Глава 15

Реализация булевых формул в базисе И, ИЛИ, НЕ линейными бинарными графами

15.1. Формульный метод

В [1] показано, что произвольная нормальная булева формула (БФ) в базисе $\{\&, \vee, !\}$ из h букв может быть реализована бинарным графом (БГ), содержащим $h + 2$ вершины (h — условных и две — операторных: $f = 0$ и $f = 1$).

Из того факта, что число вершин в бинарном графе определяется числом букв в булевой формуле, следует, что для уменьшения числа вершин в графе необходимо минимизировать число букв в формуле, например с помощью классических методов минимизации. Таким образом, эти методы, предложенные в 50-х годах для минимизации релейно-контактных схем, могут применяться и для упрощения программ, реализующих бинарные графы, названных в [1] бинарными программами (БП).

При использовании подхода, изложенного в [1], реализация повторной в указанном базисе БФ сводится к построению БГ для PN-однотипной с ней бесповторной БФ (ББФ) с последующей заменой обозначений переменных. Поэтому в дальнейшем будем рассматривать построение бинарного графа только для ББФ.

В [2] для ББФ был предложен метод построения плоскостных бинарных графов, для реализации которых бинарными программами с одноадресными условными переходами эти графы приходится линеаризовать, что весьма трудоемко [3].

В [1, 4] предложен метод непосредственного построения линейного бинарного графа (ЛБГ) по ББФ. Этот метод назван в [1] формульным. Однако он не обеспечивал планарности ЛБГ для произвольной ББФ. Изложим метод, который строит планарные ЛБГ [5, 6]. Он состоит из следующих этапов.

1. Заданная ББФ, если она содержит групповые инверсии, преобразуется к нормальному виду.

2. Преобразованная ББФ заменяется *PN*-однотипной с ней положительно монотонной ББФ.

3. *h* условных вершин располагаются горизонтально и соединяются между собой последовательно. К последней из них подсоединяется операторная вершина с пометкой «$f = 1$». При этом образуется остов линейного бинарного графа.

4. Расположим операторную вершину с пометкой «$f = 0$» после операторной вершины с пометкой «$f = 1$».

5. Условные вершины пометим символами букв в соответствии с порядком их следования в ББФ. Это завершает построение остова графа.

6. Осуществим пометку ребер остова нулями и единицами таким образом, чтобы создались условия для прохождения всего остова от его начала до операторной вершины с пометкой «$f = 1$». При этом после выполнения условия, соответствующего пометке некоторого ребра остова, должен выполняться переход к следующей вершине. Тем самым оказываются помеченными половина ребер ЛБГ.

7. Если ребро, исходящее из условной вершины, помечено нулем (единицей), то второе исходящее из нее ребро ориентируется вверх (вниз) и помечается единицей (нулем).

8. Каждое вновь введенное исходящее ребро соединим с соответствующей условной или операторной вершиной в зависимости от того, какие буквы ББФ маскируются значением переменной, указанным на рассматриваемом исходящем ребре, считая, что фрагмент ББФ до этой переменной отсутствует.

9. Если в исходную нормальную форму некоторая буква входит с инверсией, то в случае равной доступности прямых и инверсных переменных либо пометка условной вершины, соответствующей этой букве, заменяется на инверсную, либо пометка условной вершины не изменяется, а инвертируется пометка ребер, исходящих из этой вершины. При доступности только прямых переменных имеется лишь вторая из указанных возможностей.

Пример 15.1. Реализовать бесповторную булеву формулу $f = x_1 x_2 \vee x_3$.

На рис. 15.1 приведен линейный бинарный граф, построенный изложенным методом.

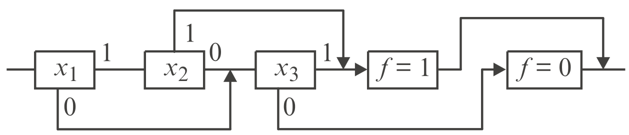

Рис. 15.1

Обратим внимание, что предложенный метод обеспечивает планарность ЛБГ, который может рассматриваться в качестве блока с одним входом и одним выходом.

Изложенный метод может использоваться для реализации не только бесповторных, но и повторных БФ из h букв, зависящих от n переменных ($h > n$). При этом с помощью предлагаемого метода реализуется ББФ из h букв, однотипная с заданной, а затем выполняются переобозначения переменных, помечающих условные вершины.

Этот метод может применяться и для реализации БФ, полученных для не полностью определенных БФУ, заданных на q_0 нулевых и q_1 единичных наборах ($q_0 + q_1 = q$). При этом из [7, 8] следует, что

$$h \leqslant \min\left(2q_1 + q_0 - 3;\ ]3q/2 - 2[\right).$$

Изложенный метод позволяет также строить формализованным образом логические схемы алгоритмов (ЛСА) [4] для БФ в рассматриваемом базисе.

При этом ЛСА строится в три этапа:

— построение линейного бинарного графа;

— преобразование построенного ЛБГ за счет изменения пометок условных вершин таким образом, чтобы все ребра остова были помечены единицами;

— построение ЛСА по преобразованному ЛБГ.

Для булевой формулы, рассмотренной в примере *15.1*, на первом этапе строится ЛБГ (рис. 15.1), который преобразуется в ЛБГ с указанным свойством (рис. 15.2).

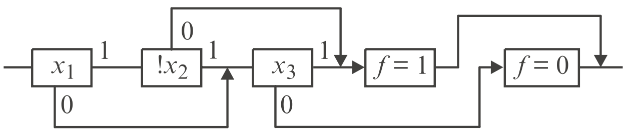

Рис. 15.2

Этому ЛБГ соответствует ЛСА вида:

$$x_1 \overset{1}{\uparrow} !x_2 \overset{2}{\uparrow} \overset{1}{\downarrow} x_3 \overset{3}{\uparrow} \overset{2}{\downarrow} f\ w \overset{4}{\uparrow} \overset{3}{\downarrow} !f \overset{4}{\downarrow},$$

где w — обозначение безусловного перехода.

В заключение раздела отметим, что при замене в ЛБГ каждой условной вершины переключательным контактом получается планарная контактная схема, реализующая одновременно функцию и ее инверсию. Эта схема может быть упрощена, если требуется реализовать только заданную функцию.

15.2. Проверка правильности построения линейного бинарного графа

Правильность построения ЛБГ может быть проверена несколькими методами.

