ТЕХНИЧЕСКАЯ
КИБЕРНЕТИКА
№ 5 • 1994

УДК 681.3.06:62-507

© 1994 г. Б. П. КУЗНЕЦОВ, А. А. ШАЛЫТО

РЕАЛИЗАЦИЯ БУЛЕВЫХ ФОРМУЛ ЛИНЕЙНЫМИ БИНАРНЫМИ ГРАФАМИ. I. СИНТЕЗ И АНАЛИЗ

Рассматривается метод синтеза булевых формул линейными бинарными графами и методы их проверки. Излагаются способы перечисления и подсчета путей в этих графах. Устанавливается связь между характеристиками путей и быстродействием.

Введение. В настоящее время в связи с широким использованием в системах управления микроЭВМ возрастает интерес к программной реализации булевых функций [1—10]. Одним из основных методов, используемых при программной реализации указанного класса функций, является бинарное программирование. Это объясняется тем, что при применении бинарных программ (БП) принципиально отсутствует необходимость в оперативной памяти для запоминания промежуточных результатов, а кроме того, как будет показано ниже, при использовании языка ассемблера некоторых микроЭВМ число команд в БП не зависит от того, каким образом входные переменные размещены в слове или как они расположены в разных словах.

В силу того что ассемблеры большинства микроЭВМ содержат лишь одноадресные условные переходы, после построения граф-схемы (ГС) она линеаризуется. Этого недостатка лишен формульный метод [1, 8], который позволяет непосредственно строить линеаризованные ГС по булевым формулам, являющимся наиболее распространенной формой задания булевых функций в системах управления [11]. При таком задании метод обладает весьма малой трудоемкостью. Этот метод, впервые использованный в [12], был усовершенствован в [1, 8]. Однако практическое применение метода требует его дальнейшего улучшения. Кроме того, в указанных работах не рассматривались вопросы анализа ГС, синтезируемых формульным методом. Настоящая работа посвящена решению указанных вопросов.

1. Метод построения линейного бинарного графа. В [1] было показано, что произвольная формула в базисе И, ИЛИ, НЕ (в дальнейшем будем называть ее формулой) из h букв может быть реализована ГС, содержащей $h+2$ вершины (h условных и две операторные — присвоение выходной переменной констант 0 и 1).

Из того факта, что число вершин в ГС определяется числом букв в реализуемой формуле, следует, что для уменьшения числа вершин (команд в БП) необходимо иметь формулу с наименьшим числом букв, что обеспечивается использованием классических методов минимизации. Таким образом при реализации БП, например в базисе микроЭВМ, методы минимизации, разработанные в 50-х годах для сокращения числа релейно-контактных элементов в схемах, остаются актуальными и современными.

Реализация предложенным методом повторной формулы сводится к построению БП для однотипной с ней бесповторной булевой формулы (ББФ) с последующей заменой обозначения переменных и поэтому в дальнейшем будем рассматривать построение БП только для бесповторных формул [11].

В [8] был предложен метод построения для заданной формулы линейного бинарного графа (ЛБГ), соответствующего БП. Однако этот подход не обеспечивал планарности ЛБГ. Ниже излагается метод, устраняющий это ограничение.

1. Заданную формулу преобразуем к нормальному виду, например, с помощью метода, предложенного в [11].

2. Преобразованную формулу заменим однотипной с ней положительно монотонной формулой, для которой первоначально и строится ЛБГ.

3. Расположим h условных вершин горизонтально и соединим их последовательно, подсоединив к последней из них операторную вершину $y = 1$. При этом образуется остов ЛБГ.

4. Расположим операторную вершину $y = 0$ после вершины $y = 1$.

5. Условные вершины пометим символами букв в соответствии с порядком их следования в формуле.

6. Осуществим пометку ребер остова нулями и единицами [8] таким образом, чтобы создались условия для прохождения всего остова от его начала до единичной операторной вершины. При этом после выполнения условия, соответствующего пометке некоторого ребра остова, должен выполняться переход к следующей вершине. Тем самым оказывается помеченной половина ребер ЛБГ.

7. Если ребро, исходящее из условной вершины, помечено нулем (единицей), то второе исходящее ребро этой вершины 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 формулы маскируются значением переменной, указанным на рассматриваемом исходящем ребре, считая, что фрагмент формулы до этой переменной отсутствует.

9. Если в исходную нормальную формулу некоторая буква входит с инверсией, тогда в случае, если прямые и инверсные выходы источников информации (ИИ) равнодоступны, то либо пометка вершины, соответствующей этой букве, заменяется на инверсную, либо пометка вершины не изменяется, а инвертируется пометка ребер, исходящих из этой вершины. При доступности прямых выходов ИИ имеется лишь вторая из указанных возможностей.

На рис. 1 приведен пример построения ЛБГ для формулы $y = x_1 x_2 \vee x_3$ на основе изложенного подхода. Предложенный метод позволяет весьма просто построить логическую схему алгоритма (ЛСА) по заданной формуле: формула реализуется ЛБГ; в ЛБГ заменяется на инверсную пометка тех ребер и условных вершин, им соответствующих, у которых ребра, входящие в остов, помечены нулем; в строку слева направо выписываются пометки всех вершин модифицированного иа предыдущем этапе ЛБГ; стрелки и их пометки в ЛСА соответствуют ребрам ЛБГ, не входящим в остов.

