ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ им. М. В. КЕЛДЫША
РОССИЙСКОЙ АКАДЕМИИ НАУК

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. ЛОМОНОСОВА
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

МАТЕРИАЛЫ VII МОЛОДЕЖНОЙ НАУЧНОЙ ШКОЛЫ ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ И ЕЕ ПРИЛОЖЕНИЯМ

(Москва, 18–23 мая 2009 г.)

ЧАСТЬ II

Москва 2009

МАТЕРИАЛЫ
VII МОЛОДЕЖНОЙ НАУЧНОЙ ШКОЛЫ
ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ И ЕЕ ПРИЛОЖЕНИЯМ

(Москва, 18–23 мая 2009 г.)

ЧАСТЬ II

Москва 2009

М34
УДК 519.7

Издание осуществлено при поддержке Российского фонда фундаментальных исследований по проекту 09–01–06027

М34 Материалы VII молодежной научной школы по дискретной математике и ее приложениям (Москва, 18–23 мая 2009 г.). Часть II. Под редакцией А. В. Чашкина. 2009.— 54 с.

Сборник содержит материалы VII молодежной научной школы по дискретной математике и ее приложениям, проходившей в Москве с 18 по 23 мая 2009 г. при поддержке Российского фонда фундаментальных исследований (проект 09–01–06027). Для студентов, аспирантов и научных работников в области дискретной математики и математической кибернетики.

Научное издание

МАТЕРИАЛЫ
VII МОЛОДЕЖНОЙ НАУЧНОЙ ШКОЛЫ
ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ И ЕЕ ПРИЛОЖЕНИЯМ
(Москва, 18–23 мая 2009 г.)

Под общей редакцией А. В. ЧАШКИНА

Ответственный за выпуск *А. Д. Яшунский*

СОДЕРЖАНИЕ

О ПРОГРАММЕ ПОСТРОЕНИЯ ДИЗЪЮНКТИВНЫХ НОРМАЛЬНЫХ ФОРМ

Е. Ю. Комарова (Москва)

1. Введение

В настоящей работе представлена программа построения дизъюнктивных нормальных форм (ДНФ). Эта программа также решает задачи распознавания выполнимости формул логики высказываний, построения моделей и эквивалентных преобразований формул алгебры логики.

2. Описание алгоритма

Исходная формула φ задается в базисе $\{\neg, \wedge, \vee, \to\}$ и представляется в виде описанной ниже таблицы, рассмотренной в работе [3]. Такие таблицы представляют собой реализацию схем из функциональных элементов, используемых в работе [1], для более эффективной работы с формулами.

Программа построения ДНФ для формулы φ состоит из четырех этапов:

1. Переход к базису $\{\neg, \wedge, \vee\}$.
2. Пронос отрицаний внутрь формулы с помощью правил де Моргана.
3. Преобразование формулы к виду дизъюнкции «элементарных конъюнкций специального вида» (конъюнкт может содержать некоторую пропозициональную переменную вместе с её отрицанием, а также повторяющиеся литеры).
4. Удаление противоречивых и повторяющихся конъюнкций, а также повторяющихся множителей в непротиворечивых конъюнкциях.

Рассмотрим подробнее алгоритм. Сама исходная формула задаётся в виде таблицы с k строками. Каждая запись может иметь следующий вид:

$$[m]\ p$$
$$[m]\ ,\neg, m_1$$
$$[m]\ ,\text{operator}, m_1, m_2$$

Здесь p — пропозициональная переменная, m — номер текущей строки, m_1, m_2 — номера строк, в которых записаны подформулы исходной формулы, operator — один из следующих операторов: $\vee, \wedge, \to$.

На первом этапе просматривается таблица и выполняются эквивалентные преобразования — замена подформул, содержащих импликации $(\psi_1 \to \psi_2)$ на подформулы с отрицанием и дизъюнкцией $((\neg\psi_1) \vee \psi_2)$. Строка с записью вида $[m]\ , \to, m_1, m_2$ заменяется на две строки: $[m]\ , \vee, m_1, m_2$ и $[k+1]\ , \neg, m_1$.

Строка	Замена
$[m]\ , \to, m_1, m_2$	$[m]\ , \vee, m_1, m_2$
	$[k+1]\ , \neg, m_1$

Таким образом, после первого этапа мы переходим к базису $\{\neg, \wedge, \vee\}$. Поскольку он является полным, то мы можем любую формулу логики высказываний представить в нём (см. [1–3]).

На втором этапе мы проносим отрицания вглубь формулы так, чтобы они стояли над пропозициональными переменными. В соответствии с этим возможны следующие случаи:

1. | Строки | Замена |
   |---|---|
   | $[m]\ , \neg, m_1$ | $[m]\ \psi$ |
   | $[m_1]\ , \neg, m_2$ | |
   | $[m_2]\ \psi$ | |

2. | Строки | Замена |
   |---|---|
   | $[m]\ , \neg, m_1$ | $[m]\ , \vee, m_1, k+1$ |
   | $[m_1]\ , \wedge, m_3, m_2$ | $[m_1]\ , \neg, m_2$ |
   | | $[k+1]\ , \neg, m_3$ |

3. | Строки | Замена |
   |---|---|
   | $[m]\ , \neg, m_1$ | $[m]\ , \wedge, m_1, k+1$ |
   | $[m_1]\ , \vee, m_3, m_2$ | $[m_1]\ , \neg, m_2$ |
   | | $[k+1]\ , \neg, m_3$ |

На этом этапе необходимо также учитывать то, что на строку m_1 могут быть ссылки из других строк таблицы, а это значит, что изменение или удаление этой строки приведёт к изменению исходной формулы. Это будет неэквивалентным преобразованием, что недопустимо. Во избежание возможной ошибки перед преобразованием проверяется наличие в таблице ссылки на данную строку. В случае ее существования содержимое строки m_1 добавляется в конец таблицы, а в строке со ссылкой заменяется номер строки на новый.

На третьем этапе раскрываются скобки, если они имелись в исходной формуле, и таким образом она преобразуется к виду дизъюнкции элементарных конъюнкций специального вида. Под элементарным конъюнктом

специального вида мы понимаем конъюнкт, быть может также содержащий некоторую пропозициональную переменную вместе с её отрицанием, повторяющиеся пропозициональные переменные.

Подформулы вида $((x_1 \vee x_2) \wedge x_3)$ преобразуем к виду $((x_1 \wedge x_3) \vee (x_2 \wedge x_3))$, что в табличном виде записывается следующим образом:

Строки	Замена
$[m]\ , \wedge, m_1, m_2$	$[m]\ , \vee, m_1, k+1$
$[m_1]\ , \vee, m_3, m_4$	$[m_1]\ , \wedge, m_3, m_2$
	$[k+1]\ , \wedge, m_4, m_2$

Аналогично выполняется преобразование и для подформул, имеющих вид $(x_3 \wedge (x_1 \vee x_2))$.

Как и на втором этапе необходимо учитывать то, что на строку m_1 могут быть ссылки из других строк таблицы. И если они есть, то содержимое строки m_1 следует скопировать в конец таблицы и в строке со ссылкой заменить номер строки m_1 на новый.

На четвертом этапе выполняются упрощения — удаляются противоречивые конъюнкции, а также повторяющиеся множители в противоречивых конъюнктах. В результате выполнения этого этапа создаётся список, состоящий из списков элементарных конъюнкций, который будет соответствовать итоговой ДНФ (этот список может быть пустым). На этом этапе также выполняется упрощение ДНФ в соответствии с правилами поглощения.

По завершении работы программы могут быть получены следующие результаты: формула невыполнима, либо формула выполнима. Если формула выполнима, то модель для нее можно получить, если взять любой конъюнкт из списка, содержащего ДНФ, и означить соответствующим образом в нём пропозициональные переменные. (В качестве результата может быть также общезначимость формулы.)

Реализуемый программой алгоритм является корректным и полным. Это следует из того, что на каждом его шаге выполняются только эквивалентные преобразования (см. [2]), поэтому данная программа может также использоваться для выполнения эквивалентных преобразований формул.

Программа достаточно просто допускает дальнейшее расширение базиса, в котором представлены исходные формулы такими операциями, как $\oplus$, $\sim$, $|$, $\downarrow$, $\not\to$.

Статистика работы программы на случайных формулах приведена в следующей таблице. Время работы программы измеряется в миллисекундах.

Число переменных в формуле	5	10	30
Число разобранных форул	100	100	100
Число выполнимых формул	78	65	71
Число невыполнимых формул	22	35	29
Среднее число строк до разбора	21,6	73,4	150,7
Среднее число строк после разбора (с сокращ.)	60,7	2297,9	396,6
Среднее число строк после разбора (без сокращ.)	94,5	3591,7	332,1
Среднее время разбора (без сокращ.)	5,1	1907,3	189,2
Среднее время разбора (с сокращ.)	10,5	1658,4	204,3

Список литературы

1. Ложкин С. А. Лекции по основам кибернетики: Учебное пособие. — М.: Издательский отдел Факультета ВМиК МГУ им. М.В. Ломоносова, 2004.

2. Яблонский С. В. Элементы математической кибернетики: Учебник // С.В.Яблонский — М.: Высш. школа, 2007.

3. Хелемендик Р. В. Элементы математической логики и возможности ее применения // Учебное пособие. М.: МАТИ, 2009. В печати.

О МОНОТОННЫХ ЗАМКНУТЫХ КЛАССАХ ФУНКЦИЙ МНОГОЗНАЧНОЙ ЛОГИКИ С БЕСКОНЕЧНОЙ НАДСТРУКТУРОЙ

В. Б. Ларионов (Москва)

Данная работа посвящена изучению решетки замкнутых классов k-значной логики. Введем необходимые определения.

Обозначим множество $E_k = \{0, 1, \ldots, k-1\}$. Будем использовать следующие стандартные обозначения. Множество всех функций k-значной логики обозначим P_k. Для любого подмножества $A \subseteq P_k$ через $[A]$ будем обозначать замыкание относительно операции суперпозиции (для функций далее везде будет идти речь именно об этом типе замыкания).

