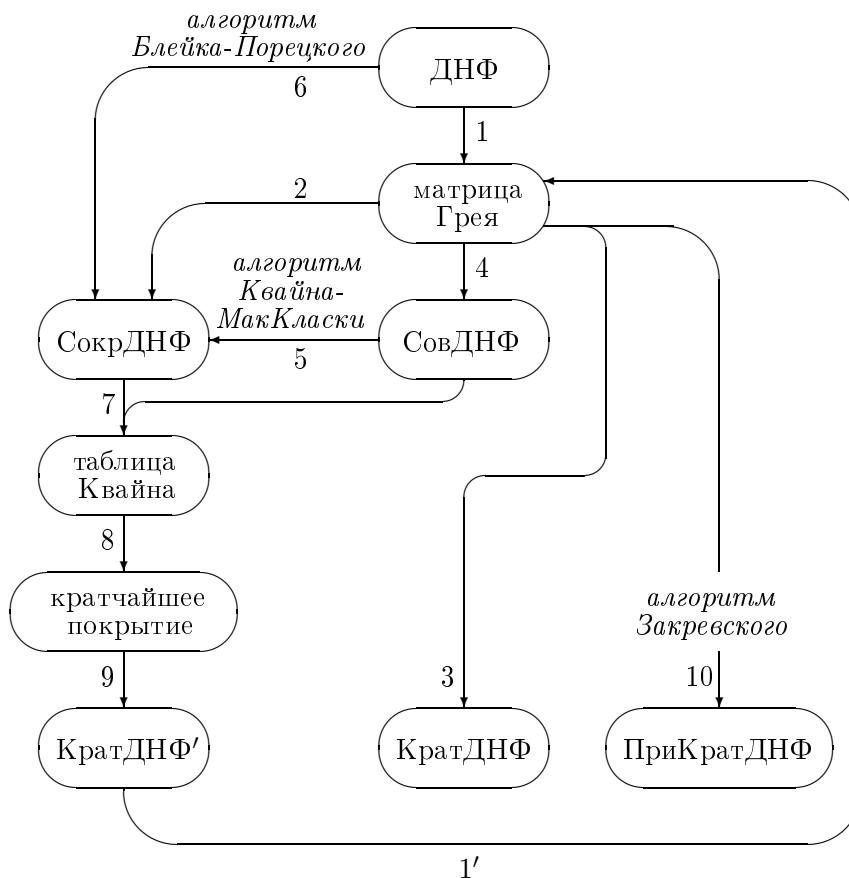


Схема контрольной работы (решение каждой из 11 задач начинать с постановки задачи и делать выводы из сравнения полученных разными способами:

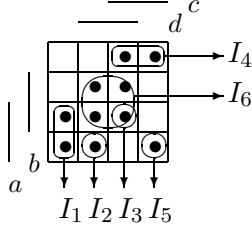
- сокращенных ДНФ;
- матриц Грея;
- кратчайших и приближенной кратчайшей ДНФ).

Задания на контрольную работу (ДНФ)