Изложенный подход позволяет для рассмотренной выше формулы построить ЛСА, которая имеет следующий вид:

$$x_1 \overset{1}{\uparrow} \bar{x}_2 \overset{2}{\uparrow} \overset{1}{\downarrow} x_3 \overset{3}{\uparrow} \overset{2}{\downarrow} y \overset{3}{\downarrow} \bar{y}.$$

Перейдем к вопросу построения БП. Последняя на основе ЛБГ реализуется следующим образом: вводится оператор ω, соответствующий команде безусловного перехода, образуя линеаризованный бинарный граф (БГ) — ЛНБГ; по ЛНБГ строится БП. Покажем, что для БП, построенном предлагаемым методом с помощью языка макроассемблера ЭВМ PDP или VAX, число команд не зависит от того, каким образом входные переменные размещены в слове или как они расположены в разных словах.

П р и м е р 1. Построить БП, реализующую формулу $y = (x_1 x_2 \vee x_3\bar{x}_4) x_5$, при условии, что переменные x_1, x_2, x_3 размещены в 14, 6 и 3 разрядах первого слова, а переменные x_4 и x_5 — в 5 и 2 разрядах второго слова (рис. 2).

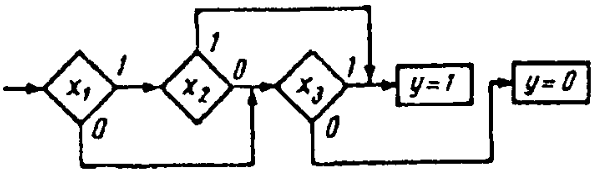

Рис. 1

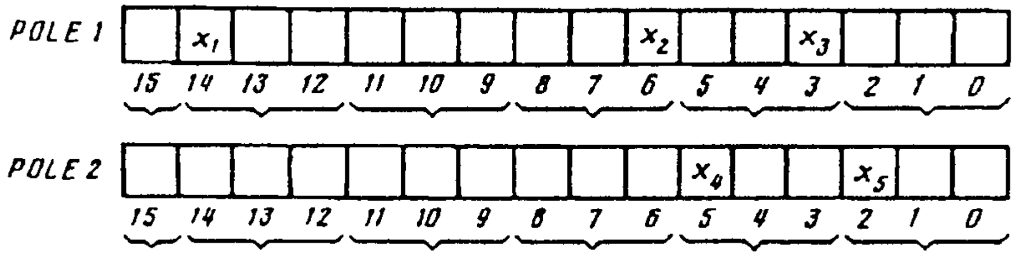

Рис. 2

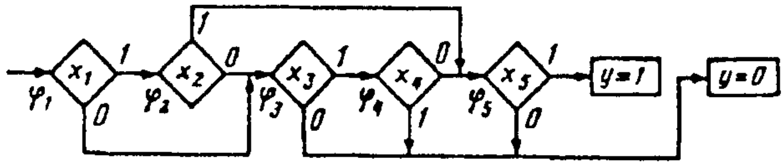

Рис. 3

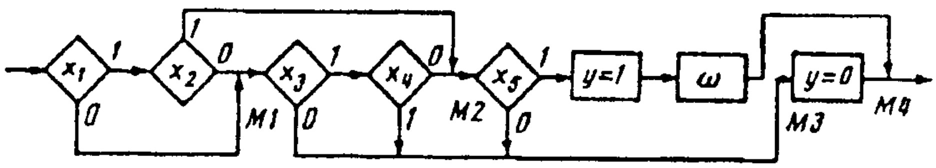

Рис. 4

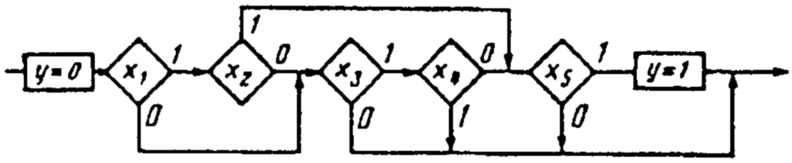

Рис. 5

На рис. 3, 4 приведены ЛБГ и ЛНБГ, построенные предлагаемым методом. Текст фрагмента программы, написанной по ЛНБГ, приведен ниже:

```
BIT # 040000, POLE1
BEQ M1
BIT # 000100, POLE1
BNE M2
M1: BIT # 000010, POLE1
BEQ M3
BIT # 000040, POLE2
BNE M3
M2: BIT # 000004, POLE2
BEQ M3
MOV # 1, y
BR M4
M3: CLR y
M4:
```

Константы 040000, 000100, 000010, 000040, 000004, записанные в восьмиричной

Таблица 1

Реализация бинарной программы

Наименование команды	Количество		Емкость памяти
	слов	команд	
BIT#C, POLEi	3	h	$3h$
BEQ(BNE)	1	h	h
MOV#1, y	3	1	3
BR	1	1	1
CLRy	2	1	2
Итого	—	$2h+3$	$4h+6$

системе счисления, определяются местом расположения в слове переменных $x_1 \ldots x_5$ соответственно.