Пусть на E_k задано некоторое отношение частичного порядка r. Возьмем два произвольных набора $\tilde{a} = (a_1, \ldots, a_n)$ и $\tilde{b} = (b_1, \ldots, b_n)$ из E_k^n. Будем говорить, что $\tilde{a}$ не превосходит $\tilde{b}$ относительно частичного порядка r и записывать $\tilde{a} \leqslant_r \tilde{b}$, если для любого $1 \leqslant i \leqslant n$ справедливо неравенство $a_i \leqslant_r b_i$.

Определение. Функция $f(x_1, \ldots, x_n)$ называется *монотонной относительно частичного порядка* r, если для любых двух наборов $\tilde{a}, \tilde{b} \in E_k^n$ таких, что $\tilde{a} \leqslant_r \tilde{b}$, выполнено $f(\tilde{a}) \leqslant_r f(\tilde{b})$. Множество всех функций из P_k, монотонных относительно r, называется *монотонным классом* M_r.

Для краткости мы будем задавать частичный порядок r частично упорядоченным множеством (ЧУМ) H из элементов E_k, а соответствующий монотонный класс обозначать M_H.

В 1965 году Розенберг описал все предполные классы функций в k-значных логиках, $k \geqslant 3$ ([1]). Было показано, что они образуют шесть семейств: C, B, S, M, U, L. Детальное описание этих семейств можно найти в [2].

Семейство M является подмножеством множества классов монотонных функций. Известно ([3]), что монотонный класс является предполным (принадлежит множеству M) тогда и только тогда, когда частичный порядок, сохраняемый им, обладает в точности одним минимальным и одним максимальным элементом. В связи с этим возникает вопрос о том, какое положение в решетке замкнутых классов занимают остальные монотонные классы. А именно, может ли бесконечное множество замкнутых классов находиться над каким-либо из монотонных классов. В работе [4], показано, что существует обширное множество монотонных классов, обладающих указанным свойством. В данной статье доказывается, что подобный класс существует в четырёхзначной логике и не существует в меньших. Также расширяется множество монотонных классов с бесконечной надструктурой.

Будем считать известным понятие сохранения функцией предиката. Для некоторого предиката R через A_R обозначим замкнутый класс функций из P_k, сохраняющих R.

Далее будут использоваться некоторые факты из теории Галуа о соответствии между замкнутыми классами предикатов и функций.

На множестве всех предикатов определяется три операции: конъюнкция, отождествление переменных и взятие проекции по переменной (добавление квантора существования). Размер данной публикации не позволяет привести полные определения указанных операций, их можно найти в [5]. На основе этих операций стандартным образом вводится понятие формулы.

Пусть T — некоторая система предикатов. Замыканием $[T]$ системы T называется множество предикатов, реализуемых всевозможными формулами над T.

Отметим, что если $R_1 \in [R_2]$, то $A_{R_2} \subseteq A_{R_1}$ ([5]).

Если предикат p получен из R при помощи указанных трех операций, то $A_R \subseteq A_p$ (это доказано в [5]).

Пусть $R(x_1, x_2)$ — предикат, задающий некоторый монотонный класс M_r ($R(x_1, x_2) = TRUE$ тогда и только тогда, когда $x_1 \leqslant_r x_2$). Определим следующие предикаты:

$$R_i(x_1, \dots, x_{2i+2}) = \exists y_1, \dots, y_{2i+1} R(x_1, y_1) \& R(y_2, y_1) \& R(x_2, y_2) \&$$

$$\& R(x_3, y_2) \& R(y_2, y_3) \& R(y_4, y_3) \& R(x_4, y_4) \& R(x_5, y_4) \& R(y_4, y_5) \& \dots \&$$

$$\& R(y_{2i}, y_{2i-1}) \& R(x_{2i}, y_{2i}) \& R(x_{2i+1}, y_{2i}) \& R(y_{2i}, y_{2i+1}) \& R(x_{2i+2}, y_{2i+1}).$$

Из сказанного выше следует, что $M_r \subseteq A_{R_i}$.

Определение. Определим T как множество, состоящее из всех ЧУМ L, которые содержат подмножество H, состоящее из четырех элементов: $0, 1, 2, 3$. Элементы $0, 3$ несравнимы между собой и больше двух несравнимых элементов $1, 2$. Причем в L не появляются пути из 0 в 3 по элементам, являющимся максимумами 1 и 2 (под максимумом тут и далее мы подразумеваем элемент, больший указанных). Уточним это понятие: не существует последовательности элементов $z_1, \dots, z_m$ в L такой, что $z_1 = 0$, $z_m = 3$, z_i сравнимо с z_{i+1} $(i = 1, \dots, m-1)$ (т.е. либо $z_i \leqslant z_{i+1}$, либо $z_{i+1} \leqslant z_i$), все z_i — максимумы 1 и 2.

Теорема 1. *Над монотонным классом, сохраняющим любое ЧУМ из определенного выше множества T, находится бесконечная цепочка замкнутых классов.*

Объем данной статьи позволяет дать лишь набросок доказательства. Рассмотрим логику P_4 и ЧУМ H. Пусть предикат R задаёт указанное ЧУМ, R_i — предикаты построенные из R по приведённым выше формулам.

Определим функции f_i следующим образом:

$$f_i = \left| \begin{array}{ccccccccccc} x_1 & x_2 & x_3 & x_4 & \dots & x_{i-3} & x_{i-2} & x_{i-1} & x_i & & f(\tilde{x}) \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & & 2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 3 & & 2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 3 & 3 & & 2 \\ 0 & 0 & 0 & 0 & \dots & 1 & 3 & 3 & 3 & & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & 1 & \dots & 3 & 3 & 3 & 3 & & 2 \\ 0 & 0 & 1 & 3 & \dots & 3 & 3 & 3 & 3 & & 2 \\ 0 & 1 & 3 & 3 & \dots & 3 & 3 & 3 & 3 & & 2 \\ 1 & 3 & 3 & 3 & \dots & 3 & 3 & 3 & 3 & & 2 \\ 3 & 3 & 3 & 3 & \dots & 3 & 3 & 3 & 3 & & 3 \\ & & & \multicolumn{5}{c}{\text{на остальных наборах}} & & & 1 \end{array} \right| \qquad (1)$$

Рассмотрим произвольный набор $\tilde{a} \in E_4^{2h+2}$ $(h \geqslant 1)$. Разрежем его на $h+2$ куска: первая компонента $\{a_1\}$, h кусков из двух компонент $\{a_{2(s+1)}, a_{1+2(s+1)}\}$ $(s = 0, \ldots, h-1)$, последняя компонента $\{a_{2h+2}\}$. Будем говорить, что $\tilde{a} \in T_h$, тогда и только тогда, когда один кусок или два соседних куска $\tilde{a}$ содержат 0 и 3 или между кусками с этими значениями идут только куски со значениями $1, 2$ (с точностью до порядка) каждый (имеется в виду, что в каждом таком куске встречаются сразу и 1, и 2).

Лемма 1. *$R_h(\tilde{b}) = TRUE$ тогда и только тогда, когда $\tilde{b} \notin T_h$.*

Лемма 2. *$f_i \notin A_{R_i}$, $i \geqslant 2$.*

Доказательство. Обозначим наборы из E_4^i, на которых функция f_i равна 2, через $\tilde{d}_1, \ldots, \tilde{d}_i$ (в том порядке, в котором они сверху вниз записаны в таблице (1)). Набор, на котором функция f_i равна 0 (соответственно, равна 3) обозначим через $\tilde{d}_0$ (соответственно, $\tilde{d}_{i+1}$).

Обозначим через $\tilde{c}_j$ набор из E_4^i $(2, 2, \ldots, 2, 1, 2, \ldots, 2)$, где 1 стоит на j-м месте $(j = 1, \ldots, i)$.

Рассмотрим матрицу размера $2i+2$ на i, в которой в строках сверху вниз записаны следующие наборы: $\tilde{d}_0, \tilde{c}_1, \tilde{d}_1, \tilde{c}_2, \tilde{d}_2, \ldots, \tilde{c}_i, \tilde{d}_i, \tilde{d}_{i+1}$. Если мы применим функцию f_i к каждой строке указанной матрице, то получим столбец $\overline{f} = [01212\ldots123]$, принадлежащий множеству T_i. Первые i столбцов $\overline{x}_1, \ldots, \overline{x}_n$ таблицы (1) не принадлежат множеству T_i, следовательно, по лемме 1 получаем, что $R_i(\overline{x}_j) = TRUE$, $j = 1, \ldots, i$, а $R(\overline{f}) = FALSE$. Откуда $f_i \notin A_{R_i}$.

Лемма доказана.

Лемма 3. *$f_i \in A_{R_{i-1}}$, $i > 2$.*

Доказательство. Предположим, что мы взяли какие-то $2(i-1)+2$ строки из таблицы (1) для f_i. Обозначим получившиеся по вертикали векторы $\overline{x}_1, \ldots, \overline{x}_i$ и $\overline{f}$. Предположим, что $R_{i-1}(\overline{f}) = FALSE$. По лемме 1 вектор $\overline{f} \in T_{i-1}$. Это означает (см. определение T_h), что вектор $\overline{f}$ после куска с 0 содержит пары со значениями $1, 2$ и потом кусок с 3. Отметим, что по определению предиката R_{i-1}, число указанных двоек s не превосходит $i-1$. Не ограничивая общности, будем предполагать, что указанный выше 0 (соответственно, 3) содержится в разряде вектора $\overline{f}$ с номером h_0 (соответственно, h_{2s+1}), а пары $1, 2$ — в разрядах с номерами $h_1, \ldots, h_{2s}$. При этом будем считать, что в парах единицы идут впереди двоек (иначе мы можем перенумеровать переменные).

Покажем, что найдётся столбец $\overline{x}_j$ такой, что $R_{i-1}(\overline{x}_j) = FALSE$.