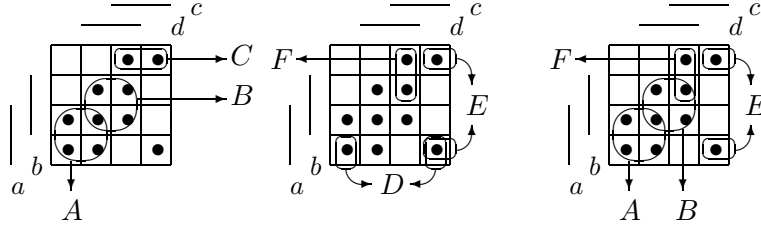
- 1) $abcd \vee \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee \bar{a}c\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{b}c\bar{d} \vee a\bar{b}\bar{c}\bar{d}$
- 2) $\bar{a}b\bar{c} \vee \bar{a}c\bar{d} \vee b\bar{c}\bar{d} \vee abcd \vee acd \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}cd$
- 3) $abcd \vee a\bar{c}\bar{d} \vee \bar{a}c\bar{d} \vee \bar{a}bd \vee \bar{a}\bar{b}c\bar{d} \vee a\bar{b}c\bar{d} \vee a\bar{b}\bar{c}$
- 4) $\bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}b\bar{c} \vee \bar{a}cd \vee a\bar{b}c\bar{d} \vee \bar{a}\bar{b}c \vee ac\bar{d} \vee a\bar{b}\bar{c}\bar{d}$
- 5) $\bar{a}\bar{c}\bar{d} \vee \bar{a}cd \vee bcd \vee abc\bar{d} \vee \bar{a}\bar{b}c\bar{d} \vee ab\bar{c}\bar{d} \vee \bar{b}c\bar{d}$
- 6) $a\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}bc\bar{d} \vee abd \vee \bar{a}\bar{b}\bar{c} \vee ac\bar{d} \vee a\bar{c}\bar{d}$
- 7) $\bar{a}\bar{c}\bar{d} \vee \bar{a}bd \vee a\bar{c}\bar{d} \vee a\bar{b}cd \vee \bar{a}bc\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{c}$
- 8) $\bar{a}\bar{b}\bar{c} \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}bc \vee \bar{a}bc\bar{d} \vee \bar{a}c\bar{d} \vee ab\bar{c}\bar{d} \vee acd$
- 9) $\bar{a}\bar{c}\bar{d} \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}cd \vee \bar{a}bd \vee \bar{a}\bar{b}cd \vee abc\bar{d} \vee a\bar{b}\bar{d}$
- 10) $abcd \vee \bar{a}\bar{b}\bar{c} \vee b\bar{c}\bar{d} \vee bcd \vee abd \vee \bar{a}bc\bar{d} \vee a\bar{b}c\bar{d}$
- 11) $\bar{a}\bar{c}\bar{d} \vee abcd \vee \bar{a}b\bar{c} \vee bcd \vee a\bar{b}\bar{d} \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d}$
- 12) $\bar{a}\bar{b}c \vee a\bar{c}\bar{d} \vee \bar{a}bcd \vee abcd \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee abc \vee acd$
- 13) $ab\bar{c} \vee acd \vee b\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c} \vee \bar{a}\bar{b}cd \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee abc\bar{d}$
- 14) $\bar{a}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c} \vee \bar{a}bc \vee \bar{a}b\bar{c}\bar{d} \vee ac\bar{d} \vee a\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}c\bar{d}$
- 15) $\bar{a}c\bar{d} \vee \bar{a}\bar{c}\bar{d} \vee \bar{a}bc\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}bc \vee ab\bar{c} \vee a\bar{b}\bar{d}$
- 16) $b\bar{c}\bar{d} \vee \bar{a}bc\bar{d} \vee \bar{a}\bar{b}\bar{d} \vee acd \vee a\bar{b}c \vee abcd \vee \bar{b}\bar{c}\bar{d}$
- 17) $a\bar{c}\bar{d} \vee b\bar{c}\bar{d} \vee abc \vee a\bar{b}\bar{d} \vee a\bar{b}\bar{c}\bar{d} \vee \bar{a}bc\bar{d} \vee \bar{a}\bar{b}c\bar{d}$
- 18) $a\bar{c}\bar{d} \vee ab\bar{c} \vee acd \vee a\bar{b}cd \vee \bar{a}bc \vee a\bar{b}c\bar{d} \vee \bar{a}\bar{b}\bar{c}\bar{d}$
- 19) $\bar{a}\bar{b}\bar{c} \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}c\bar{d} \vee abc \vee a\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{d} \vee ac\bar{d}$
- 20) $acd \vee abd \vee b\bar{c}\bar{d} \vee \bar{a}b\bar{c} \vee abcd \vee a\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}cd$
- 21) $a\bar{b}c\bar{d} \vee abcd \vee \bar{a}b\bar{c} \vee cd \vee a\bar{b}\bar{c} \vee \bar{a}\bar{b}cd \vee \bar{a}\bar{c}\bar{d}$
- 22) $a\bar{b}\bar{d} \vee \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee b\bar{c}\bar{d} \vee a\bar{b}c\bar{d} \vee ab\bar{c}\bar{d} \vee \bar{a}\bar{b}c\bar{d}$
- 23) $\bar{a}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{d} \vee a\bar{b}\bar{c}\bar{d} \vee ab\bar{c} \vee \bar{b}\bar{c}\bar{d} \vee \bar{a}bc \vee a\bar{b}c\bar{d}$
- 24) $bcd \vee \bar{a}bd \vee a\bar{b}\bar{d} \vee \bar{a}\bar{b}c \vee a\bar{b}c\bar{d} \vee abcd \vee ab\bar{c}\bar{d}$
- 25) $b\bar{c}\bar{d} \vee a\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{d} \vee abc\bar{d} \vee \bar{a}bc \vee \bar{a}cd \vee a\bar{b}cd$
- 26) $abd \vee \bar{a}bc\bar{d} \vee \bar{a}cd \vee abcd \vee \bar{a}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{d} \vee a\bar{b}c\bar{d}$
- 27) $\bar{a}\bar{c}\bar{d} \vee \bar{a}bc \vee \bar{a}\bar{b}\bar{c} \vee a\bar{b}cd \vee a\bar{b}\bar{c}\bar{d} \vee acd \vee \bar{a}\bar{b}c\bar{d}$
- 28) $ac\bar{d} \vee a\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}cd \vee \bar{a}cd \vee \bar{a}\bar{b}c \vee abcd \vee b\bar{c}\bar{d}$
- 29) $a\bar{b}\bar{c}\bar{d} \vee a\bar{c}\bar{d} \vee abd \vee a\bar{b}c\bar{d} \vee ac\bar{d} \vee \bar{a}\bar{b}c \vee \bar{a}\bar{b}\bar{c}\bar{d}$
- 30) $a\bar{b}\bar{d} \vee \bar{a}b\bar{c} \vee ac\bar{d} \vee ab\bar{c}\bar{d} \vee abd \vee \bar{a}\bar{b} \vee \bar{a}bc\bar{d}$