Пусть POLE1 = 1101110110011011 = 156633; POLE2 = 0011011101000111 = =033507. При этом $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$, $x_5 = 1$, $y = 1$.

В табл. 1 приведены соотношения для подсчета числа команд и емкости памяти (в словах) для БП, реализуемых предлагаемым методом. Отметим, что, если оператор $y = 0$ разместить в начале программы, то вводить оператор ω не требуется. При этом БП строится непосредственно по ЛБГ (рис. 5) и за счет некоторого снижения быстродействия упрощается на одну команду.

2. Проверка правильности построения ЛБГ. Правильность построения ЛБГ может быть проверена несколькими способами. Первый из них состоит в восстановлении по ЛБГ соответствующей ему формулы и сравнении ее с формулой нормального вида, по которой ЛБГ строился. Метод получения формулы по ЛБГ, который называется ретрансляцией, изложен в [8]. Ретрансляция (чтение) осуществляется путем свертки ЛБГ, при которой на каждом шаге число условных вершин уменьшается на одну и тем самым формула восстанавливается за h шагов.

Рассмотрим другой, более интересный с теоретической точки зрения, метод чтения. Он базируется на просмотре БГ от выходов к входу и представлении каждой условной вершины в качестве мультиплексора «2 в 1». При этом выход i-й условной вершины, который при прямом просмотре ЛБГ является ее входом, обозначается символом φ_i, а преобразованне, осуществляемое этой вершиной, описывается соотношением $\varphi_i = \bar{x}_i\psi_0 \vee x_i\psi_1$, где $\psi_0(\psi_1)$ — обозначение вершины ЛБГ, связанной с входом i-й вершины и помеченной нулем (единицей). Этот метод может использоваться для произвольных ациклических БГ, однако для ЛБГ он открывает интересные свойства последних.

Пример 2. Выполнить анализ ЛБГ, представленного на рис. 3. При этом имеет место следующая система соотношений:

$$\varphi_5 = \bar{x}_5 0 \vee x_5 = x_5;\ \varphi_4 = \bar{x}_4\varphi_5 \vee x_4 0 = \bar{x}_4 x_5;\ \varphi_3 = \bar{x}_3 0 \vee x_3\varphi_4 = x_3\bar{x}_4 x_5;$$

$$\varphi_2 = \bar{x}_2\varphi_3 \vee x_2\varphi_5 = \bar{x}_2 x_3\bar{x}_4 x_5 \vee x_2 x_5 = (x_2 \vee x_3\bar{x}_4)\, x_5;$$

$$\varphi_1 = \bar{x}_1\varphi_3 \vee x_1\varphi_2 = \bar{x}_1 x_3\bar{x}_4 x_5 \vee x_1\,(x_2 \vee x_3\bar{x}_4)\, x_5 =$$

$$= (\bar{x}_1 x_3\bar{x}_4 \vee x_1 x_2 \vee x_1 x_3\bar{x}_4)\, x_5 = (x_1 x_2 \vee x_3\bar{x}_4)\, x_5 = y.$$

Таким образом, предложенный метод обеспечивает восстановление формулы последовательно по ее фрагментам, несмотря на наличие в общем случае скобок произвольной глубины: $\varphi_5 = x_5$; $\varphi_4 = \bar{x}_4 x_5$; $\varphi_3 = x_3\bar{x}_4 x_5$; $\varphi_2 = (x_2 \vee x_3\bar{x}_4)\, x_5$; $\varphi_1 = (x_1 x_2 \vee x_3\bar{x}_4)\, x_5$.

Этот подход был впервые изложен в [13], где он применялся в качестве нового метода вычисления булевых формул. Отметим, что при использовании обратной польской записи, которая обычно предлагается для вычисления формул, реализация осуществляется не по фрагментам, а по подформулам, что более сложно.

Перейдем к рассмотрению второго способа проверки правильности. Он состоит в подаче на ЛБГ каждого из 2^k различных наборов в стандартном порядке, формировании линейного массива из значений достигаемых при этом операторных вершин и сравнении его со столбцом значений таблицы истинности, построенной по реализуемой формуле. Изложенный подход требует перечисления всех 2^k наборов, что резко ограничивает размерность проверяемых графов.

Для упрощения проверки ЛБГ рассмотрим следующие характеристики: число и суммарную длину путей, которые наряду с известными показателями качества БП [1] определяют их логическую сложность. Исследование этих показателей позволит сформулировать третий подход для проверки правильности построения ЛБГ.

3. Перечисление путей в ЛБГ. Обычно в качестве тестов для проверки правильности программ используются пути в ее управляющем графе [14]. При этом поиск путей [15] осуществляется непосредственно по графу (возможно ошибочному), который предстоит проверить. Для устранения этого недостатка необходимо пути перечислять не по проверяемому графу, а по спецификации, по которой этот граф строился. Для представленного класса программ спецификацией является формула. Покажем, что задача перечисления путей в правильном графе может быть решена на основе преобразований заданной формулы. Рассмотрим два метода перечисления путей в ЛБГ по формуле.