П е р в ы й из них состоит в построении по ББФ таблицы истинности и использовании ее в качестве теста для проверки ЛБГ. Этот метод при больших значениях h весьма трудоемок, так как число тестов при этом равно 2^h.

В т о р о й метод, названный ретрансляцией [9], состоит в свертке ЛБГ с целью получения бесповторной булевой формулы, которую он реализует, и сравнения ее с исходной. Свертка выполняется за h шагов, причем на каждом шаге число условных вершин в ЛБГ уменьшается на одну.

Т р е т и й, наиболее интересный, метод состоит в замене каждой условной вершины в ЛБГ мультиплексором «2 в 1» с целью получения изоморфной схемы, в которой на выходе каждого элемента может быть записана формула. Ошибка в ЛБГ может быть обнаружена, как только фрагмент формулы, получающийся при «чтении» схемы от входов к выходу, будет отличаться от соответствующего фрагмента заданной ББФ.

Пример 15.2. Проверить правильность построения ЛБГ на рис. 15.1 с помощью третьего из указанных методов.

На рис. 15.3 приведена схема из мультиплексоров «2 в 1».

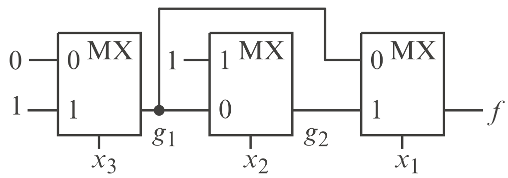

Рис. 15.3

Эта схема описывается следующими соотношениями:

$$g_1 = 0\,\&\,!x_3 \vee 1\,\&\,x_3 = x_3,\; g_2 = g_1!x_2 \vee 1\,\&\,x_2 = x_2 \vee x_3,$$
$$f = g_1!x_1 \vee g_2 x_1 = x_1 x_2 \vee x_3.$$

Таким образом, ЛБГ построен правильно.

При использовании этого метода в о с с т а н о в л е н и е ББФ осуществляется н е п о п о д ф о р м у л а м, а п о ф р а г м е н т а м формулы, несмотря на наличие в ней в общем случае скобок произвольной глубины. При этом отметим, что фрагмент g_2 формулы f не является ее подформулой. Это свойство было впервые установлено в [10].

Ч е т в е р т ы й метод состоит в применении в качестве тестов для проверки программы путей в ее управляющем графе [11]. В этой работе поиск путей выполняется непосредственно по графу (возможно, ошибочному), который собственно и предстоит проверить.

Для устранения этого недостатка предлагается определять пути не по проверяемому графу, а по спецификации, по которой граф строился. Для ЛБГ спецификацией является реализуемая формула. Покажем, что задача перечисления путей в правильном графе может быть решена на основе преобразований заданной ББФ [12].

15.3. Перечисление путей в линейном бинарном графе

Рассмотрим два метода перечисления путей, которые должны быть в правильно построенном ЛБГ [13].

Первый из них состоит в преобразовании бесповторной булевой формы (ее инверсии) в ортогональную дизъюнктивную нормальную форму (ОДНФ) [12]. ОДНФ [14, 15] формулы перечисляет единичные пути, а ОДНФ инверсии — нулевые.

Конъюнкция не требует ортогонализации, а ортогонализация дизъюнкции выполняется по правилу:

$$g_1 \vee g_2 \vee \ldots \vee g_n = g_1 \vee !g_1 g_2 \vee \ldots \vee !g_1 !g_2 \ldots g_n.$$

Пример 15.3. Перечислить пути в линейном бинарном графе по формуле из примера *15.1.*

Построим ОДНФ этой формулы и ее инверсии:

$$\begin{gathered} f = x_1 x_2 \vee x_3 = x_1 x_2 \vee !(x_1 x_2) x_3 = x_1 x_2 \vee (!x_1 \vee !x_2) x_3 = \\ = x_1 x_2 \vee (!x_1 \vee x_1 !x_2) x_3 = x_1 x_2 \vee !x_1 x_3 \vee x_1 !x_2 x_3; \\ !f = (!x_1 \vee !x_2) !x_3 = (!x_1 \vee x_1 !x_2) !x_3 = !x_1 !x_3 \vee x_1 !x_2 !x_3. \end{gathered}$$

Каждая конъюнкция в этих ОДНФ соответствует одному пути в ЛБГ, а ранг рассматриваемой конъюнкции (число букв в ней) — длине этого пути. Таким образом, в данном случае

$$S = S_1 + S_0 = 3 + 2 = 5; \quad L = L_1 + L_0 = 7 + 5 = 12,$$

где $S(L)$, $S_1(L_1)$, $S_0(L_0)$ — число (длина) всех, единичных и нулевых путей.

Изложенный метод весьма трудоемок.

Рассмотрим второй метод, который позволяет одновременно ортогонализовать формулу и ее инверсию, что эквивалентно одновременному перечислению всех путей в правильно построенном ЛБГ.

Разложим по Шеннону формулу и все ее остаточные слева направо, последовательно по всем переменным.

Основная особенность этого метода состоит в том, что подформулы, получаемые в результате разложения, которые умножаются на ноль, не уничтожаются. В получаемой в результате полного разложения формуле раскрываются скобки. Пометка конъюнкции нулем (единицей) свидетельствует о принадлежности ее ОДНФ инверсной (прямой) формулы или о том, что этой конъюнкции соответствует нулевой (единичный) путь в ЛБГ.

Пример 15.4. Перечислить пути в линейном бинарном графе по формуле из примера *15.1.*

Построим одновременно ОДНФ этой формулы и ее инверсии:

$$
\begin{gathered}
f = x_1 x_2 \vee x_3 = !x_1 (x_3) \vee x_1 (x_2 \vee x_3) = \\
= !x_1 (!x_3 \& 0 \vee x_3 \& 1) \vee x_1 (!x_2 x_3 \vee x_2 \& 1) = \\
= !x_1 (!x_3 \& 0 \vee x_3 \& 1) \vee x_1 \big(!x_2 (!x_3 \& 0 \vee x_3 \& 1) \vee x_2 \& 1\big) = \\
= !x_1 !x_3 \& 0 \vee !x_1 x_3 \& 1 \vee x_1 !x_2 !x_3 \& 0 \vee x_1 !x_2 x_3 \& 1 \vee x_1 x_2 \& 1.
\end{gathered}
$$

Изложенный метод может быть применен также и для перечисления путей в ЛБГ по повторной формуле.