Пример. ДНФ = $a\bar{c}\bar{d} \vee a\bar{b}\bar{c}d \vee abcd \vee \bar{a}\bar{b}c \vee a\bar{b}c\bar{d} \vee bd$.
 $K_1 \quad K_2 \quad K_3 \quad K_4 \quad K_5 \quad K_6$

1) Построим матрицу Грея функции по заданной ДНФ.



2, 3) Найдем сокращенную и кратчайшую ДНФ по матрице Грея. Выделяем все максимальные интервалы $A-E$ (левая и средняя матрицы), они задают сокращенную ДНФ. На правой матрице выделяем минимальное число максимальных интервалов, образующих достаточное множество, они задают кратчайшую ДНФ.



$$\text{СокрДНФ} = a\bar{c} \vee bd \vee \bar{a}\bar{b}c \vee a\bar{b}\bar{d} \vee \bar{b}c\bar{d} \vee \bar{a}cd.$$

$A \quad B \quad C \quad D \quad E \quad F$

$$\text{КратДНФ} = a\bar{c} \vee bd \vee \bar{b}c\bar{d} \vee \bar{a}cd.$$

$A \quad B \quad E \quad F$

4) Построим совершенную ДНФ функции по матрице Грея. Каждая точка задает конъюнкцию совершенной ДНФ.

$$\text{СовДНФ} = \bar{a}\bar{b}cd \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee ab\bar{c}\bar{d} \vee ab\bar{c}d \vee abcd \vee a\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{c}d \vee a\bar{b}c\bar{d}.$$

5) Построим сокращенную ДНФ функции из совершенной ДНФ алгоритмом Квайна-МакКласки. Представляем совершенную ДНФ списком точек, упорядочиваем их по весу и разбиваем на классы.

Список 0	Список 0 (упорядоченный)
0 0 1 1	C_1 1) 0 0 1 0 *
0 0 1 0	2) 1 0 0 0 *
0 1 0 1	C_2 3) 0 0 1 1 *
0 1 1 1	4) 0 1 0 1 *
1 1 0 0	5) 1 1 0 0 *
1 1 0 1	6) 1 0 0 1 *
1 1 1 1	7) 1 0 1 0 *
1 0 0 0	C_3 8) 0 1 1 1 *
1 0 0 1	9) 1 1 0 1 *
1 0 1 0	C_4 10) 1 1 1 1 *

Выполняем все неполные склеивания векторов классов C_i и C_{i+1} , $i = 1, 2, 3$, отмечая участников склеивания. Результаты заносим в список 1 (с номерами склеиваемых векторов). Повторяя аналогичные действия для списка 1, получаем список 2 и приводим в нем подобные. Склеивание векторов из списка 2 невозможно.

Список 1	Список 2
C_1 1) 0 0 1 – (1,3)	1) 1 – 0 – (3,10)
2) – 0 1 0 (1,7)	2) 1 – 0 – (4,9)
3) 1 – 0 0 (2,5) *	3) – 1 – 1 (7,12)
4) 1 0 0 – (2,6) *	4) – 1 – 1 (8,11)
5) 1 0 – 0 (2,7)	
C_2 6) 0 – 1 1 (3,8)	Список 3
7) 0 1 – 1 (4,8) *	пуст
8) – 1 0 1 (4,9) *	
9) 1 1 0 – (5,9) *	
10) 1 – 0 1 (6,9) *	
C_3 11) – 1 1 1 (8,10) *	
12) 1 1 – 1 (9,10) *	

Неотмеченные векторы всех списков задают сокращенную ДНФ (обозначения конъюнкций взяты из задачи 2).