Первый из них основан на преобразовании формулы и ее инверсии в ортогональную дизъюнктивную нормальную форму (ОДНФ) [16]. ОДНФ называется такая ДНФ, в которой произведение двух любых конъюнкций равно нулю. При ортогонализации дизъюнкция преобразуется на основе правила

$$f_1 \vee f_2 \vee \ldots \vee f_n = f_1 \vee \bar{f}_1 f_2 \vee \ldots \vee \bar{f}_1 \bar{f}_2 \ldots \bar{f}_{n-1} f_n.$$

ОДНФ заданной формулы перечисляет единичные пути в ЛГБ, а ОДНФ инверсии — нулевые [8].

Пример 3. Перечислить пути в ЛБГ по формуле $y = x_1 x_2 \vee x_3$. Для этого определим ОДНФ этой формулы и ее инверсии:

$$y = x_1 x_2 \vee x_3 = x_1 x_2 \vee \overline{x_1 x_2} x_3 = x_1 x_2 \vee (\bar{x}_1 \vee \bar{x}_2)\, x_3 =$$

$$= x_1 x_2 \vee (\bar{x}_1 \vee x_1 \bar{x}_2)\, x_3 = x_1 x_2 \vee \bar{x}_1 x_3 \vee x_1 \bar{x}_2 x_3;$$

$$\bar{y} = \overline{x_1 x_2 \vee x_3} = (\bar{x}_1 \vee \bar{x}_2)\, \bar{x}_3 = (\bar{x}_1 \vee x_1 \bar{x}_2)\, \bar{x}_3 = \bar{x}_1 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3.$$

Каждая из конъюнкций ОДНФ соответствует одному пути в ЛБГ. При этом ранг конъюнкции равен длине пути. Таким образом $S = S^0 + S^1 = 5$; $L = L^0 + L^1 = 12$, для S, S^0, S^1 — число всех, нулевых и единичных путей в ЛБГ, а L, L^0, L^1 — длины этих путей.

Использованный метод ортогонализации для сложных формул весьма трудоемок. С целью упрощения рассмотрим другой метод ортогонализации, позволяющий одновременно ортогонализовать формулу и ее инверсию, что эквивалентно одновременному перечислению единичных и нулевых путей в правильно построенном ЛБГ.

Преобразуем, используя разложение Шеннона, формулу и все ее подформулы слева направо, последовательно по всем переменным. Основная особенность предлагаемого метода состоит в том, что подформулы, получающиеся в результате разложения и умножаемые на ноль, не уничтожаются. В получаемой в результате полного разложения формуле раскрываются скобки. При этом пометка нулем (единицей) конъюнкции свидетельствует о принадлежности ее ОДНФ инверсной

Таблица 2

Покрытие минтермов путями

Путь	Выход БЛГ	Минтермы
1. x_1x_2	1	1. $x_1x_2\bar{x}_3$ 2. $x_1x_2x_3$
2. $\bar{x}_1x_3$	1	3. $\bar{x}_1\bar{x}_2x_3$ 4. $\bar{x}_1x_2x_3$
3. $x_1\bar{x}_2x_3$	1	5. $x_1\bar{x}_2x_3$
4. $\bar{x}_1x_3$	0	6. $\bar{x}_1\bar{x}_2\bar{x}_3$ 7. $\bar{x}_1x_2\bar{x}_3$
5. $x_1\bar{x}_2\bar{x}_3$	0	8. $x_1\bar{x}_2\bar{x}_3$

(прямой) формулы или о том, что этой конъюнкции соответствует нулевой (единичный) путь в ЛБГ.

Пример 4. Выполнить с помощью предлагаемого метода ортогонализацию формулы $y = x_1x_2 \vee x_3$. При этом $y = x_1x_2 \vee x_3 = \bar{x}_1 (x_3) \vee x_1 (x_2 \vee x_3) = \bar{x}_1 (\bar{x}_3 0 \vee \vee x_3 1) \vee x_1 (\bar{x}_2 x_3 \vee x_2 1) = \bar{x}_1 (\bar{x}_3 0 \vee x_3 1) \vee x_1 (x_2 (\bar{x}_3 0 \vee x_3 1) \vee x_2 1) = \bar{x}_1\bar{x}_3 0 \vee \bar{x}_1 x_3 1 \vee \vee x_1\bar{x}_2\bar{x}_3 0 \vee x_1\bar{x}_2 x_3 1 \vee x_1 x_2 1$. Этот же метод может быть использован и для перечисления путей в БГ, соответствующем повторной формуле.

Покажем, что только перечисление всех путей позволяет восстановить всю таблицу истинности, а меньшее их количество эту проблему не решает. Этот факт иллюстрируется табл. 2, в которой показано, какие минтермы восстанавливают каждый путь. Так как пути ортогональны, то они покрывают минтермы без пересечения. Из табл. 2 следует, что исключение хотя бы одного пути не позволит восстановить всю таблицу.

Тестирование по всем путям — третий метод проверки правильности построения БГ. Из изложенного следует чрезвычайно важный факт, что использование в качестве тестов путей, покрывающих все вершины или все ребра управляющего графа, является недостаточным.