Рассмотрим, номера каких строк таблицы (1) соответствуют номерам разрядов вектора $\overline{f}$ при указанной выше выборке $2(i-1)+2$ строк. Очевидно, что строки с номерами 1 и $i+2$ соответствуют разрядам h_0 и h_{2s+1}, поскольку функция f_i принимает значение 0 (соответственно, 3) только на одном наборе. Если разряду h_2 (первая двойка рассматриваемого участка вектора $\overline{f}$) соответствует не вторая строка, то $R_{i-1}(\overline{x}_i) = FALSE$ (поскольку в соседних кусках вектора $\overline{x}_i$ будут 0 и 3). Разряду h_4 (вторая двойка рассматриваемого участка вектора $\overline{f}$) по тем же причинам может соответствовать либо вторая, либо третья строка таблицы (1) (иначе в соседних кусках вектора $\overline{x}_{i-1}$ будут 0 и 3), и так далее. В конце получим, что разряду h_{2s} (последняя двойка рассматриваемого участка вектора $\overline{f}$) могут соответствовать только строки с номерами $2, \ldots, s+1 \leqslant i$ (то есть только наборы $\tilde{d}_1, \ldots \tilde{d}_{i-1}$). С другой стороны, разряду h_{2s} может соответствовать только строка с номером $i+1$, иначе в соседних кусках вектора $\overline{x}_1$ будут 0 и 3. Получаем, что в каком-то из векторов $\overline{x}_1, \ldots, \overline{x}_i$ будет нарушено условие из определения множества T_i.

Итак, берём в левой части таблицы наборы $\overline{x}_1, \ldots, \overline{x}_i$ такие, что выполняется $R_{i-1}(\overline{x}_j) = TRUE$. Получаем, что соответствующий столбец справа (столбец значений) $\overline{f}$ удовлетворяет $R_{i-1}(\overline{f}) = TRUE$ (иначе согласно приведённым выше рассуждениям нашёлся бы столбец слева, на котором предикат ложен).

Лемма 4. $A_{R_i} \subseteq A_{R_{i-1}}$, $i \geqslant 2$.

Для ЧУМ, содержащих множество H, доказательство теоремы получается аналогично.

Теорема 2. *Минимальной логикой с монотонным классом с бесконечной надструктурой является* P_4.

В работе [4] показано, что в P_k для $k \leqslant 5$ не существует монотонных классов с бесконечной надструктурой, образованных ЧУМ с одним минимальным или одним максимальным элементом. Поэтому для доказательства последней теоремы достаточно воспользоваться тем фактом, что остальные монотонные классы в P_3 (имеющие более одного минимума и максимума) являются предпредполными.

Работа выполнена при поддержке РФФИ, грант 09-01-00701.

Список литературы

1. Rosenberg I. La structure des fonctions de plusieur variables sur un ensemble fini // Comptes Rendus, de l'Academ. — Paris. — 1965. — 260. — 3817-3819.

2. Яблонский С. В., Гаврилов Г. П., Набебин А. А. Предполные классы в многозначных логиках — М.: Издательский дом МЭИ, 1997.

3. Мартынюк В. В. Исследование некоторых классов функций в многозначных логиках // Проблемы кибернетики. — 1960. — вып. 3. — С. 49–61.

4. Ларионов В. Б. О положении некоторых классов монотонных k-значных функций в решетке замкнутых классов. // МАТЕРИАЛЫ XVII Международной школы-семинара «СИНТЕЗ И СЛОЖНОСТЬ УПРАВЛЯЮЩИХ СИСТЕМ» имени академика О. Б. Лупанова (Новосибирск, 27 октября–1 ноября 2008 г.) — Издательство Института математики. — Новосибирск. —2008. —с. 90-95.

5. Марченков С. С. Замкнутые классы булевых функций —М.: Физматлит, 2000.

О МИНИМАЛЬНЫХ СЛОЖНЫХ ЭЛЕМЕНТАХ РЕШЕТКИ НАСЛЕДСТВЕННЫХ КЛАССОВ ГРАФОВ

Д. С. Малышев (Нижний Новгород)

1. Введение

К настоящему времени накоплено огромное количество результатов о полиномиальной разрешимости и NP-полноте различных задач на графах во многих классах графов. Способы получения новых сведений такого рода могут быть самими разнообразными, но можно выделить два распространенных подхода:

1. Поиск более широких «простых» классов, объемлющих ранее известные.

2. Поиск NP-полных сужений для известных «сложных» случаев.

Вместе с тем, при рассмотрении представительных семейств классов графов можно ставить задачи более общего характера, чем анализ сложности для индивидуального класса. В частности, можно поставить целью выявление пределов, до которых возможны расширения полиномиальной сложности и сужения с «противоположным» сложностным статусом. Тем самым, речь фактически идет о нахождении границы между «простыми» и «сложными» классами из рассматриваемого семейства. В данной публикации исследуется эта граница для некоторых задач на графах в семействе

наследственных классов графов, т.е. классов графов, замкнутых относительно удаления вершин.

Формализуем понятия «простого» и «сложного» класса графов. Пусть П — какая-либо NP-полная задача на графах. Наследственный класс графов назовем П-*простым*, если задача П для графов из этого класса полиномиально разрешима, и П-*сложным* в противном случае. Далее везде предполагаем справедливость неравенства P≠NP и не включаем его явно в формулировки полученных результатов.

Естественной идеей решения задачи демаркации является поиск *максимальных* П-*простых* и *минимальных* П-*сложных классов*, т.е. тупиковых классов графов соответствующей сложности из рассматриваемой решетки. К сожалению, использование понятия максимального простого класса графов оказывается безрезультатным. Так, В. Е. Алексеев в работе [1] установил, что ни один П-простой класс не является максимальным простым (правда, в [1] это утверждается только про задачу о независимом множестве, но все рассуждения из данной работы легко переносятся на общий случай). Вместе с тем, до недавнего времени про минимальные сложные классы ничего не было известно.

Первый результат о подобного рода классах был получен автором в работе [2]. Там рассматривалась задача распознавания принадлежности наследственному классу графов $\mathbf{X}$ (задача РП[$\mathbf{X}$]), и было доказано следующее утверждение.

Теорема 1. *Для любого класса графов* $\mathbf{X}$ *ни один* РП[$\mathbf{X}$]-*сложный класс графов не является минимальным* РП[$\mathbf{X}$]-*сложным.*

Поскольку классические NP-полные при $k > 2$ задачи о вершинной k-раскраске и о реберной k-раскраске являются переформулировками задачи распознавания принадлежности наследственному классу графов, то они — примеры задач, для которых нет минимальных сложных классов.

В той же работе [2] были найдены минимальные сложные классы графов для некоторой модификации классической задачи о раскраске. Речь идет о задаче о вершинном списковом ранжировании (задаче ВСР), впервые введенной в работе [3]. Постановка этой задачи состоит в следущем: задан граф G и список $L^{V(G)} = \{L^{V(G)}(v_1), L^{V(G)}(v_2), \ldots, L^{V(G)}(v_n)\}$, где $V(G) = \{v_1, v_2, \ldots, v_n\}$. Множество $L^{V(G)}(v_i)$ (называемое *палитрой цветов вершин графа* G) является конечным множеством номеров разрешенных цветов для вершины v_i. Всюду далее номер цвета будет отождествляться с самим цветом, поэтому на множестве цветов из $L^{V(G)}$ естественным образом индуцируется отношение «быть больше». $L^{V(G)}$-*раскраской* называется такое отображение $c : V(G) \to \bigcup\limits_{i=1}^{n} L^{V(G)}(v_i)$, что выполняются

следующие два условия:

(1) Для любой вершины x число $c(x)$ принадлежит множеству $L^{V(G)}(x)$.

(2) Каждый путь, соединяющий две одноцветные вершины u и v, содержит такую вершину w, что $c(w) > c(u)$.

Задача BCP для данного графа G и палитры цветов вершин $L^{V(G)}$ состоит в том, чтобы определить, имеет ли граф $L^{V(G)}$-раскраску.

В работе [2] рассматривались *кометы*, т.е. графы, получаемые отождествлением центральной вершины звезды с одной из концевых вершин простого пути. Класс **Comet** — наследственное замыкание множества комет. Значение этого класса графов раскрывает следуюший результат работы [2].

Теорема 2. *Класс* **Comet** *является минимальным* BCP-*сложным.*

Установлению минимальности некоторого класса графов для задачи BCP посвящена оставшаяся часть настоящей работы.

2. Вспомогательный результат

Пусть S_i — результат подразбиения каждого ребра графа $K_{1,i}$, а **Star** — наследственное замыкание класса $\bigcup\limits_{i=1}^{\infty} S_i$.

Лемма 1. *Класс* **Star** — BCP-*сложный.*

Доказательство. Доказательство основано на сведении задачи о вершинном списковом ранжировании в классе деревьев высоты не более чем два (обозначаемом далее **THTree**) к той же задаче в классе **Star**. Пусть $G \in$ **THTree**. Корень дерева G будем обозначать через r, его нелистовых потомков обозначим через $x_1, x_2, \ldots, x_p$. Для любого $i \in \{1, 2, \ldots, p\}$ под множеством $\{y_1^{(i)}, y_2^{(i)}, \ldots, y_{j_i}^{(i)}\}$ будем понимать множество потомков вершины x_i. В работе [3] доказано, что класс **THTree** является сложным для задачи о вершинном списковом ранжировании даже для палитр цветов вершин специального вида (называемых далее *упрощенными*). Это такие палитры $L^{V(G)}$, у которых множества

$$L^{V(G)}(r), L^{V(G)}(y_1^{(1)}), \ldots, L^{V(G)}(y_{j_1}^{(1)}),$$
$$L^{V(G)}(y_1^{(2)}), \ldots, L^{V(G)}(y_{j_2}^{(2)}), \ldots, L^{V(G)}(y_1^{(p)}), \ldots, L^{V(G)}(y_{j_p}^{(p)})$$

имеют мощность один и для которых

$$L^{V(G)}(x_1) = \{\alpha_1, \beta_1\}, L^{V(G)}(x_2) = \{\alpha_2, \beta_2\}, \ldots, L^{V(G)}(x_p) = \{\alpha_p, \beta_p\}.$$

Данные палитры также обладают тем свойством, что $L^{V(G)}(r)$ — наименьший среди цветов из $L^{V(G)}$ и тем свойством, что для любого $i \in \{1, 2, \ldots, p\}$ и любого $j \in \{1, 2, \ldots, j_i\}$ цвет из $L^{V(G)}(y_j^{(i)})$ больше α_i и меньше β_i.

