Тема. Минимизация булевых функций (БФ) используя Алгоритм Рота.

Теоретические основы метода минимизации булевых функций с использованием алгоритма Рота рассмотрены в учебном пособии (стр. 107 - 116).

Прежде чем выполнять практически задания необходимо познакомиться с этим материалом.

================== Пример 1 ==================

Рассмотрим пример минимизации булевой(логической) функции (БФ) представленной множеством 0-кубов (кубов нулевой размерности) методом Рота. Данная функция может быть получена, например, из таблицы истинности. Исходное покрытие функции задано множествами кубов на которых функция принимает единичное значение − L (единичных) и множества кубов на которых функция не определена (не существует) – N (безразличных). Необходимо понимать, что БФ может быть задана кубами любой, и более высокой, размерности (как например в пособии).

$$L = \begin{pmatrix} 00000 \\ 11001 \\ 00010 \\ 10010 \\ 10110 \\ 01101 \\ 00100 \end{pmatrix} \qquad N = \begin{pmatrix} 00011 \\ 00110 \\ 00111 \\ 01111 \\ 11101 \end{pmatrix}$$

Как отмечалось в пособии, алгоритм Рота (извлечения) выполняется в несколько этапов. Они подробно рассмотрены в пособии. Для понимания выполняемых действий и получаемых при этом результатов введем некоторые понятия. Множество Ci – множество кубов, к которым применяется операция * для образования кубов i+1 (и возможно более высокой) размерности. Множество A_i – множество новых кубов (i-ой и более высокой размерности) полученных на шаге $C_{i-1}*C_{i-1}$. Множество B_i – множество, получаемое из множества C_{i-1} удалением из него полученных на этом (i-1) шаге простых импликант Z_{i-1}. Множество Z_{i-1} – множество простых импликант, полученных в результате операции $C_{i-1}*C_{i-1}$. Итоговое множество простых импликант Z получается объединением всех множеств $Z_0 \dots Z_n$, полученных на n шагах операции *.

Первый этап алгоритма Рота – нахождение множества простых импликант. Поиск простых импликант (т.е. таких импликант которые не порождают импликант более высокой размерности) производится с использованием операции умножения кубов (*). Для простоты понимания операцию * в алгоритме можно ассоциировать с операцией «склеивания». Вначале сформируем исходное покрытие C_0 заданное объединением множеств кубов L и N. Для выполнения операции C_i*C_i используем таблицу. Операцию C_i*C_i будем выполнять до тех пор, пока во множестве C_i будет содержаться <u>более</u> одного куба. Два и более кубов множества C_i могут породить новый куб большей размерности.

Выполняется операция (C_0*C_0):

C_0*C_0	00000	11001	00010	10010	10110	01101	00100	00011	00110	00111	01111	11101
00000	-											
11001		-										
00010	000y0		-									
10010			y0010	-								
10110				10y10	-							
01101						-						
00100	00y00						-					
00011			0001y					-				
00110			00y10		y0110		001y0		-			
00111								00y11	0011y	-		
01111						011y1				0y111	-	
11101		11y01				y1101						-
A1	000x0 00x00	11x01	x0010 0001x 00x10	10x10	x0110	011x1 x1101	001x0	00x11	0011x	0x111		

В таблице в пустых ячейках содержатся кубы с двумя и более координатами ***x***, т.е. не являющиеся новыми кубами. Для простоты они не отражаются в таблице. Наряду с пустыми клетками в таблице содержатся ячейки с кубами одной из координат которых является **у**. Это показывает, что по этой координате произошло «склеивание». Понятно, что в этой таблице, и в последующих, новые кубы – это те, в которых есть только одна такая координата. В результирующем кубе (во множестве A_1) координата **у** заменяется на **х**.

$$A_1 = \begin{pmatrix} 000x0 \\ 00x00 \\ 11x01 \\ x0010 \\ 0001x \\ 00x10 \\ 10x10 \\ x0110 \\ 011x1 \\ x1101 \\ 001x0 \\ 00x11 \\ 0011x \\ 0x111 \end{pmatrix} \qquad Z_0 = \varnothing$$

В результате выполнения операции C_0*C_0 формируется два множества: множество новых кубов A_1, образовавшихся в результате операции * (или склеивания) кубов из исходного множества C_0, и множество кубов Z_0, которые не образовали новых кубов на этом шаге операции *.

Как видно из таблицы на первом этапе в образовании новых кубов множества A_1 приняли участие все кубы исходного множества C_0, поэтому нет кубов, которые следовало бы включить во множество Z_0.