15.4. Использование всех путей в качестве полного теста

Покажем, что только перечисление в с е х путей в ЛБГ позволяет восстановить всю таблицу истинности (ТИ), а меньшее их количество эту проблему не решает [6].

Этот факт иллюстрируется табл. 15.1, в которой показано, какие минтермы восстанавливает каждый путь. Так как пути ортогональны, то они покрывают минтермы без пересечения. Из этой таблицы следует, что исключение хотя бы одного пути из теста не позволяет восстановить всю таблицу истинности.

Т а б л и ц а 15.1

Путь	Выход ЛБГ	Минтермы
1. x_1x_2	1	1. $x_1x_2!x_3$ 2. $x_1x_2x_3$
2. $!x_1x_3$	1	3. $!x_1!x_2x_3$ 4. $!x_1x_2x_3$
3. $x_1!x_2x_3$	1	5. $x_1!x_2x_3$
4. $!x_1!x_3$	0	6. $!x_1!x_2!x_3$ 7. $!x_1x_2!x_3$
5. $x_1!x_2!x_3$	0	8. $x_1!x_2!x_3$

Из изложенного следует, что использование в качестве тестов путей, покрывающих все вершины или все дуги управляющего графа [16], в качестве полного теста недостаточно. Только в с е пути позволяют восстановить всю таблицу истинности.

15.5. Подсчет числа путей в бинарном графе

Подсчет числа путей в БГ можно осуществить с помощью метода Флойда [17], который весьма трудоемок, так как основан на возведении матрицы смежности в соответствующую степень.

Так как в данном случае интерес представляют только пути от крайней левой условной вершины до операторных вершин, то можно применить более эффективный метод, предложенный Дейкстрой [17]. Однако и этот метод весьма трудоемок, так как не учитывает специфику БГ.

Предложим два существенно менее трудоемких метода, первый из которых осуществляет подсчет числа путей от выходов к входу, а второй — в обратном порядке. Первый метод позволяет определять величину S, а второй — величины S_1 и S_0 [18].

П е р в ы й м е т о д, при котором БГ рассматривается от выходов к входу, состоит из следующих этапов.

1. Выходы операторных вершин помечаются символом 1.

2. Вход условной вершины помечается числом, которым отмечен выход вершины, связанной с этим входом, а выход условной вершины помечается числом, равным сумме пометок ее входов.

3. Пометка выхода первой условной вершины равна числу путей в БГ.

Пример 15.5. Подсчитать число путей в ЛБГ (рис. 15.1).

Линейный бинарный граф на рис. 15.4 иллюстрирует изложенный метод. Из этого рисунка следует, что $S = 5$.

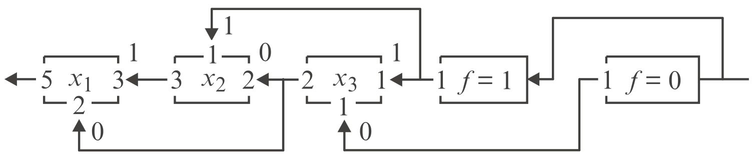

Рис. 15.4

В т о р о й м е т о д, идея которого предложена Акерсом [19], основан на рассмотрении БГ от входа к выходам и состоит в следующем.

1. Вход первой условной вершины помечается символом 1.

2. Пометка входа каждой из остальных вершин графа совпадает с пометкой связанной с ним компоненты графа — выхода условной вершины или точки объединения дуг.

3. Пометка каждого выхода условной вершины совпадает с пометкой ее входа.

4. В каждой точке объединения дуг используется аналог закона Кирхгофа: пометка точки равна сумме пометок выходов условных вершин, связанных с этой точкой.

5. Пометка на входе операторной вершины $f = 1$ ($f = 0$) равна числу единичных (нулевых) путей в БГ.

Пример 15.6. Подсчитать число путей в ЛБГ на рис.15.1.

Линейный бинарный граф на рис. 15.5 иллюстрирует изложенный метод. Из этого рисунка следует, что $S_1 = 3$, $S_0 = 2$.

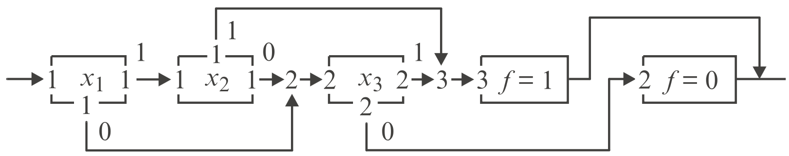

Рис. 15.5

Отметим интересный факт, состоящий в том, что если для положительно монотонной бесповторной пороговой формулы, записанной в порядке неубывания весов, построить ЛБГ и провести разметку с помощью второго метода, то числа, помечающие входы условных вершин, определяют веса соответствующих переменных, а пометка входа операторной вершины $f = 0$ определяет порог функции. При этом отметим, что если для БФУ $f(x_1, \ldots, x_i, \ldots, x_n)$ найдена пороговая реализация $(p_1, \ldots, p_i, \ldots, p_n; T)$, то для БФУ $f(x_1, \ldots, !x_i, \ldots, x_n)$ пороговая реализация имеет вид: $(p_1, \ldots, -p_i, \ldots, p_n; T - p_i)$, а для функции, двойственной f, эта реализация записывается в виде: $\left(p_1, \ldots, p_n; \sum_{i=1}^{n} p_i - T + 1\right)$ [20].

Пример 15.7. Определить пороговую реализацию для пороговой функции $f = x_3 \vee x_2 !x_1$.

Рассмотрим *PN*-однотипную БПФ $f = x_3 \vee x_2 x_1$ и запишем ее в порядке неубывания весов: $f = x_1 x_2 \vee x_3$. Из ЛБГ на рис. 15.5 следует, что $p_1 = 1$, $p_2 = 1$, $p_3 = 2$; $T = 2$. Таким образом, для заданной БПФ пороговая реализация имеет вид: (– 1, 1, 2; 1).

15.6. Подсчет числа путей в блочном бинарном графе

Первый метод осуществляет подсчет путей по блочному ЛБГ, в котором известно число единичных и нулевых путей в каждом блоке, а второй и третий — по реализуемой формуле. Рассмотрим эти методы на примерах.