Рассмотрим сужение задачи ВСР в классе **THTree** на палитры описанного выше вида. Пусть $G \in$ **THTree** и пусть $L^{V(G)}$ — упрощенная палитра цветов вершин этого графа. Можно считать, что разность любых двух различных цветов из $L^{V(G)}$ не меньше чем 2 (это предположение не уменьшает общности, поскольку в противном случае можно умножить каждый из цветов палитры на 2). Рассмотрим вершину x_1 графа G. Множество потомков x_1 произвольным образом разделим на два множества A, B, мощности которых отличаются не более чем на 1. Построим по графу G и палитре $L^{V(G)}$ граф G' и палитру $L^{V(G')}$ следующим образом. Удалим из G вершину x_1 и добавим вершины x_1', x_1'', x_1'''. Добавим также ребра $(r, x_1'), (r, x_1''), (r, x_1''')$, ребра, инцидентные x_1' и всевозможным вершинам из A, а также ребра, инцидентные x_2'' и всевозможным вершинам из B. Для любой вершины $x \in V(G) \cap V(G')$ выполняется равенство $L^{V(G)}(x) = L^{V(G')}(x)$. Пусть $\alpha = \alpha_1 + 1$ и $\beta = \beta_1 - 1$. Положим $L^{V(G')}(x_1') = \{\alpha_1, \beta\}, L^{V(G')}(x_1'') = \{\alpha, \beta_1\}, L^{V(G')}(x_1''') = \{\alpha, \beta\}$. Покажем, что $L^{V(G)}$-раскраска существует тогда и только тогда, когда существует $L^{V(G')}$-раскраска.

Предположим, что существует $L^{V(G')}$-раскраска. Возможны только следующие три случая:

1.1. $L^{V(G')}$-раскраска такова, что $c(x_1') = \alpha_1, c(x_1'') = \beta_1, c(x_1''') \in \{\alpha, \beta\}$. Окрасим вершину x_1 в цвет β_1, цвет остальных вершин из $V(G) \setminus \{x_1\}$ совпадает с их цветом в $L^{V(G')}$-раскраске. Легко проверить, что построенное таким образом отображение является $L^{V(G)}$-раскраской.

1.2. $L^{V(G')}$-раскраска такова, что $c(x_1') = \alpha_1, c(x_1'') = \alpha, c(x_1''') = \beta$. Окрасим вершину x_1 в цвет α_1, а цвет остальных вершин из $V(G) \setminus \{x_1\}$ совпадает с их цветом в $L^{V(G')}$-раскраске. Легко проверить, что построенное таким образом отображение является $L^{V(G)}$-раскраской.

1.3. $L^{V(G')}$-раскраска такова, что $c(x_1') = \beta, c(x_1'') = \beta_1, c(x_1''') = \alpha$. Этот случай симметричен предыдущему.

Предположим, что существует $L^{V(G)}$-раскраска. Возможны только два случая:

2.1. $L^{V(G)}$-раскраска такова, что $c(x_1) = \alpha_1$. Построение $L^{V(G')}$-раскраски выполняется обратным образом к случаю 1.2.

2.2. $L^{V(G)}$-раскраска такова, что $c(x_1) = \beta_1$. Построение $L^{V(G')}$-раскраски выполняется обратным образом к случаю 1.3.

Итак, $L^{V(G)}$-раскраска существует тогда и только тогда, когда существует $L^{V(G')}$-раскраска, при этом палитра $L^{V(G')}$ остается упрощенной. Применяя описанную выше дихотомию достаточное число раз, мы получаем граф $G^* \in$ **Star**. Ясно, что вопрос о существовании $L^{V(G^*)}$-раскраски эквивалентен вопросу о существовании $L^{V(G)}$-раскраски. Заметим, что входные данные задачи ВСР для графа G^* ограничены сверху полиномом от входных данных той же задачи для графа G. Отсюда следует, что задача ВСР в классе **THTree** полиномиально сводима к той же задаче в классе **Star**. Лемма доказана.

3. Основной результат

Основным результатом данной публикации является следующая

Теорема 3. *Класс* **Star** — *минимальный* ВСР-*сложный.*

Доказательство. Предыдущая лемма означает, что класс **Star** является ВСР-сложным. Докажем его минимальность, доказав, что для любого графа G из **Star** класс **Star** $\cap\, Free(\{G\})$ является ВСР-простым.

Граф G является порожденным подграфом графа S_i для некоторого i. Поэтому справедливо включение **Star** $\cap\, Free(\{G\}) \subseteq$ **Star** $\cap\, Free(\{S_i\})$. Очевидно, что произвольная компонента связности любого графа из класса **Star** $\cap\, Free(\{S_i\})$ имеет не более чем i нелистовых вершин. В работе [3] доказано, что для любого фиксированного k класс лесов, содержащих не более чем k нелистовых вершин, является ВСР-простым. Отсюда следует, что для любого $G \in$ **Star** задача ВСР в классе графов **Star** $\cap\, Free(\{G\})$ полиномиально разрешима. Теорема доказана.

Список литературы

1. Alekseev V.E. On easy and hard hereditary classes of graphs with respect to the independent set problem // Discrete Applied Mathematics. — 2004. — V. 132. — P. 17—26.

2. Малышев Д.С. О минимальных сложных классах графов // Дискретный анализ и исследование операций (направлено к публикации).

3. Dereniowski D. The complexity of list ranking of trees // Ars Combinatoria. — 2008. — V. 86. — P. 97—114.

О РАНГЕ НЕЯВНЫХ ПРЕДСТАВЛЕНИЙ НАД ОДНИМ КЛАССОМ ФУНКЦИЙ ТРЕХЗНАЧНОЙ ЛОГИКИ

Е. В. Михайлец (Москва)

Понятие неявной выразимости функций k-значной логики было введено А. В. Кузнецовым как одно из обобщений понятия выразимости функций суперпозициями [1].

Рассмотрим некоторую систему функций k-значной логики. Обозначим ее через A, $A \subseteq P_k$. *Системой неявных уравнений* над системой функций A будем называть всякую систему уравнений вида

$$\begin{cases} \Phi_1(x_1, \ldots, x_n, y) = \Psi_1(x_1, \ldots, x_n, y), \\ \ldots \\ \Phi_q(x_1, \ldots, x_n, y) = \Psi_q(x_1, \ldots, x_n, y), \end{cases}$$

где левые и правые части уравнений представляют собой суперпозиции над системой функций A, т. е. $\Phi_i, \Psi_i \in [A]$, $1 \leqslant i \leqslant q$.

Будем говорить, что функция $f(x_1, \ldots, x_n)$ k-значной логики *неявно выразима* над системой функций A, если существует система неявных уравнений над A указанного вида, имеющая при любых фиксированных значениях $x_1, \ldots, x_n$ единственное решение $y = f(x_1, \ldots, x_n)$. При этом соответствующую систему уравнений будем называть *неявным представлением* функции $f(x_1, \ldots, x_n)$ над A.

Множество всех функций f, $f \in P_k$, неявно выразимых над системой функций A, назовем *неявным расширением* системы A и будем обозначать через $I(A)$ [2]. Непосредственно из определения неявного представления следует, что $[A] \subseteq I(A)$. Таким образом, неявное расширение является одним из обобщений операции замыкания по суперпозиции.

Если неявное расширение системы A содержит все функции k-значной логики, т. е. $I(A) = P_k$, то систему A называют *неявно полной* в P_k.

Изначально понятие неявной выразимости над различными классами функций k-значной логики было введено А. В. Кузнецовым [1]. В дальнейшем исследования в этой области были продолжены в работах О. М. Касим-Заде. В работе [2] О. М. Касим-Заде дано полное решение проблемы неявной выразимости и неявной полноты в P_2. В P_3 проблема неявной полноты решена Е. А. Ореховой. В ее работе [3] описаны двадцать семь минимальных неявно полных замкнутых классов функций в P_3 и доказано, что произвольная система функций трехзначной логики неявно полна тогда и

только тогда, когда ее замыкание по суперпозиции содержит хотя бы один из данных минимальных неявно полных классов.

Решив проблему неявной полноты, естественно поставить вопрос об изучении метрических характеристик неявных представлений функций над неявно полными замкнутыми классами.

Рассмотрим произвольную функцию f из неявного расширения некоторой системы A функций k-значной логики, $f \in I(A)$. Следуя [4], назовем *рангом* функции f над системой A и будем обозначать через $m_A(f)$ наименьшее число уравнений, достаточное для построения неявного представления f над A.

Далее, введем функцию Шеннона $m_A(n) = \max m_A(f)$, называемую *ранговой функцией* системы A (максимум берется по всем функциям k-значной логики, принадлежащим неявному расширению системы A и существенно зависящим не более чем от n переменных).

О. М. Касим-Заде в работе [4] получил оценки роста ранговой функции для всех замкнутых классов булевых функций. Из результатов работы [4] следует, что для любого неявно полного замкнутого класса в P_2 ранговая функция имеет линейный порядок роста.

Автором было исследовано поведение ранговых функций для некоторых неявно полных классов функций в P_k. Выяснилось, что для широкого диапазона неявно полных систем функций k-значной логики ранговые функции имеют линейный порядок роста $\Theta(n)$. В частности, для классов функций, монотонных относительно произвольного частичного порядка, заданного на множестве E_k, $E_k = \{0, 1, \ldots, k-1\}$, и содержащего хотя бы одну пару сравнимых элементов, выражение для ранговой функции определяется следующей теоремой.

Теорема 1 [5]. *Пусть $k \geqslant 2$ и на множестве E_k задан частичный порядок $\mathfrak{M}$, содержащий хотя бы одну пару сравнимых элементов. Пусть A — класс всех функций в P_k, монотонных относительно частичного порядка $\mathfrak{M}$. Тогда система функций A неявно полна в P_k и при всех натуральных n для ранговой функции $m_A(n)$ имеет место равенство*

$$m_A(n) = \left\lceil \frac{n+2}{2} \right\rceil .$$

В связи с полученными результатами возник вопрос, существует ли при $k \geqslant 3$ неявно полная система функций в P_k, у которой порядок роста ранговой функции выше линейного.

Для ответа на данный вопрос автором было исследовано поведение ранговых функций для всех минимальных неявно полных классов в P_3, описанных Е. А. Ореховой [3].