4. Подсчет числа путей. Подсчет числа путей в БГ может осуществляться с помощью метода Флойда [15], который состоит в возведении в степень матрицы смежности заданного графа, что является весьма трудоемким. В силу того, что в рассматриваемой задаче интерес представляет только число путей от крайней условной вершины до операторных, можно воспользоваться более эффективным алгоритмом, предложенным Дейкстрой [15]. Однако и этот алгоритм в силу того, что он естественно не учитывает специфику графов БП, также весьма трудоемок. Поэтому для БГ рассмотрим два более эффективных метода.

С помощью первого из них подсчет осуществляется от выходов БГ к его входу, а второго — в обратном порядке. При этом первый метод обеспечивает подсчет общего числа путей, а второй — единичных и нулевых путей в отдельности.

Первый метод состоит в следующем: 1) в БГ устраняется пометка всех дуг; 2) входы единичной и нулевой операторных вершин помечаются символом «*I*»; 3) пометка входов условных вершин осуществляется последовательно от выходных вершин к первой таким образом, чтобы при определении пометки входа некоторой вершины пометка входов вершин, связанных с рассматриваемой, была уже выполнена. При этом вход каждой условной вершины помечается символом *l*, где *l* — сумма пометок вершин, связанных с рассматриваемой; 4) величина пометки входа первой условной вершины равна числу путей в БГ. На рис. 6

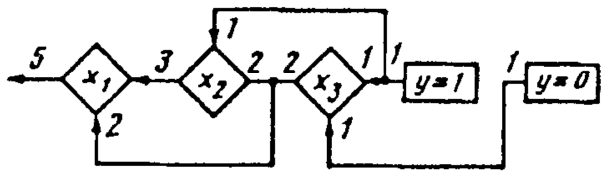

Рис. 6

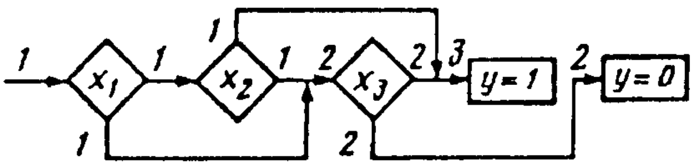

Рис. 7

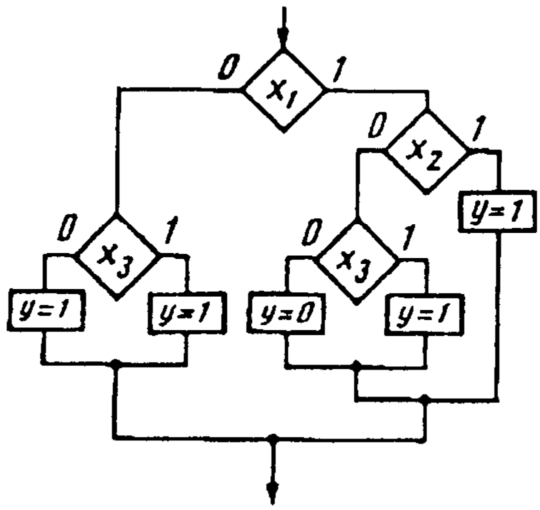

Рис. 8

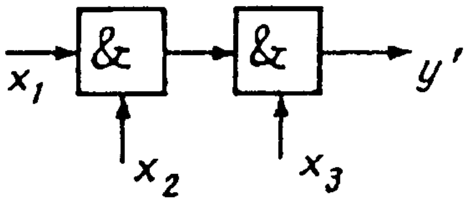

Рис. 9

приведен пример подсчета числа путей в ЛБГ, реализующем формулу $y =$ $= x_1x_2 \vee x_3$. Из рис. 6 следует, что $S = 5$.

Второй метод, идея которого предложена Акерсом [17], состоит в следующем 1) в БГ устраняется пометка всех дуг; 2) вход БГ помечается символом «I» 3) пометка каждого выхода рассматриваемой условной вершины совпадает с пометко входа этой вершины; 4) в каждой точке объединения ребер справедлив анало закона Кирхгофа: пометка ребра, исходящего из точки объединения, равна сумм пометок ребер, входящих в эту точку; 5) величина пометки на входе единично (нулевой) операторной вершины равна числу единичных (нулевых) путей в БГ На рис. 7 приведен пример подсчета «единичных» и «нулевых» путей в ЛБГ реализующем формулу $y = x_1x_2 \vee x_3$. Из рисунка следует, что $S^1 = 3$, $S^0 = 2$.

Отметим интересный факт, состоящий в том, что при использовании второго метода величина пометки на входе некоторой вершины в ЛБГ равна числу вершин этого типа в структурированном методом дублирования БГ, соответствующем рассматриваемому ЛБГ [9]. Из рис. 7 следует, что в соответствующем структурированном БГ (рис. 8) имеется одна вершина x_1, одна вершина x_2, две вершины x_3, три вершины $y = 1$ и две вершины $y = 0$.

Определяемые вторым методом пометки по Акерсу могут применяться и для другой цели — определения величины весов переменных и порога для положительно монотонных бесповторных пороговых формул (ПМ БПФ). Формула в базисе И и ИЛИ является ПМБПФ, если она реализуется в виде каскада из двухвходовых элементов И и ИЛИ [17]. При этом чем ближе к выходу каскада размещается входная переменная, тем большим весом она обладает.