Для выполнения следующего шага (C_1*C_1) формирования множества простых импликант, необходимо сформировать $C_1 = A_1 \cup B_1$. Сформируем множество $B_1 = C_0 - Z_0 = C_0$. Далее в множестве C_0 выполняется операция поглощения кубов меньшей размерности кубами большей размерности. В результате чего все кубы множества B_1 оказываются поглощенные кубами множества A_1, таким образом множество $C_1 = A_1$.

Следующий шаг – выполнение операции C_1*C_1 представлен в таблице:

Из полученных новых кубов образуется множество A_2 , а из кубов, кото-

C_1*C_1	000x0	00x00	11x01	x0010	0001x	00x10	10x10	x0110	011x1	x1101	001x0	00x11	0011x	0x111
000x0	-													
00x00		-												
11x01			-											
x0010				-										
0001x					-									
00x10		00xy0				-								
10x10						y0x10	-							
x0110				x0y10				-						
011x1									-					
x1101										-				
001x0	00yx0										-			
00x11						00x1y						-		
0011x					00y1x								-	
0x111														-
A₂	00xx0	00xx0		x0x10	00x1x	x0x10 00x1x								

рые не участвовали в образовании новых – множество Z_1.

$$A_2 = \begin{Bmatrix} 00xx0 \\ x0x10 \\ 00x1x \end{Bmatrix}, \quad Z_1 = \begin{Bmatrix} 11x01 \\ 011x1 \\ x1101 \\ 0x111 \end{Bmatrix}.$$

Множество C_2, формируется аналогично множеству C_1 как и на предыдущем шаге (из кубов множеств A_2 и B_2). Следующий этап – выполнение операции C_2*C_2 представлен в таблице:

C_2*C_2	00xx0	x0x10	00x1x
00xx0	-		
x0x10		-	
00x1x			-
A_3			

Из таблицы следует, что $A_3 = \varnothing$. Таким образом, новых кубов при выполнении операции C_2*C_2 не было получено.

$$Z_2 = \begin{Bmatrix} 00xx0 \\ x0x10 \\ 00x1x \end{Bmatrix}, \qquad C_3 = A_3 \bigcup B_3 = \varnothing.$$

На этом процесс выявления простых импликант окончен. Таким образом, сформировано множество простых импликант Z:

$$Z = \bigcup_{i=0}^{2} Z_i = Z_0 \bigcup Z_1 \bigcup Z_2 = \begin{Bmatrix} 11x01 \\ 011x1 \\ x1101 \\ 0x111 \\ 00xx0 \\ x0x10 \\ 00x1x \end{Bmatrix}$$

Далее необходимо выяснить, не содержатся ли в этом множестве «лишние» простые импликанты. «Лишние» простые импликанты – это такие, которые могут быть исключены из результирующей (минимальной) функции без нарушения ее корректности. Иначе говоря, это такие наборы, которые «дублируются» другими наборами минимальной функции. Для этого переходим к следующему этапу алгоритма.

Второй этап алгоритма Рота – определение *L*-экстремалей (обязательных простых импликант). Для определения *L*-экстремалей выполняется операция вычитания (#) кубов, результат представлен в таблице:

z#(Z − z)	11x01	011x1	x1101	0x111	00xx0	x0x10	00x1x
11x01	-	yzz1z 011x1	0zzzz 01101	y0zyz 0x111	yyz1y 00xx0	0yzyy x0x10	yyzy0 00x1x
011x1	yz0zz 11x01	-	zzzzz ø	z0zzz 00111	zy0zy 00xx0	1y0zy x0x10	zy0z0 00x1x
x1101	zz0zz 11001	zzz1z 01111	-	zyzyz 00111	zy01y 00xx0	zy0yy x0x10	zy0y0 00x1x
0x111	yzyyz 11001	zzzzz ø		-	zz00y 00xx0	1z0zy x0x10	zz0z0 0001x 00x10
00xx0	yyzzy 11001			zzzzy 00111	-	1zzzz 10x10	zzzz1 00011 zzzzz ø
x0x10	zyzyy 11001			zzzzy 00111	zzz0z 00x00	-	zzzzy 00011
00x1x	yyzyz 11001			zzzzz ø	zzzyz 00x00	yzzzz 10x10	-
Остаток	11001	Ø	Ø	Ø	00x00	10x10	00011

Если после последовательного вычитания из некоторой простой импликанты всех остальных получаем в качестве остатка не пустой куб, то это означает, что только этот исходный (уменьшаемый) куб покрывает этот остаток (куб). Далее если этот остаток еще и содержит единичный набор из множества L, то данная (исходная) простая импликанта будет обязательной или *L*-экстремалью. Проверим, принадлежат ли остатки множеству L с помощью операции пересечения кубов (∩). Результат представлен в таблице:

$z\#(Z-z)\cap L$	00000	11001	00010	10010	10110	10110	01101	00100
11001	Ø	11001	Ø	Ø	Ø	Ø	Ø	Ø
00x00	00000	Ø	Ø	Ø	Ø	Ø	Ø	00100
10x10	Ø	Ø	Ø	10010	10110	Ø	Ø	Ø
00011	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø

Из таблицы следует, что в остатках 11001, 00x00 и 10x10 содержатся наборы из множества L единичных наборов функции. Это значит, что простые импликанты 11x01, 00xx0 и x0x10 являются *L*-экстремалями, а куб 00011 не пересекается с кубами комплекса *L*, и значит соответствующая ему простая импликанта 00x1x не является *L*-экстремалью. Таким образом, получено множество *L*-экстремалей:

$$E = \begin{Bmatrix} 11x01 \\ 00xx0 \\ x0x10 \end{Bmatrix}.$$

Итак, обязательная часть минимального покрытия исходной функции получена. Далее, необходимо выяснить, какие из вершин исходного комплекса кубов *L* не покрываются кубами из множества E (*L*-экстремалями). Все кубы из исходного комплекса *L* должны быть обязательно покрыты. Для этого из каждого куба комплекса *L* вычтем (#) элементы множества *E*. В результате вычитания получим $L_1 = L\#E$. Результат вычитания представлен в таблице:

$L\#E$	00000	11001	00010	10010	10110	10110	01101	00100
11x01	yyzyy 00000	zzzzz ø	yyzyy 00010	zyzyy 10010	zyzyy 10110	zyzyy 10110	yzzzz 01101	yyzyz 00100
00xx0	zzzzz ø		zzzzz ø	yzzzz 10010	yzzzz 10110	yzzzz 10110	zyzzy 01101	zzzzz ø
x0x10				zzzzz ø	zzzzz ø	zzzzz ø	zyzyy 01101	

Из таблицы видно, что не покрывается L-экстремалями куб L_1 = {01101}. Этот куб необходимо покрыть какими либо простыми импликантами, которые не стали L-экстремалями.

$$\hat{Z} = Z \backslash E = \begin{Bmatrix} 11x01 \\ 011x1 \\ x1101 \\ 0x111 \\ 00xx0 \\ x0x10 \\ 00x1x \end{Bmatrix} \backslash \begin{Bmatrix} 11x01 \\ 00xx0 \\ x0x10 \end{Bmatrix} = \begin{Bmatrix} 011x1 \\ x1101 \\ 0x111 \\ 00x1x \end{Bmatrix}$$

Теперь из полученного множества $\hat{Z}$ надо выбрать куб с минимальной ценой (максимальной размерностью), чтобы покрыть набор L_1 = {01101}. Для этого выполним пересечение набора из множества L_1 с кубами из $\hat{Z}$. Результат пересечения приведен в таблице:

	$\hat{Z} \cap L_1$	01101
a	011x1	01101
b	x1101	01101
c	0x111	ø
d	00x1x	ø

Из таблицы видно, что кубы 0x111 и 00x1x не пересекаются с кубом 01101. Следовательно остаются два куба 011x1 и x1101 которые одинаково пересекаются (покрывают) оставшийся (непокрытый) куб 01101. Цена у обоих кубов одинаковая (это оба куба первой размерности). В противном случае необходимо выбрать куб (кубы) максимальной размерности (с большим числом свободных координат **x**).

Следовательно, могут быть получены две тупиковые формы:

$$f^1_{\text{МДНФ}} = \{\ 11x01,\ 00xx0,\ x0x10,\ 011x1\},$$

$$f^2_{\text{МДНФ}} = \{\ 11x01,\ 00xx0,\ x0x10,\ x1101\}.$$

$$f^1_{\text{МДНФ}} = x_1 x_2 \bar{x}_4 x_5 \vee \bar{x}_1 \bar{x}_2 \bar{x}_5 \vee \bar{x}_2 x_4 \bar{x}_5 \vee \bar{x}_1 x_2 x_3 x_5$$

$$f^1_{\text{МДНФ}} = x_1 x_2 \bar{x}_4 x_5 \vee \bar{x}_1 \bar{x}_2 \bar{x}_5 \vee \bar{x}_2 x_4 \bar{x}_5 \vee x_2 x_3 \bar{x}_4 x_5$$

Функциональные схемы для полученных тупиковых форм предлагается построить самостоятельно. При этом желательно попрактиковаться в построении этих схем не только в традиционном базисе: И, ИЛИ, НЕ, но и в других функционально полных базисах [стр.85-86 учебного пособия]:

1. И-НЕ; элемент И с инверсным выходом.
2. И, НЕ; $\{\wedge, \text{не}\}$.
3. XOR, ИЛИ, 1(const 1); $\{v, \oplus, 1\}$.
4. ИЛИ-НЕ; элемент ИЛИ с инверсным выходом.