Функция $g(x_1, \ldots, x_n)$ k-значной логики называется *двойственной* функции $f(x_1, \ldots, x_n) \in P_k$ относительно некоторой подстановки σ на k-элементном множестве $E_k = \{0, 1, \ldots, k-1\}$, если при любых значениях переменных $x_1, \ldots, x_n$ выполняется соотношение

$$\sigma(f(x_1, \ldots, x_n)) = g(\sigma(x_1), \ldots, \sigma(x_n)).$$

Множество всех функций, двойственных функциям некоторого замкнутого по суперпозиции класса Σ относительно фиксированной подстановки σ, также является замкнутым классом и называется *двойственным* классу Σ относительно этой подстановки. В дальнейшем, называя функции или классы функций двойственными, будем иметь ввиду, что они двойственны относительно некоторой нетождественной подстановки.

Несложно показать, что ранговые функции двойственных друг другу классов функций k-значной логики тождественно равны. Таким образом, при поиске ранговых функций достаточно ограничиться рассмотрением различных с точностью до двойственности минимальных неявно полных классов в P_3.

В P_3 существует шесть различных попарно не двойственных минимальных неявно полных замкнутых классов. Данные шесть классов функций в совокупности со всеми классами, двойственными им, исчерпывают двадцать семь минимальных неявно полных классов, описанных Е. А. Ореховой в работе [3].

Автором было исследовано поведение ранговой функции для всех шести классов. Для двух из них удалось получить экспоненциальные нижние и верхние оценки ранговой функции, для оставшихся четырех классов порядок роста ранговой функции оказался линейным.

Сформулируем результаты, касающиеся экспоненциальных оценок роста ранговой функции.

Для задания функций одной и двух переменных в P_3 будем использовать таблицы значений [3]. Рассмотрим систему функций, заданных следующими таблицами:

$\max(x,y)$			$\min_{01}(x,y)$			$w(x)$	0	1
0	1	2	0	0	2	0	0	1
1	1	2	0	1	2	0	0	1
2	2	2	2	2	2	1	0	1

Обозначим замыкание данной системы по суперпозиции через W_1. Е. А. Ореховой [3] доказана неявная полнота в P_3 системы функций W_1. В работе [6] автора опубликован следующий результат.

Теорема 2. *Для ранговой функции системы W_1 при всех натуральных n справедливы оценки:*

$$2^{(n+1)/2} - \frac{1}{2} \leqslant m_W(n) \leqslant 2 \cdot 3^n.$$

Далее, сформулируем основной результат данной работы — экспоненциальную оценку для ранговой функции указанного ниже неявно полного класса в P_3.

Рассмотрим следующую систему функций трехзначной логики:

$\max(x,y)$			$\min_{01}(x,y)$			$r(x)$	0	1
0	1	2	0	0	2	2	0	1
1	1	2	0	1	2	2	0	1
2	2	2	2	2	2	0	0	1

Ее замыкание по суперпозиции будем обозначать через W_2. Класс функций W_2 неявно полон в P_3 [3] и для него справедливо следующее утверждение.

Теорема 3. *Для ранговой функции системы функций W_2 при всех натуральных n имеют место оценки:*

$$2^n \leqslant m_W(n) \leqslant n2^n.$$

Таким образом, получена качественная картина поведения ранговой функции для всех минимальных неявно полных замкнутых классов в P_3.

Автор выражает благодарность своему научному руководителю О. М. Касим-Заде за всестороннее внимание к данной работе.

Работа выполнена при финансовой поддержке РФФИ (проект №08–01–00863), программы «Ведущие научные школы РФ» (проект НШ–4470.2008.1) и программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения» (проект «Задачи оптимального синтеза управляющих систем»).

Список литературы

1. Кузнецов А. В. О средствах для обнаружения невыводимости или невыразимости // Логический вывод. М.: Наука, 1979. С. 5–33.

2. Касим-Заде О. М. О неявной выразимости булевых функций // Вестник МГУ. Серия 1. Математика. Механика. 1996. № 2. С. 44–49.

3. Орехова Е. А. Об одном критерии неявной полноты в трехзначной логике // Математические вопросы кибернетики. Вып. 12. М.: Наука. Физматлит, 2003. С. 27–74.

4. Касим-Заде О. М. Об одной метрической характеристике неявных и параметрических представлений булевых функций // Математические вопросы кибернетики. Вып. 6. М.: Наука. Физматлит, 1996. С. 133–188.

5. Михайлец Е. В. О ранге неявных представлений над классами монотонных функций k-значной логики // Материалы VI Молодежной научной школы по дискретной математике и ее приложениям. Ч. II. М. 2007. С. 26–29.

6. Михайлец Е. В. О ранге неявных представлений над одним классом функций трехзначной логики // Вестник МГУ. Серия 1. Математика. Механика. 2008. № 5. С. 65–70.

О НЕКОТОРЫХ КЛАССАХ, ПОРОЖДЕННЫХ ОДНОСЛОЙНЫМИ СИММЕТРИЧЕСКИМИ ФУНКЦИЯМИ МНОГОЗНАЧНОЙ ЛОГИКИ

А. В. Михайлович (Москва)

В работе изучаются свойства замкнутых классов функций многозначной логики. Рассматривается задача о существовании базисов для некоторых семейств замкнутых классов.

В [1] показано, что все замкнутые классы булевых функций имеют конечный базис. На случай многозначных логик этот результат не распространяется. В [2] показано, что при $k \geqslant 3$ в P_k существуют замкнутые классы как со счетным базисом, так и классы, не имеющие базиса. В [3] рассмотрены некоторые семейства замкнутых классов, порожденные симметрическими функциями, и для них приведены критерии базируемости и конечной порожденности. В [4] получен аналогичный критерий базируемости для классов функций k-значной логики, $k \geqslant 3$, порождающие системы которых содержатся в множествах, обладающих некоторыми специальными свойствами. В данной работе изучается подмножество NS_k^1 множества всех симметрических однослойных функций из P_k. Показывается, что оно обладает этими свойствами, и доказываются критерии базируемости и конечной порожденности для классов, порожденных функциями из NS_k^1. Все необходимые определения можно найти в [3 – 5].

Будем говорить, что множество $\mathrm{A} \subseteq P_k$, $k \geqslant 3$ обладает свойством $(*)$, если для любого $G \subseteq \mathrm{A}$ выполняется равенство $(\cup[\{g\}]) \cap \mathrm{A} = [\cup\{g\}] \cap \mathrm{A}$, где объединения берутся по всем функциям g из множества G.

Пусть $f, g \in \mathrm{A} \subseteq P_k$, $k \geqslant 3$. Будем говорить, что функция f не превосходит функцию g относительно отношения $\preceq$ (обозначение $f \preceq g$), если

$f \in [\{g\}]$. Будем говорить, что функции f и g эквивалентны (обозначение $f \sim g$), если $f \preceq g$ и $g \preceq f$. Пусть $f(x_1, \ldots, x_n)$, $g(x_1, \ldots, x_m) \in \mathrm{A}$ $n, m \geqslant 1$. Будем говорить, что функция f не превосходит функцию g относительно отношения $\trianglelefteq_{\mathrm{A}}$ (обозначение $f \trianglelefteq_{\mathrm{A}} g$), если существует формула Φ над A, реализующая функцию f, и подформула формулы Φ, имеющая вид $g(\mathcal{B}_1, \ldots, \mathcal{B}_n)$, где $\mathcal{B}_1, \ldots, \mathcal{B}_n$ — некоторые формулы над $\mathfrak{A}$. Очевидно, что если $f \preceq g$, то $f \trianglelefteq_{\mathrm{A}} g$.

Будем говорить, что множество $\mathrm{A} \subseteq P_k$, $k \geqslant 3$, обладает свойством $(**)$, если для любых $f, g \in \mathrm{A}$, таких, что $f \trianglelefteq_{\mathrm{A}} g$ и $g \trianglelefteq_{\mathrm{A}} f$, выполняется соотношение $f \sim g$ и для любых функций $f, g, h \in \mathrm{A}$, таких, что $f \trianglelefteq_{\mathrm{A}} g$, $g \trianglelefteq_{\mathrm{A}} h$, $f \preceq h$, выполняется по крайней мере одно из следующих соотношений: $f \preceq g$, $g \preceq h$.

Пусть $n \geqslant 1$, $k \geqslant 3$. Множество всех наборов из E_k^n, которые получаются друг из друга перестановкой компонент, будем называть слоем. Слой из $\{1, 2, \ldots, k-1\}^n$, содержащий e единиц, d_1 двоек, d_2 троек, $\ldots$, d_{k-2} символов $k-1$, обозначим через $\mathcal{L}(e, d_1, \ldots, d_{k-2})$, $0 \leqslant e \leqslant n$, $0 \leqslant d_i \leqslant n$, $i = 1, \ldots, k-2$. Обозначим через R_k множество всех функций из P_k, принимающих значения только из множества $\{0, 1\}$ и равных нулю на единичном наборе и на всех наборах, содержащих хотя бы один нуль. Пусть $f(x_1, \ldots, x_n) \in R_k$. Положим $N_f = \{\widetilde{\alpha} \in E_k^n | f(\widetilde{\alpha}) = 1\}$. Обозначим через NS_k^1 множество всех таких функций $f(x_1, \ldots, x_n)$ из R_k, что для некоторых чисел e, $d_1, \ldots, d_{k-2} \in \mathbb{N} \cup \{0\}$, среди которых встречается по крайней мере два числа, отличных от нуля, выполняется равенство $N_f = \mathcal{L}(e, d_1, \ldots, d_{k-2})$.

Отметим следующие свойства функций из множества NS_k^1.