Размещая входные переменные в порядке неубывания весов, построим ЛБГ. При этом величина пометки по Акерсу на входе каждой условной вершины равна весу переменной, помечающей эту вершину, а порог равен числу нулевых путей в ЛБГ.

Если для пороговой функции $y = f(x_1, \ldots, x_i, \ldots, x_k)$ найдены веса и порог $(p_1, \ldots, p_i, \ldots, p_k; T)$, то для функции $y(x_1, \ldots, \bar{x}_i, \ldots, x_k)$ эти параметры имеют вид $(p_1, \ldots, -p_i, \ldots, p_k; T - p_i)$, а для функции двойственной y — вид $(p_1, \ldots, p_i, \ldots, p_k; \sum_{i=1}^{k} p_i - T + 1)$ [18].

Пример 5. Проверить формулу $y = x_3 \vee \bar{x}_1 x_2$ на пороговость и в случае положительного решения определить ее веса и порог. Заменим формулу однотипной и положительно монотонной $y' = x_3 \vee x_1 x_2$. Формула реализуется каскадом (рис. 9) и, следовательно, является пороговой. Из рассмотрения каскада следует, что $p_1 = p_2 < p_3$. Перепишем формулу y' в порядке неубывания весов: $y'' = x_1 x_2 \vee x_3$. Построим для этой формулы ЛБГ (рис. 1) и определим пометки по Акерсу (рис. 7). Таким образом формула y'' имеет следующие параметры: $p_1' = p_2' = 1, p_3 = 2, T' = S^0 = 2$. Следовательно, для формулы y справедливо $p_1 = -1,\ p_2 = 1,\ p_3 = 2;\ T = T' - p_1' = 1$.

Выше был изложен метод подсчета числа путей для случая, когда каждая условная вершина соответствует одной переменной. Условной вершине в общем случае может соответствовать формула, что приводит к необходимости модификации этого метода для блочных бинарных графов (ББГ). Модифицированный метод состоит из следующих этапов: 1) в БГ устраняется пометка всех дуг; 2) единичный (нулевой) выход i-й условной вершины помечается числом $\zeta = p_i\,(q_i)$, соответствующим количеству единичных (нулевых) путей, порождаемых этой вершиной; 3) каждая дуга, исходящая из i-й условной вершины, помечается символом v_i. При этом $v_i = \zeta\, j$, где j — число, помечающее дугу, входящую в i-ю условную вершину (при $i = 1$, $j = 1$); 4) каждая дуга, входящая в вершину, помечается символом j, где $j = \sum_{i=1}^{t} v_i$, а t — число дуг, входящих в рассматриваемую вершину.

Пример 6. Пусть задан блочный ЛБГ (БЛБГ), реализующий формулу $y = x_1 x_2 \vee x_3$ (рис. 1), где $x_1 = z_1 z_2\ (p_1 = 1,\ q_1 = 2)$, $x_2 = z_3 z_4 \vee z_5 z_6\ (p_2 = 3,\ q_2 = 4)$; $x_3 = z_7 z_8 z_9\ (p_3 = 1,\ q_3 = 3)$. Осуществим пометку ребер этого графа и определим число единичных и нулевых путей в нем с помощью предложенного метода (рис. 10). Из рис. 10 следует, что $S^1 = 9$, $S^0 = 18$.

Выше были изложены методы подсчета числа путей по графу. Рассмотрим методы подсчета числа путей в ЛБГ и БЛБГ по реализуемой формуле на основе двух методов ортогонализации, рассмотренных выше.

Первый метод базируется на понятии ортогональной скобочной формулы (ОСФ). ОСФ строится по скобочной формуле в базисе И, ИЛИ, НЕ за счет ортогонализации дизъюнкций (конъюнкции ортогонализации не требуют).

Для формулы, соответствующей графу, и ее инверсии определяются ОСФ. В этих ОСФ знаки дизъюнкции (конъюнкции) заменяются знаками сложения (умножения), символ i-й переменной (ее инверсии) заменяется числом, равным количеству единичных (нулевых) путей, соответствующих этой переменной. Величина, подсчитанная по преобразованной ОСФ формулы (ее инверсии), определяет число единичных (нулевых) путей в графе.