Пример 15.8. Определить число путей в блочном ЛБГ, реализующем формулу $f = z_1 z_2 \vee z_3$, где $z_1 = x_1 x_2$ ($S_{11} = 3$, $S_{01} = 2$), $z_2 = x_3 x_4 \vee x_5 x_6$ ($S_{12} = 3$, $S_{02} = 4$), $z_1 = x_1 x_2 x_3$ ($S_{13} = 1$, $S_{03} = 3$).

Построим блочный ЛБГ (рис. 15.6) и подсчитаем от входа к выходу число путей в нем. При этом $S_1 = 9$, $S_0 = 18$.

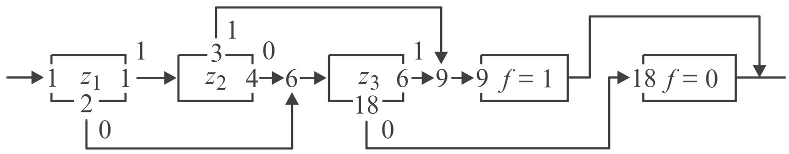

Рис. 15.6

Пример 15.9. Определить число путей в блочном ЛБГ по формуле, приведенной в предыдущем примере.

Построим ортогональные скобочные формы для заданной формулы и ее инверсии: $f = z_1 z_2 \vee (!z_1 \vee z_1 !z_2) z_3$, $!f = (!z_1 \vee z_1 !z_2) !z_3$.

Запишем по этим формам соотношения для подсчета числа путей:

$$S_1 = S_{11} S_{12} + (S_{01} + S_{11} S_{02}) S_{13} = 1 \cdot 3 + (2 + 1 \cdot 4) \cdot 1 = 9;$$

$$S_0 = (S_{01} + S_{11} S_{02}) S_{03} = (2 + 1 \cdot 4) \cdot 3 = 18.$$

Пример 15.10. Определить число путей в блочном ЛБГ по формуле, приведенной в примере *15.8*.

Выполним разложение Шеннона ББФ и всех ее остаточных. Оно получается при $x_i = z_i$ из соотношения, приведенного в примере *15.4*. Исключая в полученном соотношении все нули и единицы, запишем выражение для подсчета числа путей:

$$S = S_{01}(S_{03} + S_{13}) + S_{11}\big(S_{02}(S_{03} + S_{13}) + S_{12} \big) =$$
$$= 2(1 + 3) + 1\big(4(1 + 3) + 3\big) = 27.$$

Изложенные выше подходы рассмотрены в [21].

15.7. Оценки числа и суммарной длины путей в линейных бинарных графах

В [22, 23] доказано, что

$$h + 1 \leqslant S(h) \leqslant F_{h+2};$$
$$(1/2) h(h+3) \leqslant L(h) \leqslant (1/5)\big(4hF_{h+1} + 3(h+1)F_h \big),$$

где $S(h)$, $L(h)$ — число и суммарная длина путей в ЛБГ, содержащем h условных вершин; F_i — i-е число ряда Фибоначчи, в котором $F_1 = F_2 = 1$ [24].

В табл. 15.2 приведены значения F_h, $S_н$, $S_в$, $L_н$, $L_в$ при h = = 1, ... , 14.

Т а б л и ц а 15.2

h	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F_h	1	1	2	3	5	8	13	21	34	55	89	144	233	377
$S_н$	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_в$	2	3	5	8	13	21	34	55	89	144	233	377	610	987
$L_н$	2	5	9	14	20	27	35	44	54	65	77	90	104	119
$L_в$	2	5	12	25	50	96	180	331	600	1075	1908	3360	5878	10255

Известно [24], что

$$F_h = \frac{a^h - b^h}{\sqrt{5}};\ a = \frac{1+\sqrt{5}}{2} \approx 1.618;\ b = \frac{1-\sqrt{5}}{2};\ F_h \approx \left]\frac{a^h}{\sqrt{5}}\right[.$$

Таким образом, при $h > 1$ справедливо неравенство: $S_в(h) < 2^h$.

При $h \to \infty$ $F_{h+1} \approx aF_h$, и поэтому $L_в(h) \lesssim 2hF_h$.

При $h > 10$ $L_в < 2^h$.

Оценим среднюю длину путей:

$$\frac{1}{2}\left(h + \frac{2h}{h+1}\right) \leqslant L_{ср}(h) \lesssim \frac{2hF_h}{F_{h+2}} \approx 0.744h.$$

15.8. Анализ оценок числа путей в линейных бинарных графах

Верхние оценки числа и длины путей в ЛБГ получены на основе анализа таких знакопеременных положительно монотонных БПФ (разд. 1.4.3), у которых [25, 26]

$$p_i = F_i,\ \sum_{i=1}^{h} p_i = F_{h+2} - 1.$$

Для этого класса булевых формул, записанных в порядке неубывания весов переменных, справедливо соотношение [27]: $T = S_0$.

Для формул этого класса с последней операцией ИЛИ:

$$S_0 = F_h,\ S_1 = F_{h+1},\ S = F_{h+2},\ T + \sum_{i=1}^{h} p_i = F_{h+2} + F_h - 1.$$

Для формул этого класса с последней операцией И

$$S_0 = F_{h+1},\ S_1 = F_h,\ S = F_{h+2},\ T + \sum_{i=1}^{h} p_i = F_{h+3} - 1.$$

Формулы этого класса обладают удивительным свойством: при одном порядке переменных число путей определяется верхней оценкой, а при противоположном порядке переменных — нижней. Таким образом, каждая формула этого класса одновременно является и самой сложной и самой простой по рассматриваемому критерию среди всех формул в указанном базисе из h букв [21]. Другие объекты в дискретной математике, обладающие таким свойством, автору не известны.

Пример 15.11. Определить число путей в линейных бинарных графах, реализующих булевы формулы

$$f=(x_1x_2\vee x_3)x_4\vee x_5 \quad \text{и} \quad f=x_5\vee x_4(x_3\vee x_2x_1)$$

Так как в этом случае $h = 5$, то $6 \leqslant S(5) \leqslant 13$. На рис. 15.7 приведен ЛБГ, построенный по первой формуле, содержащий 13 путей, а на рис. 15.8 приведен ЛБГ, содержащий шесть путей, который построен по второй формуле .