Утверждение 1. *Пусть $f(x_1, \ldots, x_n)$ — произвольная функция из NS_k^1, $k \geqslant 3$, $n \geqslant 1$, $\Phi(x_1, \ldots, x_n)$ — формула над NS_k^1, реализующая функцию f, Φ_1 — подформула формулы Φ, имеющая вид $g(\mathcal{B}_1, \ldots, \mathcal{B}_m)$, где $g \in \mathrm{NS}_k^1$, а $\mathcal{B}_1, \ldots, \mathcal{B}_m$ — формулы над NS_k^1. Тогда среди формул $\mathcal{B}_1, \ldots, \mathcal{B}_m$ есть символы переменных, причем символ каждой переменной из множества $\{x_1, \ldots, x_n\}$ встречается одинаковое число раз.*

Следствие 1. *Пусть $f(x_1, \ldots, x_n)$, $g(x_1, \ldots, x_m)$ — произвольные функции из NS_k^1, такие, что $f \trianglelefteq_{\mathrm{NS}_k^1} g$ и $n, m \geqslant 1$. Тогда выполняется неравенство $n \leqslant m$.*

Теорема 1 [4]. *Пусть $\mathrm{A} \subseteq P_k$, $k \geqslant 3$, множество A обладает свойствами $(*)$ и $(**)$, G — произвольное множество, состоящее из попарно неэквивалентных функций множества A, $F = [G]$. Тогда класс F имеет базис тогда и только тогда, когда каждая функция из G содержится в некоторой ограниченной максимальной цепи множества G.*

Теорема 2. *Пусть G — произвольное множество, состоящее из попарно неконгруэнтных функций множества NS_k^1, $k \geqslant 3$, а $F = [G]$. Тогда выполняются следующие утверждения.*

(1) *Класс F имеет базис тогда и только тогда, когда каждая функция из G содержится в некоторой ограниченной максимальной цепи множества G относительно $\preceq$.*

(2) *Класс F имеет конечный базис тогда и только тогда, когда множество G конечно.*

Доказательство. Пусть G — множество, состоящее из попарно неконгруэнтных функций множества NS_k^1, $k \geqslant 3$, $F = [G]$.

Покажем сначала, что множество NS_k^1 обладает свойством $(*)$. Пусть H — подмножество множества NS_k^1. Тогда положим $\mathrm{A} = (\cup[\{f\}]) \cap \mathrm{NS}_k^1$ и $\mathrm{B} = [\cup\{f\}] \cap \mathrm{NS}_k^1$, где объединения берутся по всем функциям f из множества H. Очевидно, что $\mathrm{A} \subseteq \mathrm{B}$. Покажем, что выполняется обратное включение.

Пусть $f(x_1, \ldots, x_n) \in \mathrm{B}$, Φ — некоторая формула над H, реализующая функцию f. Пусть Φ_1 — простая подформула формулы Φ, имеющая вид $g(x_{i_1}, \ldots, x_{i_m})$, $g \in H$. Пусть $N_f = \mathcal{L}(e, d_1, \ldots, d_{k-2})$, $N_g = \mathcal{L}(a, b_1, \ldots, b_{k-2})$, e, d_i, a, $b_i \in \mathbb{N}$, $i = 1, \ldots, k-2$. В силу утверждения 1 существует число $t \in \mathbb{N}$, такое, что среди $x_{i_1}, \ldots, x_{i_m}$ каждая переменная из множества $\{x_1, \ldots, x_n\}$ встречается t раз. Пусть $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n) \in N_f$. Поскольку Φ — формула над R_k, то выполняется равенство $\Phi_1(\widetilde{\alpha}) = 1$. Кроме того, в силу равенств $g(\alpha_{i_1}, \ldots, \alpha_{i_m}) = \Phi_1(\widetilde{\alpha}) = 1$, выполняется соотношение $(\alpha_{i_1}, \ldots, \alpha_{i_m}) \in N_g$. Следовательно, $a = te$, $b_i = td_i$, $i = 1, \ldots, k-2$. Тогда

$$f(x_1, \ldots, x_n) = g(\underbrace{x_1, \ldots, x_1}_{t}, \underbrace{x_2, \ldots, x_2}_{t}, \ldots, \underbrace{x_n, \ldots, x_n}_{t}),$$

т.е. $f \in [\{g\}] \subseteq \mathrm{A}$. Поэтому $\mathrm{A} = \mathrm{B}$.

Покажем теперь, что множество NS_k^1 обладает свойством $(**)$.

Пусть $f(x_1, \ldots, x_n), g(x_1, \ldots, x_p) \in \mathrm{NS}_k^1$, $f \trianglelefteq_{\mathrm{NS}_k^1} g$ и $g \trianglelefteq_{\mathrm{NS}_k^1} f$. Так как $f \trianglelefteq_{\mathrm{NS}_k^1} g$ и $g \trianglelefteq_{\mathrm{NS}_k^1} f$, то в силу следствия 1 выполняются неравенства $n \leqslant p$ и $p \leqslant n$. Поэтому $n = p$. Поскольку $f \trianglelefteq_{\mathrm{NS}_k^1} g$, то существуют формула Φ над NS_k^1, реализующая функцию f, и подформула Φ_1 формулы Φ, имеющая вид $g(\mathcal{B}_1, \ldots, \mathcal{B}_n)$, где $\mathcal{B}_1, \ldots, \mathcal{B}_n$ — формулы над NS_k^1. В силу утверждения 1 каждая переменная из множества $\{x_1, \ldots, x_n\}$ встречается среди $\mathcal{B}_1, \ldots, \mathcal{B}_n$. Поэтому Φ_1 является простой подформулой формулы Φ. Без ограничения общности будем считать, что Φ_1 имеет вид $g(x_1, \ldots, x_n)$. Поскольку Φ является формулой над R_k, то для каждого набора $\widetilde{\alpha}$ из N_f выполняется равенство $\Phi_1(\widetilde{\alpha}) = 1$, то есть $g(\widetilde{\alpha}) = 1$. Таким образом, $N_f \subseteq N_g$.

Аналогичным образом можно показать, что из соотношения $g \trianglelefteq_{\mathrm{NS}_k^1} f$ следует включение $N_g \subseteq N_f$. Поэтому $N_f = N_g$. А значит, $f \sim g$.

Пусть $f(x_1, \ldots, x_n), g(x_1, \ldots, x_p), h(x_1, \ldots, x_q) \in \mathrm{NS}_k^1$, $f \trianglelefteq_{\mathrm{NS}_k^1} g \trianglelefteq_{\mathrm{NS}_k^1} h$, $f \preceq h$. Покажем, что выполняются соотношения $f \preceq g \preceq h$. Пусть $N_f = \mathcal{L}(e, d_1, \ldots, d_{k-2})$, $N_g = \mathcal{L}(a, b_1, \ldots, b_{k-2})$, $N_h = \mathcal{L}(u, v_1, \ldots, v_{k-2})$ и e, d_i, a, b_i, u, $v_i \in \mathbb{N}$, $i = 1, \ldots, k-2$. Поскольку $f \trianglelefteq_{\mathrm{NS}_k^1} g$, то существуют формула Φ над NS_k^1, реализующая функцию f, и подформула Φ_1 формулы Φ, имеющая вид $g(\mathcal{B}_1, \ldots, \mathcal{B}_p)$, где $\mathcal{B}_1, \ldots, \mathcal{B}_p$ — формулы над NS_k^1. В силу утверждения 1 существует число $l \in \mathbb{N}$, такое, что среди $\mathcal{B}_1, \ldots, \mathcal{B}_p$ каждая переменная из $x_1, \ldots, x_n$ встречается l раз. Поэтому выполняются соотношения $a \geqslant le$, $b_i = ld_i$, $i = 1, \ldots, k-2$. Так как $g \trianglelefteq_{\mathrm{NS}_k^1} h$, то аналогично можно показать, что существует число $w \in \mathbb{N}$, такое, что $u \geqslant wa$, $v_i = wb_i$, $i = 1, \ldots, k-2$. Кроме того, так как $f \preceq h$, то существуют формула Ψ над $\{h\}$, реализующая функцию f, и простая подформула формулы Ψ, имеющая вид $h(x_{i_1}, \ldots, x_{i_q})$. В силу утверждения 1 существует $c \in \mathbb{N}$, такое, что каждая переменная из $x_1, \ldots, x_n$ встречается среди $x_{i_1}, \ldots, x_{i_q}$ ровно c раз. Поскольку Φ является формулой над R_k, для любого набора $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ из N_f выполняются равенства $h(\alpha_{i_1}, \ldots, \alpha_{i_q}) = f(\alpha_1, \ldots, \alpha_n) = 1$. А значит, $(\alpha_{i_1}, \ldots, \alpha_{i_q}) \in N_h$. Поэтому $u = ce$, $v_i = cd_i$, $i = 1, \ldots, k-2$. Следовательно, выполняются равенства $c = wl$ и $a = le$, $b_i = ld_i$, $u = wa$, $v_i = wb_i$, $i = 1, \ldots, k-2$. А значит,

$$f(x_1, \ldots, x_n) = g(\underbrace{x_1, \ldots, x_1}_{l}, \underbrace{x_2, \ldots, x_2}_{l}, \ldots, \underbrace{x_n, \ldots, x_n}_{l});$$

$$g(x_1, \ldots, x_p) = h(\underbrace{x_1, \ldots, x_1}_{w}, \underbrace{x_2, \ldots, x_2}_{w}, \ldots, \underbrace{x_p, \ldots, x_p}_{w}).$$

Поэтому выполняются соотношения $f \in [\{g\}]$, $g \in [\{h\}]$. Следовательно, $f \preceq g \preceq h$.

Таким образом, множество NS_k^1 обладает свойствами $(*)$ и $(**)$.

Покажем теперь, что для любых функций $f, g \in \mathrm{NS}_k^1$ из соотношения $f \not\equiv g$ следует $f \not\sim g$. Предположим, что это не так. Пусть $f, g \in \mathrm{NS}_k^1$, $f \not\equiv g$ и $f \sim g$. Пусть $N_f = \mathcal{L}(e, d_1, \ldots, d_{k-2})$, $N_g = \mathcal{L}(a, b_1, \ldots, b_{k-2})$ и e, d_i, a, $b_i \in \mathbb{N}$, $i = 1, \ldots, k-2$. Поскольку $f \sim g$, то $f \preceq g$. Поэтому существуют формула Ψ над $\{g\}$, реализующая функцию f, и простая подформула формулы Ψ, имеющая вид $g(x_{i_1}, \ldots, x_{i_p})$. В силу утверждения 1 существует число $c \in \mathbb{N}$, такое, что каждая переменная из $x_1, \ldots, x_n$ встречается среди $x_{i_1}, \ldots, x_{i_p}$ c раз. Поскольку Φ является формулой над R_k, то для каждого набора $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ из N_f, выполняются равенства $g(\alpha_{i_1}, \ldots, \alpha_{i_p}) = f(\alpha_1, \ldots, \alpha_n) = 1$, то есть $(\alpha_{i_1}, \ldots, \alpha_{i_p}) \in N_g$. Поэтому

выполняются равенства $a = ce$, $b_i = cd_i$, $i = 1, \ldots, k-2$. Аналогичным образом можно показать, что в силу соотношения $g \preceq f$ найдется такое число $t \in \mathbb{N}$, что выполняются равенства $e = ta$, $d_i = tb_i$, $i = 1, \ldots, k-2$. Следовательно, $c = t = 1$ и $a = e$, $b_i = d_i$, $i = 1, \ldots, k-2$. А значит, $f \cong g$. Получили противоречие. Поэтому из соотношения $f \ncong g$ следует $f \not\sim g$.