$$\text{СокрДНФ} = \overline{a}\overline{b}c \vee \overline{b}c\overline{d} \vee a\overline{b}\overline{d} \vee \overline{a}cd \vee a\overline{c} \vee bd.$$

C
 E
 D
 F
 A
 B

6) Построим сокращенную ДНФ функции из исходной алгоритмом Блейка-Порецкого. Представляем ДНФ списком троичных векторов, удаляем третий вектор, как поглощаемый шестым. Затем, обобщенно склеивая каждый очередной вектор со всеми предыдущими, получаем, как видно из списка, векторы с седьмого по пятнадцатый. Они либо поглощаются предыдущими, либо остаются в списке, поглощая некоторые из предыдущих векторов.

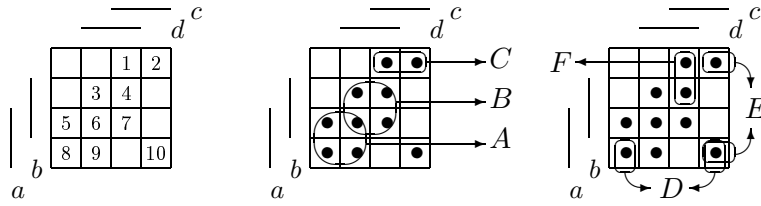
1) 1 - 0 0	12
2) 1 0 0 1	7
3) 1 1 1 1	6
4) 0 0 1 -	
5) 1 0 1 0	8
6) - 1 - 1	
7) 1 0 0 - (2,1)	12
8) 1 0 - 0 (5,1)	12
9) 1 1 0 - (6,1)	12
10) 0 - 1 1 (6,4)	
11) - 0 1 0 (8,4)	
12) 1 - 0 - (9,7)	
13) 0 0 1 - (11,10)	4
14) 1 0 - 0 (12,11)	
15) - 0 1 0 (14,4)	11

Решение образуют шесть непоглощенных векторов, они задают сокращенную ДНФ (обозначения конъюнкций взяты из задачи 2).

$$\text{СокрДНФ} = \underbrace{\bar{a}\bar{b}c}_{C} \vee \underbrace{bd}_{B} \vee \underbrace{\bar{a}cd}_{F} \vee \underbrace{\bar{b}c\bar{d}}_{E} \vee \underbrace{a\bar{c}}_{A} \vee \underbrace{a\bar{b}\bar{d}}_{D}.$$

Вывод. Сокращенные ДНФ, полученные в задачах 2), 5), 6), совпадают, следовательно, они найдены верно.

7) Построим таблицу Квайна функции по ее совершенной и сокращенной ДНФ. На левой матрице нумеруем точки, на остальных повторяем все максимальные интервалы из задачи 2).



Проверяя принадлежность точек максимальным интервалам, строим таблицу Квайна:

	0011 0010 0101 0111 1100 1101 1111 1000 1001 1010										
1-0-					1	1		1	1		A
-1-1			1	1		1	1				B
001-	1	1									C
10-0								1		1	D
-010		1								1	E
0-11	1			1							F
	1	2	3	4	5	6	7	8	9	10	

8) Найдём кратчайшее покрытие таблицы Квайна. В ней нет однострочного покрытия, но строки A и B – ядерные, включаем их в покрытие и вычеркиваем из матрицы вместе с покрытыми столбцами (с третьего по девятый), получаем матрицу Q_1 . В ней нет однострочных покрытий и ядерных столбцов, но есть строка-предшественница: $D \preceq E$, поэтому удаляем строку D . В новой матрице Q_2 появилась ядерная строка E , включаем ее в покрытие и удаляем из матрицы вместе со вторым и десятым столбцами, получаем матрицу Q_3 .

$Q_1 =$	1	1		C
			1	D
		1	1	E
	1			F
	1	2	10	

$Q_2 =$	1	1		C
		1	1	E
	1			F
	1	2	10	

$Q_3 =$	1	C
	1	F
	1	

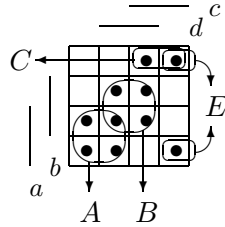
Строка C образует однострочное покрытие матрицы Q_3 . Включив ее в решение, получаем кратчайшее покрытие $\{A, B, C, E\}$.

9) Запишем кратчайшую ДНФ по кратчайшему покрытию таблицы Квайна.

$$\text{КратДНФ}' = a\bar{c} \vee bd \vee \bar{a}\bar{b}c \vee \bar{b}c\bar{d}.$$

$\begin{matrix} & A & B & C & E \end{matrix}$

1') Построим матрицу Грея по полученной кратчайшей ДНФ.