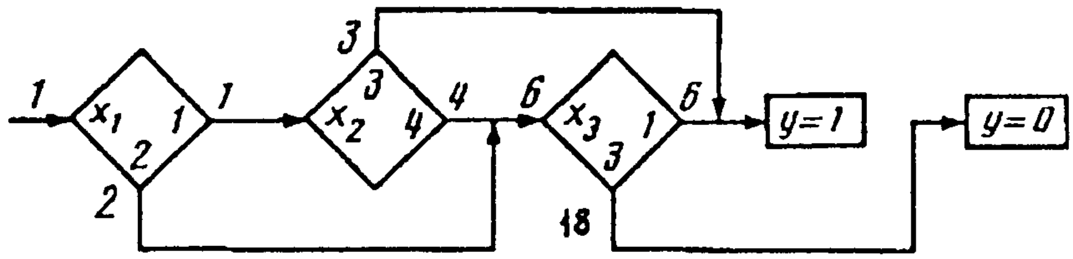

Рис. 10

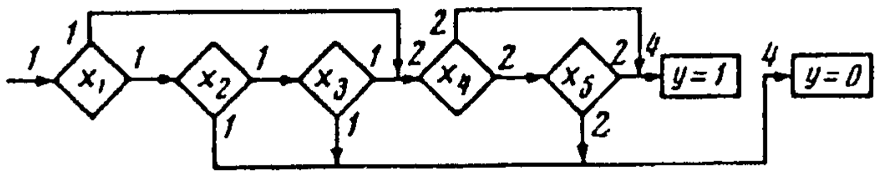

Рис. 11

Пример 7. Пусть задана формула $y = z_1 \vee z_2 (z_3 \vee z_4)$, соответствующая БЛБГ. Переменные этой формулы определяются следующим образом: $z_1 = x_1x_2$; $z_2 = x_3x_4 \vee x_5x_6$; $z_3 = x_7x_8x_9$; $z_4 = x_{10} (x_{11} \vee x_{12})$. Число единичных и нулевых путей в ЛБГ этих формул соответственно равно: $p_1 = 1$, $q_1 = 2$; $p_2 = 3$, $q_2 = 4$; $p_3 = 1$, $q_3 = 3$; $p_4 = 2$, $q_4 = 2$. ОСФ формулы и ее инверсии имеют вид $y = z_1 \vee \bar{z}_1z_2 (z_3 \vee \bar{z}_3z_4)$; $\bar{y} = \bar{z}_1 (\bar{z}_2 \vee z_2\bar{z}_3\bar{z}_4)$. Выполняя указанные выше замены, получим $S^1 = p_1 + q_1p_2 (p_3 + q_3p_4) = 43$; $S^0 = q_1 (q_2 + p_2q_3q_4) = 44$.

Второй метод базируется на ортогонализации формулы на основе разложения Шеннона. Выполнив разложение формулы и всех подформул по всем переменным, получим ортогональное скобочное выражение (ОСВ). Исключим в ОСВ нули и единицы и осуществим замены, указанные в предыдущем методе. Величина, подсчитанная по преобразованной ОСВ, определяет суммарное число путей в графе. Используем этот метод для задачи, рассмотренной в предыдущем примере. При этом ОСВ формируется следующим образом: $y = z_1 \vee z_2 (z_3 \vee z_4) =$ $= \bar{z}_1z_2 (z_3 \vee z_4) \vee z_1 1 = \bar{z}_1 (\bar{z}_2 0 \vee z_2 (z_3 \vee z_4)) \vee z_1 1 = \bar{z}_1 (\bar{z}_2 0 \vee z_2 (\bar{z}_3 z_4 \vee z_3 1)) \vee z_1 1 =$ $= \bar{z}_1 (\bar{z}_2 0 \vee z_2 (\bar{z}_3 (\bar{z}_4 0 \vee z_4 1) \vee z_3 1)) \vee \bar{z}_1 1$.

Выполняя указанные исключения и замены, получим $S = q_1 (q_2 +$ $+ p_2 (q_3 (q_4 + p_4) + p_3)) + p_1 = 87$.

5. Вычисление показателей, характеризующих быстродействие ЛБГ. В [1] предложен рекуррентный метод вычисления средней длины путей на всех наборах по реализуемой формуле. Ниже предлагаются соотношения для вычисления этой величины по ЛБГ или одной из аналитической форм перечисления путей, рассмотренных выше.

Для подсчета суммарной длины путей в ЛБГ можно воспользоваться равенством $L = \sum_{i=1}^{k} iN_i$, где N_i — число путей длины i, а для подсчета суммарной длины путей на всех наборах соотношением — $L^* = \sum_{i=1}^{k} iN_i 2^{k-i}$. Для использования этих формул необходимо определить длины всех путей. При этом количество нулевых и единичных путей является важным ориентиром для контроля перечисления всех путей. После этого определяются значения N_i и вычисляются значения L и L^*.

Пример 8. Вычислим L и L^* для формулы $y = (x_1 \vee x_2x_3)(x_4 \vee x_5)$, рассмотренной в [1]: 1) построим ЛБГ (рис. 11); 2) $S^1 = 4$, $S^0 = 4$. Перечислим длины единичных (нулевых) путей: 2, 3, 4, 5 (2, 3, 3, 5); 3) $N_1 = 0$, $N_2 = 2$, $N_3 = 3$, $N_4 = 1$, $N_5 = 2$; 4) $L = 1 \cdot 0 + 2 \cdot 2 + 3 \cdot 3 + 4 \cdot 1 + 2 \cdot 5 = 27$; $L^* = 1 \cdot 0 \cdot 2 + 2 \cdot 2 \cdot 2 + 3 \cdot 3 \cdot 2 + 4 \cdot 1 \cdot 2 + 5 \cdot 2 \cdot 2 = 86$.

Из соотношений для L и $L^{м}$ следует, что

$$l_{ср} = \frac{1}{S}\sum_{i=1}^{h} iN_i, \quad l^{м}_{ср} = \frac{1}{2^h}\sum_{i=1}^{h} iN_i 2^{h-i}.$$

Для рассмотренного примера $l_{ср} = 27/8$; $l^{м}_{ср} = 86/32$.