Перейдем теперь непосредственно к доказательству теоремы. Справедливость утверждения (1) следует из теоремы 1. А именно, поскольку множество NS_k^1 обладает свойствами $(*)$ и $(**)$, то класс F имеет базис тогда и только тогда, когда каждая функция из G содержится в некоторой ограниченной максимальной цепи множества G относительно $\preceq$.

Докажем теперь утверждение (2). Достаточность очевидна. Докажем необходимость. Пусть класс F имеет конечный базис. Тогда существует конечный базис $\mathfrak{A}$ класса F, такой, что $\mathfrak{A} \subseteq G$. Пусть $\mathfrak{A} = \{f_1(x_1, \ldots, x_{n_1}), \ldots, f_s(x_1, \ldots, x_{n_s})\}$. Пусть $f(x_1, \ldots, x_n) \in G$, Φ — формула над $\mathfrak{A}$, реализующая функцию f, Φ_1 — некоторая подформула формулы Φ, имеющая вид $f_i(\mathcal{B}_1, \ldots, \mathcal{B}_{n_i})$, где $1 \leqslant i \leqslant s$, $\mathcal{B}_1, \ldots, \mathcal{B}_m$ — формулы над $\mathfrak{A}$. В силу утверждения 1 среди $\mathcal{B}_1, \ldots, \mathcal{B}_m$ встречается каждая переменная из множества $\{x_1, \ldots, x_n\}$. Тогда выполняется неравенство $n \leqslant n_i$. Таким образом, для любой функции $f(x_1, \ldots, x_n) \in G$ выполняется неравенство $n \leqslant \max(n_i)$, где максимум берется по всем $i = 1, \ldots, s$. Следовательно, множество G конечно (так как оно состоит из попарно неконгруэнтных функций).

Таким образом, теорема доказана.

Автор выражает благодарность профессору А. Б. Угольникову за постановку задачи и постоянное внимание к работе.

Работа выполнена при финансовой поддержке РФФИ (проект 08–01–00863) и программы поддержки ведущих научных школ РФ (проект НШ–4470.2008.1), программы фундаментальных исследований ОМН РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Задачи оптимального синтеза управляющих систем»).

Список литературы

1. Post E. L. The two-valued iterative systems of mathematical logic // Annals of Math. Studies. Princeton Univ. Press. —1941. —5.

2. Янов Ю. И., Мучник А. А. О существовании k-значных замкнутых классов, не имеющих конечного базиса // ДАН СССР. — 1959. — 127, № 1. — С. 44–46.

3. Михайлович А. В. О замкнутых классах трехзначной логики, порожденных симметрическими функциями // Вестник Моск. ун-та. Сер. 1, Математика. Механика. 2008. №4. — С. 54–57.

4. Михайлович А. В. О замкнутых классах многозначной логики, порожденных функциями со специальными свойствами // Материалы XVII междунар. школы-семинара "Синтез и сложность управляющих систем" (Новосибирск, 27 октября – 1 ноября 2008 г.). — Новосибирск. Изд-во Института математики, 2008. — С. 117 – 122.

5. Яблонский С. В. Введение в дискретную математику. — М.: Высшая школа, 2001. 384 с.

РЕГУЛЯРИЗАЦИЯ НЕКОТОРЫХ ОЦЕНОК СЛОЖНОСТИ УМНОЖЕНИЯ МНОГОЧЛЕНОВ

И. С. Сергеев (Москва)

В настоящей работе собрано несколько замечаний о том, как дополнить некоторые стандартные алгоритмы умножения, чтобы зависимость оценок сложности этих алгоритмов от степени перемножаемых многочленов была более регулярной.

1. Пусть над кольцом K определены прямое и обратное дискретное преобразование Фурье (ДПФ) порядка n, допускающие эффективную по сложности реализацию (понятия сложности и ДПФ см., например, в [2] и [5], соответственно), и это число n нельзя увеличить по каким-либо причинам. При этом требуется перемножить многочлены $f, g \in K[x]$ суммарной степени, меньшей, чем mn, где $m \leqslant n/2$.

Универсальный способ решения этой проблемы восходит к Шёнхаге и Штрассену [10, 8] и заключается в следующем. При помощи замены переменной $x^{\lfloor n/2 \rfloor} = y$ многочлены f и g погружаются в кольцо расширения $(K[x]/(x^n - 1))[y]$. Суммарная степень d новых многочленов по переменной y ограничена сверху величиной $2m + O(1)$. Используя ДПФ над кольцом $K[x]/(x^n - 1)$ с выбором примитивного корня в виде x^k, умножение выполняется со сложностью $O(d \log d)$ операций сложения-вычитания и умножения на степени примитивного корня и d нетривиальным умножениям в $K[x]/(x^n - 1)$ в предположении, что суммарный порядок ДПФ, применяемых к одному многочлену, равен в точности d, и каждое из ДПФ может быть реализовано быстрым алгоритмом. Сложения-вычитания и умножения на степени x в $K[x]/(x^n - 1)$ выполняется со сложностью $O(n)$, а произвольное умножение — со сложностью $2F(n) + F^*(n) + O(n)$, где $F(n)$ и $F^*(n)$ — сложность соответственно прямого и обратного ДПФ порядка n в кольце K. Поэтому для сложности алгоритма в перечисленных

предположениях (быстрое умножение в кольце расширения, подходящий вид чисел n, m, d) можно указать оценку

$$2m(2F(n) + F^*(n)) + O(mn \log m). \tag{1}$$

Предлагаемый способ основан на использовании *многократного ДПФ*. Предположим наличие в кольце K элементов $\omega = \zeta_0, \zeta_1 \ldots, \zeta_{m-1}$, таких, что

$$\zeta_i/\zeta_j \neq \omega^k, \quad \exists \zeta_i^{-1}, \quad \exists (\zeta_i^n - \zeta_j^n)^{-1} \tag{2}$$

при любых k и $i \neq j$, где ω — примитивный корень степени n (тот самый, который используется в ДПФ порядка n).

Воспользуемся m-кратным ДПФ порядка $n \times \ldots \times n$, которое применительно к многочлену $f(x)$ состоит в вычислении значений

$$f(\zeta_i \omega^k), \quad i = 0, \ldots, m-1, \quad k = 0, \ldots, n-1.$$

Такое преобразование можно реализовать, вычислив сначала многочлены $\psi_i(x) = f(\zeta_i x) \bmod (x^n - 1)$, и применив к каждому из них ДПФ порядка n. Действительно,

$$\psi_i(\omega^k) = f(\zeta_i \omega^k).$$

Обозначим $f = \sum f_i x^i$ и $\psi_i = \sum \psi_{i,j} x^j$. Тогда набор коэффициентов многочленов ψ_i при x^k может быть найден из соотношения

$$\begin{bmatrix} \psi_{0,k}/\zeta_0^k \\ \psi_{1,k}/\zeta_1^k \\ \cdots \\ \psi_{m-1,k}/\zeta_{m-1}^k \end{bmatrix} = \begin{bmatrix} 1 & \zeta_0^n & \zeta_0^{2n} & \cdots & \zeta_0^{(m-1)n} \\ 1 & \zeta_1^n & \zeta_1^{2n} & \cdots & \zeta_1^{(m-1)n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \zeta_{m-1}^n & \zeta_{m-1}^{2n} & \cdots & \zeta_{m-1}^{(m-1)n} \end{bmatrix} \times \begin{bmatrix} f_k \\ f_{n+k} \\ f_{2n+k} \\ \cdots \\ f_{(m-1)n+k} \end{bmatrix}.$$

Умножение матрицы на вектор в правой части можно интерпретировать как вычисление значений многочлена $\sum f_{in+k} y^i$ в точках ζ_j^n для всех $j = 0, \ldots, m-1$. Это, как известно (см. [1]), можно выполнить за $O(M(m) \log m)$ операций, где $M(m)$ — верхняя оценка сложности умножения многочленов степени m, удовлетворяющая при $m_1 \leqslant m_2$ соотношению $M(m_1)/m_1 \leqslant M(m_2)/m_2$. Для нахождения $\psi_{i,k}$ после этого надо выполнить еще m умножений на элементы ζ_i^k.

Следовательно, сложность m-кратного ДПФ можно оценить как

$$mF(n) + O(nM(m) \log m).$$

Покажем, что аналогичная оценка справедлива и для сложности обратного преобразования.

Применяя обратное ДПФ порядка n к каждому из наборов

$$f(\zeta_i\omega^0), f(\zeta_i\omega^1), \ldots, f(\zeta_i\omega^{n-1}),$$

найдем многочлены $\psi_i(x)$, $i = 0, \ldots, m-1$. Замена переменной $x \to x/\zeta_i$ превращает многочлен $\psi_i(x)$ в многочлен $\phi_i(x) = f(x) \bmod (x^n - \zeta_i^n)$. Эти вычисления реализуются со сложностью $O(mn)$. Остается по набору остатков

$$\phi_i(x) = f(x) \bmod (x^n - \zeta_i^n), \quad i = 0, \ldots, m-1,$$

восстановить многочлен $f(x)$.