 • • •

================== П р и м е р 2 ==================

Рассмотрим еще один пример минимизации БФ представленной множеством 0-кубов методом Рота. Исходное покрытие функции задано множествами единичных кубов – L, множество кубов на которых функция не определена – N пусто.

$$L = \begin{pmatrix} 0000 \\ 0001 \\ 0101 \\ 1100 \\ 1101 \\ 1000 \end{pmatrix} \qquad N = \varnothing$$

Вначале (как и в предыдущем примере) сформируем множество кубов C_0=L U N =L.

Первый этап алгоритма Рота – нахождение множества простых импликант. Выполняется операция $C_0 * C_0$, результаты выполнения которой показаны в таблице:

$C_0 * C_0$	0000	0001	0101	1100	1101	1000
0000	-					
0001	000y	-				
0101	0y0y	0y01	-			
1100	yy00	yy0y	y10y	-		
1101	yy0y	yy01	y101	110y	-	
1000	y000	y00y	yy0y	1y00	1y0y	-
A_1	000x x000	0x01	x101	110x 1x00		

В отличие от предыдущего примера таблица небольшая и заполнена без пустых клеток (для лучшего понимания ее формирования). Выделенные желтым цветом наборы – новые импликанты (кубы первой размерности). Как и ранее в результирующем кубе координата **y** заменяется на **x**.

В результате выполнения операции $C_0 * C_0$ формируется два множества: множество новых кубов A_1, образовавшихся в результате операции *, и множество кубов Z_0, которые не образовали новых кубов на этом шаге операции *.

$$A_1 = \begin{pmatrix} 000x \\ x000 \\ 0x01 \\ x101 \\ 110x \\ 1x00 \end{pmatrix} \qquad Z_0 = \varnothing$$

Множество Z_0 пусто, так как в образовании новых кубов множества A_1 приняли участие все кубы исходного множества C_0.

Далее сформируем множество $C_1 = A_1 \cup B_1$. Как и ранее $B_1 = C_0 - Z_0 = C_0$. После выполнения операции поглощения кубов меньшей размерности кубами большей размерности в множестве C_0 получим, что множество $C_1 = A_1$.

Следующий шаг – выполнение операции $C_1 * C_1$ представлен в таблице:

$C_1 * C_1$	000x	x000	0x01	x101	110x	1x00
000x	-					
x000	0000	-				
0x01	0001	000y	-			
x101	0y01	0y0y		-		
110x	yy0x	1y00	y101		-	
1x00	y000	y000	yx0y	110y	1100	-
A_2						

Из таблицы видно, что при выполнении второго шага ($C_1 * C_1$) новых кубов (второй размерности) не было получено. Следовательно:

$$A_2 = \varnothing \qquad Z_1 = \begin{pmatrix} 000x \\ x000 \\ 0x01 \\ x101 \\ 110x \\ 1x00 \end{pmatrix}$$

На этом процесс выявления простых импликант окончен. Таким образом сформировано множество простых импликант Z:

$$Z=\bigcup_{i=0}^{1} Z_i = Z_0 \bigcup Z_1 = \begin{pmatrix} 000x \\ x000 \\ 0x01 \\ x101 \\ 110x \\ 1x00 \end{pmatrix}$$

Как и в предыдущем примере для формированим минимальной ДНФ БФ проверим, не содержатся ли в множестве Z «лишние» простые импликанты. Переходим к следующему этапу алгоритма.