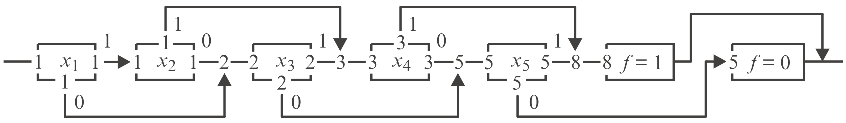

Рис. 15.7

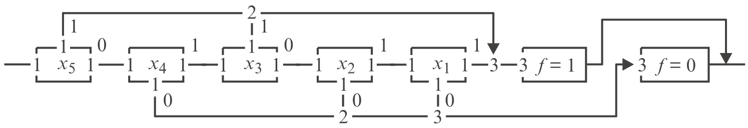

Рис. 15.8

Из рассмотрения ЛБГ (рис. 15.7) можно записать:

$$f=\operatorname{sign}(x_1+x_2+2x_3+3x_4+5x_5-5).$$

Из изложенного следует, что так как верхняя оценка достигается на тех формулах, которые могут быть реализованы также и по нижней оценке, то вопрос о верхней оценке числа путей для «оптимально» записанных формул остается открытым.

15.9. Верхняя оценка числа путей в линейных бинарных графах, реализующих инвариантные формулы

Назовем инвариантными булевы формулы, в которых изменение порядка их записи не изменяет числа путей в ЛБГ.

Автор предполагает [23], что среди инвариантных БФ наиболее сложными по рассматриваемому критерию являются дизъюнктивные (конъюнктивные) нормальные формы. В классе ДНФ наиболее сложными являются формулы, которые образованы конъюнкциями, средняя длина которых равна «e». Эти ДНФ состоят из конъюнкций, содержащих две и (или) три буквы. В [23] доказано, что

$$S_{\text{в}}(h)=\begin{cases}2^{(h/2)+1}-1, & \text{если } h=2,4,6,8,10,12,14,16;\\ (1/2)\,(3^{(h/3)+1}-1), & \text{если } \bmod_3 h=0 \text{ и } h\neq 6,12;\\ 2\cdot 3^{(h-1)/3}+1, & \text{если } \bmod_3 h=1 \text{ и } h\neq 4,10,16;\\ 3^{(h+1)/3}, & \text{если } \bmod_3 h=2 \text{ и } h\neq 2,8,14.\end{cases}$$

Отметим, что из этой оценки следует, что число путей в ЛБГ, реализующих формулы указанного класса, превышает число клик (максимальных подграфов) в графах Муна—Мозера, являющихся наиболее сложными по числу клик (K) графами. Для графов Муна—Мозера справедливо [29] соотношение:

$$\text{K}=\begin{cases}3^{h/3}, & \text{если } \bmod_3 h=0;\\ 4\cdot 3^{(h-4)/3}, & \text{если } \bmod_3 h=1;\\ 2\cdot 3^{(h-2)/3}, & \text{если } \bmod_3 h=2.\end{cases}$$

Отметим, что именно эти оценки получены в разд. 12.2 [30] для определения максимального числа конъюнкций в ДНФ, получающихся в результате раскрытия скобок в бесповторных формулах в базисе $\{\&, \vee, !\}$.

Тот факт, что полученная оценка числа путей превышает оценку числа конъюнкций в указанных ДНФ, объясняется тем, что, как было показано в разд. 15.3, пути соответствуют конъюнкциям в о р т о г о н а л ь н ы х ДНФ формулы и ее инверсии.

15.10. Оценки числа путей в линейных бинарных графах для оптимально записанных формул

В [23] было отмечено, что знакопеременные положительно монотонные бесповторные пороговые формулы, для которых достигается верхняя оценка числа путей в ЛБГ при четных $h < 18$, могут быть получены расстановкой скобок в инвариантных ДНФ, состоящих из конъюнкций длиной два. Это приводит к весьма близкой структуре ЛБГ для формул этих классов.

Поэтому можно предположить, что приведенная выше верхняя оценка числа путей, возможно, является и верхней оценкой числа путей в ЛБГ для оптимально построенных формул в базисе $\{\&, \vee, !\}$ при $h > 3$.

Таким образом, можно предположить, что

$$h+1 \leqslant S(h) \leqslant S_{\text{в}}(h) \leqslant \max\left(2^{(h/2)+1}-1, 3^{(h+1)/3}\right).$$

В заключение раздела отметим, что верхняя оценка числа путей в бинарном графе, реализующем произвольную БФУ h переменных, достигается для булевой функции $f=x_1 \oplus x_2 \oplus \ldots \oplus x_h$ и определяется соотношением

$$S_{\text{в}}=2^h.$$

Таким образом, из оценок, приведенных в настоящей главе, следует, что, например, для произвольной БФ в базисе {&, ∨, !} из 12 букв верхнюю оценку числа путей в ЛБГ, оптимально реализующем эту формулу, можно снизить с $2^{12}=4096$ до $F_{14}=377$, а возможно, и до $2^7-1=127$.

15.11. Минимизация числа и суммарной длины путей в линейных бинарных графах

Пусть задана некоторая бесповторная БФ. Обозначим в ней инвариантные подформулы новыми буквами. В преобразованной ББФ в общем случае любая перестановка может изменить число и суммарную длину путей в соответствующем ЛБГ. Например, ББФ $f=\left(x_1x_2x_3 \vee (x_4 \vee x_5)x_6\right)(x_7x_8 \vee x_9x_{10}) \vee x_{11}x_{12}$ может быть записана в виде: $f=(z_1 \vee z_2z_3)z_4 \vee z_5$.

Для полученной ББФ существует 16 вариантов записи, каждому из которых соответствует свой ЛБГ с числом путей, равным 132, 123, 117, 114, 111, 108, 102, 93, 89, 87, 87, 87, 78, 78, 69, 69 соответственно. При этом 132 пути имеет ЛБГ, построенный непосредственно по исходной формуле. Уже тот факт, что эта величина больше верхней оценки для оптимально записанных формул (при h = 12 она равна 127), свидетельствует о том, что заданная ББФ может быть записана более оптимально.