В [1] получена нижняя оценка величины $l^{м}_{ср}$, равная $2 - 1/2^{h-1}$. При этом $L^{м} = 2^{h+1} - 2$. Эти оценки достигаются на бесповторных пороговых формулах при оптимальном порядке их записи, что не было указано в [1]. Для бесповторных непороговых формул значения этих величин больше. Так, в [1] показано, что для класса ДНФ, в которых число конъюнкций равно их рангу, $\sqrt{h} \leq l^{м}_{ср}$.

Так как непороговые формулы существуют лишь при $h \geq 4$, то при $h = 1$ $L^{м} = 2$, $l^{м}_{ср} = 1$; $h = 2$ $L^{м} = 6$, $l^{м}_{ср} = 1.5$; $h = 3$ $L^{м} = 14$, $l^{м}_{ср} = 1.75$; $h = 4$ $30 \leq L^{м} \leq 32$, $1,875 \leq l^{м}_{ср} \leq 2$.

В заключение рассмотрим способ вычисления $l^{м}_{ср}(P)$ для случая, когда для j-й условной вершины заданы вероятности пребывания ее в состояниях «0» ($P^j(0)$) и «1» ($P^j(1)$). При этом для каждого пути вычисляется вероятность прохождения его. Эта величина умножается на длину пути. Суммируя полученные значения по всем путям, получим величину $l^{м}_{ср}(P)$.

Для формулы, рассмотренной в примере 3, вычисление $l^{м}_{ср}(P)$ осуществляется следующим образом: $l^{м}_{ср}(P) = P^1(1)P^2(1)\cdot 2 + P^1(0)P^3(1)\cdot 2 + P^1(1)P^2(0)P^3(1)\cdot 3 + P^1(0)P^3(0)\cdot 2 + P^1(1)P^2(0)P^3(0)\cdot 3$.

В случае, если $P^j(0) = P^j(1) = 0.5$ ($j = 1, 2, \ldots, h$), то $l^{м}_{ср}(P) = l^{м}_{ср}$. Изложенный способ близок к рассмотренному в [19, 20].

СПИСОК ЛИТЕРАТУРЫ

1. *Кузнецов О. П.* О программной реализации логических функций и автоматов//АиТ. 1977. № 7, 9.
2. *Девятков В. В.* Программная реализация управляющих алгоритмов//Автоматизированное проектирование дискретных управляющих устройств. М.: Наука, 1980.
3. *Пупырев Е. И.* Перестраиваемые автоматы и микропроцессорные системы. М.: Наука, 1984.
4. *Малюгин В. Д.* Реализация булевых функций арифметическими полиномами//АиТ. 1982. № 4.
5. *Кузьмин В. А.* Оценка сложности реализации функций алгебры логики простейшими видами бинарных программ//Методы дискретного анализа в теории кодов и схем. Вып. 29. Новосибирск: Наука СО, 1976.
6. *Иванов Г. И.* О применении частичных подстановок для временного разложения булевых функций в обобщенные граф-схемы алгоритмов//АиТ. 1983. № 11.
7. *Лапкин Л. Я.* Оценки сложности программной реализации при различных формах задания булевых функций//АиТ. 1987. № 1.
8. *Кузнецов Б. П., Шалыто А. А.* Система преобразований некоторых форм представления булевых функций//АиТ. 1985. № 11.
9. *Кузнецов Б. П., Шалыто А. А.* Структурный подход к программной реализации булевых формул//АиТ. 1985. № 5.
10. *Сагалович Ю. Л., Шалыто А. А.* Бинарные программы и их реализация асинхронными автоматами//ППИ. 1987. Вып. 1.
11. *Артюхов В. Л., Копейкин Г. А., Шалыто А. А.* Настраиваемые модули для управляющих логических устройств. Л.: Энергоатомиздат, 1981.
12. *Камынин С. С., Любимский Э. З., Шура-Бура М. Ю.* Об автоматизации программирования при помощи программирующей программы//Проблемы кибернетики. Вып. 1. М.: Физматгиз, 1958.

13. *Артюхов В. Л., Шалыто А. А.* Судовые управляющие логические системы. Л.: Ин-т повышения квалификации руководящих работников и специалистов судостроительной пром-сти, 1984.
14. *Липаев В. В.* Качество программного обеспечения. М.: Финансы и статистика, 1983.
15. *Свами М., Тхуласирами К.* Графы, сети и алгоритмы. М.: Мир, 1984.
16. *Рябинин И. А.* Основы теории и расчета надежности судовых электроэнергетических систем. Л.: Судостроение, 1971.
17. *Akers B.* Binary decision diagrams//IEEE Trans. on computers. 1978. № 6.
18. *Артюхов В. Л., Розенблюм Л. Я., Шалыто А. А.* Логические возможности некоторых типов каскадных структур//Сети связи и дискретные устройства управления. М.: Наука, 1976.
19. *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.
20. *Лаговиер Б. А.* Об алгоритмах программной реализации автоматов//АиТ. 1979. № 1.

Москва

Поступила в редакцию
28.IV.1994

РОССИЙСКАЯ АКАДЕМИЯ НАУК ISSN 0002-3388

Известия Академии наук

техническая КИБЕРНЕТИКА

1994 5