Второй этап алгоритма Рота – определение L-экстремалей. Выполним операцию вычитания (#) кубов, результат вычитания представлен в таблице:

z#(Z − z)	000x	x000	0x01	x101	110x	1x00
000x	-	1zzz 1000	zzzz ø	1yzz 1101	yyzz 110x	y1zz 1x00
x000	zzz1 0001	-		zyzy 1101	zyz1 110x	z1zz 1100
0x01	zzzz ø	yzzy 1000	-	yzzz 1101	yzz0 110x	yzzy 1100
x101		zyzy 1000		-	zzz0 1100	zzzy 1100
110x		zyzz 1000		zzzz ø	-	zzzz ø
1x00		zzzz ø			zzzz ø	-
Остаток	Ø	Ø	Ø	Ø	Ø	Ø

Как видно из таблицы в результате выполнения операции z#(Z − z) не выявлено ни одного обязательного куба (L-экстремали). В отличие от предыдущего примера операцию $z\#(Z{-}z)\cap L$ выполнять не требуется. Это справедливо во-первых потому, что множество N – пусто, и во-вторых так как нечего пересекать (остатки отсутствуют). В этом случае необходимо применить алгоритм «ветвления». Он состоит в ледующем: если L-экстремали не выявлены , то любой куб (не куб из множества N) принадлежащий множеству Z либо вводится в множество E и далее по алгоритму Рота, либо один из кубов исключается из множества Z и повторяется операция z#(Z − z) (до тех пор пока не будут получены остатки). Выполним первый вариант алгоритма ветвления. Введел в множество E куб, например, 110x.

$$E=(110x).$$

Обязательная часть минимального покрытия исходной БФ сформирована.

Определим, какие из вершин исходного комплекса кубов L не покрываются кубами из множества Е. Как и ранее (в предыдущем примере) из каждого куба комплекса L вычтем (#) элементы множества E. В результате вычитания получим $L_1 = L\#E$. Результат вычитания представлен в таблице:

$L\#E$	0000	0001	0101	1100	1101	1000
110x	yyzz 0000	yyzz 0001	yzzz 0101	zzzz ø	zzzz ø	zyzz 1000

Из таблицы видно, что не покрывается L-экстремалями кубы:

$$L_1 = \begin{pmatrix} 0000 \\ 0001 \\ 0101 \\ 1000 \end{pmatrix}.$$

Эти кубы необходимо покрыть какими либо простыми импликантами, которые не стали L-экстремалями.

$$\hat{Z} = Z \setminus E = \begin{Bmatrix} 000x \\ x000 \\ 0x01 \\ x101 \\ 110x \\ 1x00 \end{Bmatrix} \setminus \{110x\} = \begin{Bmatrix} 000x \\ x000 \\ 0x01 \\ x101 \\ 1x00 \end{Bmatrix}$$

Теперь из полученного множества $\hat{Z}$ надо выбрать куб с минимальной ценой (максимальной размерностью), чтобы покрыть наборы (кубы) из множества L_1. Для этого выполним пересечение набора из множества L_1 с кубами из $\hat{Z}$. Результат пересечения приведен в таблице:

	$\hat{Z} \cap L_1$	0000	0001	0101	1000
a	000x	0000	0001	ø	ø
b	x000	0000	ø	ø	1000
c	0x01	ø	0001	0101	ø
d	x101	ø	ø	0101	ø
e	1x00	ø	ø	ø	1000

Из таблицы видно, что кубы x000 и 0x01 максимально пересекаются с кубами из множества L_1 и следовательно их покрывают (реализуют). Следовательно, эти два куба добавляются в минимальное покрытие..

Таким образом, может быть получена тупиковая (минимальная) форма БФ:

$$f_{МДНФ} = \{\ 110x,\ x000,\ 0x01\},$$

$$f_{МДНФ} = x_1 x_2 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_3 x_4$$

Функциональные схемы для полученных тупиковых предлагается построить, как и в предыдущем примере, самостоятельно.

Практические задания.

Используя алгоритм Рота получить минимальную форму булевой функции, исходное покрытие которой задано множествами кубов L (единичных) и N безразличных):

$$L = \begin{pmatrix} 01100 \\ 01110 \\ 10001 \\ 11000 \\ 11011 \\ 01110 \end{pmatrix} \quad N = \begin{pmatrix} 01000 \\ 10011 \\ 11001 \\ 11010 \\ 10111 \end{pmatrix}$$

а)

$$L = \begin{pmatrix} 0000 \\ 0010 \\ 0110 \\ 1101 \\ 1011 \\ 1001 \end{pmatrix} \quad N = \varnothing$$

б)

$$L = \begin{pmatrix} 110x \\ 0x01 \\ 0010 \\ 1010 \end{pmatrix} \quad N = \begin{pmatrix} 0x11 \\ 1000 \\ x111 \end{pmatrix}$$

в)

$$L = \begin{pmatrix} 0001 \\ 0100 \\ 1100 \\ 1101 \\ 1010 \\ 0010 \end{pmatrix} \quad N = \begin{pmatrix} 0101 \\ 0111 \\ 0110 \end{pmatrix}$$

г)