Одной из формул, для которой ЛБГ имеет минимальное число путей, равное 69, является ББФ вида:

$$f=x_{11}x_{12} \vee \left(x_6(x_4 \vee x_5) \vee x_1x_2x_3\right)(x_7x_8 \vee x_9x_{10}).$$

В [31] предложен метод «взвешивания подформул», обеспечивающий без перебора перестроение заданной ББФ с целью построения ЛБГ с минимальным числом путей. Именно с помощью этого метода и была построена приведенная выше ББФ, реализуемая ЛБГ с минимальным числом путей.

Этот метод минимизирует также и число единичных и нулевых путей в отдельности, а также суммарную длину путей и число точек объединения дуг в ЛБГ. Оптимальность этого метода связана с бесповторностью реализуемых формул.

Метод основан на том, что если выполняется неравенство

$$\frac{S_{11}-1}{S_{01}} \leqslant \frac{S_{12}-1}{S_{02}} \leqslant \ \dots \ \leqslant \frac{S_{1h}-1}{S_{0h}}, \qquad (15.1)$$

где S_{1i} и S_{0i} — число единичных и нулевых путей в ЛБГ подформулы f_i (i = 1, … , h), то минимальное число путей в ЛБГ, реализующем конъюнкцию, достигается при ее записи в виде: $f = f_1 f_2 \dots f_h$, а если выполняется неравенство

$$\frac{S_{01}-1}{S_{11}} \leqslant \frac{S_{02}-1}{S_{12}} \leqslant \ \dots \ \leqslant \frac{S_{0h}-1}{S_{1h}}, \qquad (15.2)$$

то минимальное число путей в ЛБГ, реализующем дизъюнкцию, достигается при ее записи в виде: $f = f_1 \vee f_2 \vee \dots \vee f_h$.

Одно из следствий этого метода состоит в том, что для минимизации числа путей в ЛБГ бесповторная пороговая формула, соответствующая ему, должна быть записана в порядке невозрастания весов переменных.

Другое следствие относится к классу дизъюнктивных нормальных форм, для которых указанная минимизация обеспечивается, если в каждой формуле конъюнкции записаны в порядке неубывания их длин (рангов) [31].

Пример 15.12. Построить формулу, функционально эквивалентную непороговой ББФ $f = (x_1 \vee x_2)x_3 \vee x_4 x_5$, такую чтобы ЛБГ, ей соответствующий, содержал минимальное число путей.

Подформула $f_1 = x_1 \vee x_2$ реализуется ЛБГ с $S_{11} = 2$, $S_{01} = 1$, а подформула $f_2 = x_2$ реализуется ЛБГ с $S_{12} = 1$, $S_{02} = 1$. Так как для подформулы $f_3 = f_1 f_2$ неравенство (15.1) не выполняется, то подформула f_3 должна быть перестроена: $f_3 = f_2 f_1$. Для этой подформулы может быть построен ЛБГ с $S_{13} = 2$, $S_{03} = 2$.

Для подформулы $f_4 = x_4 x_5$ может быть построен ЛБГ с $S_{14} = 1$, $S_{04} = 2$.

Так как для формулы $f = f_3 \vee f_4$ неравенство (15.2) выполняется, то формула не перестраивается.

Таким образом, оптимально записанная ББФ имеет вид:

$$f = x_3 (x_1 \vee x_2) \vee x_4 x_5.$$

Для ЛБГ, построенного по этой формуле, S = 8 ($S_1 = S_0$ = 4), в то время как для исходной формулы S = 11 ($S_1 = 5$, $S_0 = 6$), а для формулы $f = x_4 x_5 \vee x_3 (x_1 \vee x_2)$ эти величины имеют следующие значения: $S = 9$, $S_1 = 5$, $S_0 = 4$.

При оптимизации ББФ по рассматриваемому критерию вместо соотношений 15.1 и 15.2 может использоваться также более простой эвристический критерий для определения целесообразности перестановки фрагментов формулы, предложенный автором и А. Ю. Еремеевым.

Этот подход основан на подсчете вероятностей нулевых (Q) и единичных (P) значений в столбце значений таблицы истинности, соответствующей каждой рассматриваемой подформуле, на сравнении этих показателей для подформул, объединенных конъюнкцией или дизъюнкцией, и на перестановке этих подформул по результатам сравнения.

При этом если $Q_1 > Q_2$, то ЛБГ строится по формуле $f = f_1 f_2$, в противном случае — по формуле $f = f_2 f_1$; если $P_1 > P_2$, то ЛБГ строится по формуле $f = f_1 \vee f_2$, в противном случае — по формуле $f = f_2 \vee f_1$.

Применение этого подхода бывает целесообразным, так как для подформулы f подсчет вероятностей P_i и Q_i с помощью первого метода, изложенного в разд. 1.4.2, менее трудоемок по сравнению с подсчетом числа путей S_{1i} и S_{0i}. Однако этот подход является эвристическим, так как в этом критерии используются характеристики, отличные от оптимизируемой.

Пример 15.13. Построить формулу, функционально эквивалентную непороговой ББФ $f = (x_1 \vee x_2)x_3 \vee x_4 x_5$, такую чтобы ЛБГ, ей соответствующий, содержал минимальное число путей.

В таблице истинности $p_i = q_i = 1/2$. При этом для подформулы $f_1 = x_1 \vee x_2$ справедливы соотношения: $Q_1 = q_1 q_2 = 1/4$, $P_1 = 1 - Q_1 = 3/4$, а для подформулы $f_2 = x_3$ — соотношения $Q_2 = q_3 = 1/2$, $P_2 = p_3 = 1/2$.

Так как $Q_2 > Q_1$, то $f_3 = f_2 f_1$. При этом $P_3 = P_2 P_1 = 3/8$, $Q_3 = 1 - P_3 = 5/8$.

Для подформулы $f_4 = x_4 x_5$ справедливы соотношения: $P_4 = p_4 p_5 = 1/4$, $Q_4 = 1 - P_4 = 3/4$.

Так как $P_3 > P_4$, то $f = f_3 \vee f_4$. Таким образом, искомая формула совпадает с найденной в предыдущем примере с помощью точного, но более трудоемкого метода.

Автором и С. Ф. Королюком был предложен еще более простой эвристический критерий для решения поставленной задачи: в конъюнкции на первое место ставится подформула с меньшим значением вероятности числа единиц, а для дизъюнкции — с большим значением этого показателя.

Подсчет этого показателя целесообразно проводить с помощью второго метода, изложенного в разд. 1.4.2.