Вывод. Матрицы Грея, построенные в задачах 1) и 1'), совпали, значит, кратчайшая ДНФ из задачи 9) задает исходную функцию. Кроме того, длины кратчайших ДНФ из задач 3) и 9) совпали, значит, кратчайшие ДНФ найдены верно.

10) Построим приближенную кратчайшую ДНФ по матрице Грея алгоритмом Закревского. На левой матрице показываем цены точек, на правой обозначаем точки.

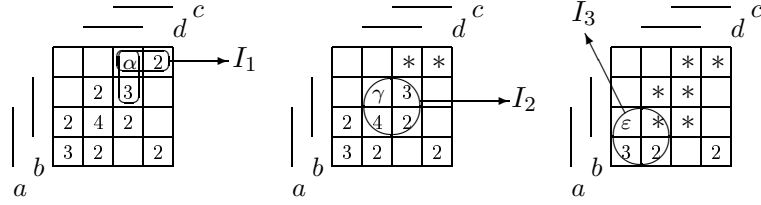
				$\overline{d^c}$
		2	2	
	2	3		
2	4	2		
3	2		2	
a	b			

				$\overline{d^c}$
		α	β	
	γ	δ		
ε	ζ	η		
θ	κ		λ	
a	b			

Выбираем точку α минимальной цены 2. Она входит в два максимальных интервала (левая матрица на следующей странице). Оба интервала содержат по две неотмеченных точки, и оба одинаковой мощности, поэтому в решение включаем любой из них. Отмечаем точки (на средней матрице).

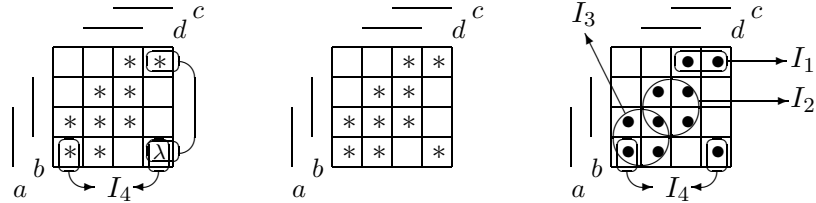
Выбираем неотмеченную точку γ минимальной цены 2. Она входит в один максимальный интервал (средняя матрица), включаем его в решение и отмечаем точки (на правой матрице).

Выбираем неотмеченную точку ε минимальной цены 2. Она входит в один максимальный интервал (правая матрица), включаем его в решение и отмечаем точки (на следующей матрице).



Выбираем неотмеченную точку λ минимальной цены 2. Она входит в два максимальных интервала. Оба содержат по одной неотмеченной точке, и оба одинаковой мощности, поэтому в решение включаем любой из них и отмечаем его точки.

Поскольку все точки отмечены, идем на конец.



$$\text{ПриКратДНФ} = \underbrace{\bar{a}\bar{b}c}_{K_1} \vee \underbrace{bd}_{K_2} \vee \underbrace{a\bar{c}}_{K_3} \vee \underbrace{a\bar{b}\bar{d}}_{K_4}.$$

Вывод. Так как приближенная кратчайшая ДНФ задает исходную функцию, и ее длина не меньше длин кратчайших ДНФ из задач 3) и 9), то приближенная кратчайшая ДНФ найдена верно. •

Литература

1. Быкова С.В., Буркатовская Ю.Б. Булевы функции. Учебно-методическое пособие. Томск: ТГУ, 2002. Ч.1. 37 с.
2. Быкова С.В., Буркатовская Ю.Б. Булевы функции. Учебно-методическое пособие. Томск: ТГУ, 2003. Ч.2. 33 с.

Содержание

11. Минимизация булевых функций	3
11.1. Получение сокращенной ДНФ – первый этап минимизации	4
11.1.1. Теорема Квайна и алгоритм Квайна-МакКласки	7
11.1.2. Теорема Блейка и алгоритм Блейка-Порецкого	10
11.1.3. Упражнения	16
11.2. Построение таблицы Квайна и поиск ее покрытий – второй этап минимизации	17
11.2.1. Таблица Квайна	17
11.2.2. Покрытия таблицы Квайна и ДНФ	18
11.2.3. Поиск всех безыбыточных покрытий	21
11.2.4. Поиск минимальных и кратчайших покрытий	26
11.2.5. Упражнения	29
11.3. Приближенная кратчайшая ДНФ	30
11.3.1. Алгоритм Закревского	31
11.3.2. Упражнения	33
11.4. Контрольная работа	34
Литература	41