Пример 15.14. Построить формулу, функционально эквивалентную непороговой ББФ $f = (x_1 \vee x_2)x_3 \vee x_4 x_5$, такую чтобы ЛБГ, ей соответствующий, содержал минимальное число путей.

В таблице истинности $p_i = q_i = 1/2$. При этом для подформулы $f_1 = x_1 \vee x_2$ справедливы соотношения: $Q_1 = q_1 q_2 = 1/4$, $P_1 = 1 - Q_1 = 3/4$, а для подформулы $f_2 = x_3$ справедливо соотношение $P_2 = p_3 = 1/2$.

Так как подформула f_3 — конъюнкция, а $P_2 < P_1$, то $f_3 = f_2 f_1$. При этом $P_3 = P_2 P_1 = 3/8$.

Для подформулы $f_4 = x_4 x_5$ справедливо соотношение $P_4 = p_4 p_5 = 1/4$.

Так как формула f_3 — дизъюнкция, а $P_3 > P_4$, то $f = f_3 f_4$. Таким образом, искомая формула совпадает с найденной в двух предыдущих примерах с помощью более трудоемких методов.

15.12. Реализация повторных формул

Реализация повторных формул с помощью ЛБГ сопряжена для отдельных типов контроллеров (без запоминания значений входных переменных на программный цикл) с проблемой риска — изменениями значений одной и той же входной переменной за время вычисления ЛБГ [31].

Для устранения риска бинарный граф может строиться по булевой формуле с помощью метода каскадов, основанного на разложении Шеннона формулы и всех ее остаточных на каждом шаге по одной из входных переменных [1, 32]. При этом в общем случае вопрос об оптимизации числа путей в БГ остается открытым.

Для БФ с малой повторностью переменных совместное использование формульного метода и метода каскадов может обеспечить построение БГ с минимальным числом путей и без «риска».

Пример 15.15. Реализовать булеву формулу $f = (x_1 x_2 \vee x_3)!x_4 \vee x_4 (x_5 \vee x_6) x_7$.

Выполним разложение заданной формулы по переменной x_4:

$$f = f_1 !x_4 \vee f_2 x_4.$$

При этом если остаточные записать в виде: $f_1 = x_3 \vee x_1 x_2$, $f_2 = x_7 (x_5 \vee x_6)$, и построить для каждой из них ЛБГ, то БГ в целом будет содержать минимальное число путей, равное восьми, а риск при вычислениях по нему отсуствует.

В заключение главы покажем, что существуют БФ, для которых предлагаемый метод построения ЛБГ существенно более эффективен по сравнению с известным методом построения БГ на основе многократного применения разложения Шеннона по одной переменной. В [33] показано, что булева формула $f_1 = x_1 x_2 \vee \ldots \vee x_{2n-1} x_{2n}$ при порядке переменных $x_1, x_2, \ldots, x_{2n}$ реализуется известным методом БГ, состоящим из $2n + 2$ вершин, в то время как булева формула $f_2 = x_1 x_{n+1} \vee \ldots \vee x_n x_{2n}$ при том же порядке переменных реализуется тем же методом БГ, содержащим 2^{n+1} вершин. При этом отметим, что формульный метод для каждой из этих формул строит планарный ЛБГ из $2n + 2$ вершин. Из изложенного также следует, что для многих булевых функций построение БФ позволяет «увидеть» порядок переменных, при котором строится более простой БГ, что не удается сделать по таблице истинности.

Так как в [34] показано, что число букв в БФ в базисе $\{\&, \vee, !\}$ может зависеть от числа переменных экспоненциально, то в следующей главе рассматриваются методы построения бинарного графа для булевых функций, которые формульным методом реализуются неэффективно. При этом отметим, что в [33] показано, что среди $2n$ функций, описывающих работу параллельного умножителя двух n-разрядных двоичных чисел, существует такая функция, которая вне зависимости от порядка переменных при больших n реализуется БГ, содержащим по крайней мере $2^{n/8}$ вершин.

Выводы

1. Предложен метод реализации произвольной булевой формулы в базисе $\{\&, \vee, !\}$ из h букв планарным линейным бинарным графом (ЛБГ) из $h + 2$ вершин.

2. Предложен метод верификации ЛБГ, состоящий в восстановлении по нему реализуемой булевой формулы. При этом восстановление формулы осуществляется не по подформулам, а по ее фрагментам, которые могут не являться подформулами.

3. Предложены два метода перечисления путей в правильно построенном ЛБГ по заданной булевой формуле.

4. Показано, что только перечисление всех путей в бинарных графах позволяет восстановить всю таблицу истинности булевой функции.

5. Предложены простые методы подсчета числа путей в бинарных графах непосредственно по этим графам.

6. Предложен метод построения пороговой реализации для произвольной положительно монотонной бесповторной пороговой формулы с помощью ЛБГ, содержащего максимальное число путей.

7. Получены оценки числа $S(h)$ и суммарной длины $L(h)$ путей в ЛБГ от числа букв h в реализуемой БФ в базисе $\{\&, \vee, !\}$, в которых используются числа Фибоначчи.

8. Определен класс формул, таких что они при одном порядке записи реализуются ЛБГ с минимальным числом путей, а при противоположном порядке записи — ЛБГ с максимальным числом путей.

9. Определена верхняя оценка числа $S(h)$ путей в линейных бинарных графах, реализующих инвариантные (относительно порядка записи) формулы в базисе $\{\&, \vee, !\}$ из h букв. Из этой оценки следу-

ет, что число путей в ЛБГ, реализующих формулы указанного класса, превышает число клик в графах Муна—Мозера, содержащих максимальное число клик среди графов с h вершинами.

10. Высказано предположение, что верхняя оценка числа путей в ЛБГ, реализующих инвариантные формулы, является также и верхней оценкой числа путей в ЛБГ для оптимально построенных произвольных булевых формул в рассматриваемом базисе.

11. Предложен точный метод, обеспечивающий минимизацию числа путей в ЛБГ, реализующем булевы формулы, бесповторные в базисе $\{\&, \vee, !\}$. Предложены также два эвристических менее трудоемких метода для решения этой задачи.

12. Изложенные методы могут рассматриваться также как новый подход к построению, минимизации, подсчету и перечислению числа конъюнкций в ОДНФ, широко применяемых в логико-вероятностных методах расчета надежности [14]. При этом число единичных путей в ЛБГ для заданной булевой формуле, бесповторной в рассматриваемом базисе, равно числу конъюнкций в однозначно построенной по ней ОДНФ, а для повторной в этом базисе формулы — верхней оценке числа указанных конъюнкций. Число конъюнкций в ОДНФ, построенной по произвольной булевой формуле в указанном базисе из h букв, не превышает числа Фибоначчи с номером $h + 1$.

Литература

1. *Кузнецов О. П.* О программной реализации логических функций и автоматов // Автоматика и телемеханика. 1977. № 7, 9.

2. *Ершов А. П.* Введение в теоретическое программирование. М.: Наука, 1977.

3. *Кириллов А. П., Кузнецов Б. П., Шалыто А. А.* Минимизация числа безусловных переходов в бинарных программах // Судостроит. пром-сть. Сер. Автоматика и телемеханика. 1986. Вып.1.

4. *Камынин С. С., Любимский Э. З., Шура-Бура М. Ю.* Об автоматизации программирования при помощи программирующей программы // Проблемы кибернетики. М.: Физматгиз, 1958. Вып.1.

5. *Артюхов В. Л., Кузнецов Б. П., Шалыто А. А.* Линейные бинарные графы: свойства, характеристики, алгоритмы // Методы и программы решения оптимизационных задач на графах и сетях. Ч.2. Тез. докл. III Всесоюз. совещ., Ташкент, 1984.

6. *Артюхов В. Л., Кузнецов Б. П., Шалыто А. А.* Настраиваемые устройства для судовых управляющих систем. Л.: ИПК СП, 1986.

7. *Кукинов А. М.* Простой метод синтеза скобочных формул для недоопределенных булевых функций // Синтез дискретных автоматов и управляющих устройств. М.: Наука, 1968.

8. *Редькин Н. П.* О сложности реализации недоопределенных булевых функций // Автоматика и телемеханика. 1969. № 9.

9. *Кириллов А. П., Кузнецов Б. П., Шалыто А. А.* Метод ретрансляции булевых формул // Вопросы судостроения. Сер. Судовая автоматика. 1984. Вып.30.

10. *Артюхов В. Л., Шалыто А. А.* Судовые управляющие логические системы. Л.: ИПК СП, 1984.

11. *Липаев В. В.* Качество программного обеспечения. М.: Финансы и статистика, 1983.

12. *Кузнецов Б. П., Шалыто А. А.* Система преобразований некоторых форм представления булевых функций // Автоматика и телемеханика. 1985. № 11.

13. *Артюхов В.Л., Кузнецов Б.П., Шалыто А.А.* Реализация булевых формул бинарными графами // Методы и программы решения оптимизационных задач на графах и сетях. Ч.2. Тез. докл. IV Всесоюз. совещ. Новосибирск, 1989.

14. *Рябинин И. А.* Основы теории и расчета надежности судовых электроэнергетических систем. Л.: Судостроение, 1971.

15. *Смирнов А. С.* Карточный метод ортогонализации функций алгебры логики // Вопросы судостроения. Сер. Судовая автоматика. 1977. Вып.16.

16. *Бичевский Я. Я., Борзов Ю. В.* Тестирование программ ЭВМ. Рига: Изд-во ЛатвГУ, 1985.

17. *Свами М., Тхуласирами К.* Графы, сети и алгоритмы. М.: Мир, 1984.

18. *Кузнецов Б. П., Шалыто А. А.* Тестирование граф-схем бинарных программ // Методы и системы технической диагностики. Математические методы кибернетики и их приложение. Межвуз. науч. сборник. Вып.14, ч. 2. Саратов: СГУ, 1990.

19. *Akers S. B.* Binary decision diagrams // IEEE Trans. on Computers. 1978. N 6.

20. *Боголюбов И. Н., Овсиевич Б. Л., Розенблюм Л. Я.* Синтез схем из пороговых и мажоритарных элементов // Сети передачи информации и их автоматизация. М.: Наука, 1965.

21. *Кузнецов Б. П., Шалыто А. А.* Реализация булевых формул линейными бинарными графами. I. Синтез и анализ // Изв. РАН. Техн. кибернетика. 1994. № 5.

22. *Артюхов В. Л., Кузнецов Б. П., Шалыто А. А.* Оценки сложности реализации булевых формул бинарными графами // Методы и программы решения оптимизационных задач на графах и сетях. Ч.2. Тез. докл. III Всесоюз. совещ. Ташкент, 1984.

23. *Кузнецов Б. П., Шалыто А. А.* Реализация булевых формул линейными бинарными графами. II. Оценки числа и суммарной длины путей // Изв. РАН. Теория и системы управления. 1995. № 3.

24. *Воробьев Н. Н.* Числа Фибоначчи. М.: Наука, 1978.

25. *Артюхов В. Л., Розенблюм Л. Я., Шалыто А. А.* Логические возможности некоторых типов каскадных структур // Сети связи и дискретные устройства управления. М.: Наука, 1976.

26. *Фридман А., Менон П.* Теория переключательных схем. М.: Мир, 1978.

27. *Кондратьев В. Н., Шалыто А. А.* Реализация булевых функций одним линейным арифметическим полиномом с маскированием // Автоматика и телемеханика. 1996. № 1.

28. *Артюхов В. Л., Кузнецов Б. П., Шалыто А. А., Плошко Н. С.* Особенности записи булевых формул при их программной реализации // Вопросы судостроения. Сер. Судовая автоматика. 1984. Вып. 31.

29. *Горбатов В. А.* Основы дискретной математики. М.: Высш. школа, 1986.

30. *Артюхов В. Л., Кузнецова О. С., Шалыто А. А.* Оценка функциональных возможностей программируемых логических матриц // Автоматика и вычисл. техника. 1985. № 2.

31. *Кузнецов Б. П., Шалыто А. А.* Реализация булевых формул линейными бинарными графами. III. Оптимизация числа и суммарной длины путей // Изв. РАН. Теория и системы управления. 1995. № 5.

32. *Поваров Г. Н.* Математическая теория синтеза контактных (1, k)-полюсников // Доклады АН СССР. 1955. № 5.

33. *Brayant R. E.* Graph-based algorithms for boolean function manipulation // IEEE Trans. on Computers. 1986. N 8.

34. *Лупанов О. Б.* Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984.