Для этого, обозначив через $\phi_{i,k}$ коэффициент многочлена ϕ_i при x^k, удобно воспользоваться соотношением

$$\begin{bmatrix} \phi_{0,k} \\ \phi_{1,k} \\ \cdots \\ \phi_{m-1,k} \end{bmatrix} = \begin{bmatrix} 1 & \zeta_0^n & \zeta_0^{2n} & \cdots & \zeta_0^{(m-1)n} \\ 1 & \zeta_1^n & \zeta_1^{2n} & \cdots & \zeta_1^{(m-1)n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \zeta_{m-1}^n & \zeta_{m-1}^{2n} & \cdots & \zeta_{m-1}^{(m-1)n} \end{bmatrix} \times \begin{bmatrix} f_k \\ f_{n+k} \\ f_{2n+k} \\ \cdots \\ f_{(m-1)n+k} \end{bmatrix}.$$

Неизвестный вектор (справа) может быть найден путем интерполяции многочлена $\sum f_{in+k}y^i$ по известным значениям $\phi_{i,k}$ в точках ζ_i^n. Это стандартным способом (см. также [1]) реализуется со сложностью $O(M(m)\log m)$. Получаем оценку

$$mF^*(n) + O(nM(m)\log m).$$

Теперь легко оценить сложность рассматриваемого умножения многочленов над K как сложность трех m-кратных ДПФ плюс mn умножений в кольце. Справедлива

Теорема 1. *Пусть в кольце K определено m-кратное ДПФ порядка $n \times \ldots \times n$ (т.е. выполнены условия (2)). Тогда сложность умножения многочленов суммарной степени, меньшей mn, над K не превосходит*

$$m(2F(n) + F^*(n)) + O(nM(m)\log m). \tag{3}$$

Сравнивая полученную оценку с (1), видим, что при $M(m) = O(m\log m)$, $F(n), F^*(n) = \Omega(n\log n)$ и небольших m, например, $m = 2^{o(\sqrt{\log n})}$, она лучше асимптотически в 2 раза. Остаточный член в (3) может быть понижен при специальном выборе элементов ζ_i.

2. Пусть в нашем распоряжении имеются схемы для умножения многочленов по модулям $x^{2^k} + 1$ с коэффициентами над кольцом, в котором

обратим элемент 2. Сложность и глубину таких схем обозначим через $\mu(k)$ и $d(k)$ (понятие глубины см. в [2]). Будем считать, что $\mu(k) \geqslant \mu(k-1)$ и $d(k) > d(k-1)$ для любого k.

Очевидно, что $\mu(k)$ служит верхней оценкой для сложности умножения многочленов суммарной степени не выше $2^k - 1$. Сложность умножения многочленов суммарной степени не выше $n-1$, где $2^k < n \leqslant 2^{k+1}$, стандартным образом оценивается как $\mu(k+1)$. В некоторых случаях удается получить оценку, равномерно растущую с ростом n, см. [6, 9].

На самом деле можно указать следующую регулярную оценку:

Теорема 2. *Пусть* $n = 2^{n_1} + 2^{n_2} + \ldots + 2^{n_s}$, *где* $n_1 > n_2 > \ldots > n_s$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$ *можно построить схему сложности не более*

$$\mu(n_1) + \mu(n_2) + \ldots + \mu(n_s) + 5,5n$$

и глубины не более

$$d(n_1) + n_1 - n_s + 3s - 2.$$

Для доказательства теоремы используются схемы умножения по модулям $x^{2^{n_i}} + 1$, $i = 1, \ldots, s$, схема приведения исходных многочленов по этим модулям и схема восстановления коэффициентов многочлена-произведения по остаткам от деления на многочлены $x^{2^{n_i}} + 1$ (она вычисляет решение СЛУ с матрицей, изображенной на рис. 1 (см. стр. 30): через D_a обозначены матрицы с элементами a на главной диагонали и нулями на остальных позициях). Несложно проверить, что последние две схемы имеют линейную сложность.

Следствие 1. *Пусть* $\mu(k) = f(2^k)$, *где для любых* $x, y \geqslant 1$ *справедливо* $f(x+y) \geqslant f(x) + f(y)$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$ *можно построить схему сложности не более* $f(n) + 5,5n$ *и глубины не более* $d(\lfloor \log_2 n \rfloor) + 4\lfloor \log_2 n \rfloor + 1$.

То, что вычисление произведения по нескольким модулям позволяет избежать удвоения мультипликативной постоянной в оценке сложности при переходе от $n = 2^k$ к произвольному числу — факт, в какой-то степени известный. Так, в работе [4] рассмотрен способ использования двух модулей, т.е. $s = 2$. В работе [3] указано, что выгодно выполнять вычисления с несколькими модулями. Аналогичная изложенной схема вычислений возникает в алгоритме умножения при помощи «усеченного» ДПФ, см. [6, 7]. В настоящей работе показано, что регуляризация оценки сложности возможна при любом алгоритме умножения.

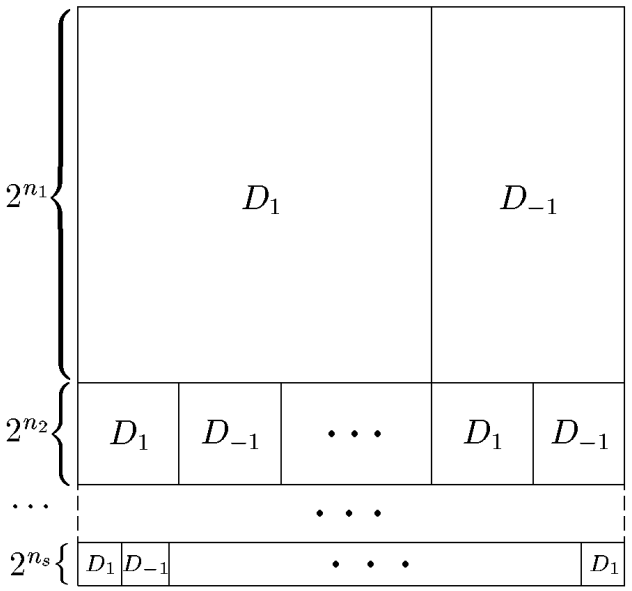

Рис. 1

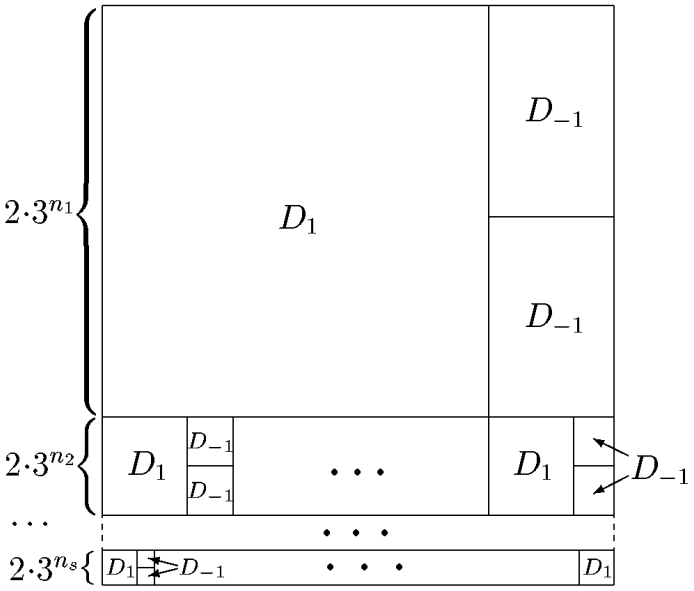

Рис. 2

Выбирая ближайшее сверху к n число n', кратное $2^{n_1-\alpha(n)}$, где $\alpha(n)$ — медленно растущая функция, и переходя к схеме умножения многочленов суммарной степени не выше $n'-1$, получаем

Следствие 2. *В условиях следствия* 1 *дополнительно предположим, что* $f(x)/x \to \infty$ *при* $x \to \infty$ *и* $f(x) = x^{O(1)}$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$ *можно построить схему сложности не более* $(1+o(1))f(n)$ *и глубины не более* $d(\lfloor \log_2(n+o(n)) \rfloor)+o(\log n)$.

3. Рассмотрим аналогичную задачу с использованием схем для умножения многочленов по модулям $x^{2\cdot 3^k}+x^{3^k}+1$ с коэффициентами над кольцом, в котором обратим элемент 3. Сложность и глубину таких схем обозначим через $\mu'(k)$ и $d'(k)$. Будем считать, что $\mu'(k) \geqslant \mu'(k-1)$ и $d'(k) > d'(k-1)+1$ для любого k.

Теорема 3. *Пусть* $n = 2\,(3^{n_1} + 3^{n_2} + \ldots + 3^{n_s})$, $n_1 > n_2 > \ldots > n_s$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$ *можно построить схему сложности не более*

$$\mu'(n_1) + \mu'(n_2) + \ldots + \mu'(n_s) + 5(n-1)$$

и глубины не более

$$d'(n_1) + 2(n_1 - n_s) + 4s - 4.$$

В случае кольца характеристики 2 *для сложности и глубины схемы справедливы оценки:*

$$\mu'(n_1) + \mu'(n_2) + \ldots + \mu'(n_s) + 4(n-1), \qquad d'(n_1) + 2(n_1 - n_s) + 3s - 2.$$

Доказательство полностью аналогично доказательству теоремы 2. Матрица возникающей в этом случае СЛУ изображена на рис. 2.

Следствие 3. *Пусть* $\mu'(k) = f(2 \cdot 3^k)$, *где для любых* $x, y \geqslant 1$ *справедливо* $f(x+y) \geqslant f(x)+f(y)$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$, *где* $n \in [2 \cdot 3^k, 3^{k+1})$, *можно построить схему сложности не более* $f(n) + 5n$ *и глубины не более* $d'(k) + 6k$, *а в случае кольца характеристики* 2 *— схему сложности* $f(n) + 4n$ *и глубины* $d'(k) + 5k + 1$.

В случае произвольного n оценку сложности вида $(1 + o(1))f(n)$ получить пока не удается. Однако, можно доказать оценку $(4/3 + o(1))f(n)$. Справедливо

Следствие 4. *Пусть в условиях следствия* 3 *дополнительно выполняется* $f(x)/x \to \infty$ *при* $x \to \infty$ *и* $f(x) = x^{O(1)}$. *Тогда для умножения многочленов суммарной степени не выше* $n-1$ *можно построить схему сложности не более*

$$\begin{cases} (1 + o(1))f(n), & 2 \cdot 3^k \leqslant n < 3^{k+1}, \\ \left(2 - \frac{3^k}{n} + o(1)\right) f(n), & 3^k \leqslant n < \frac{3^{k+1}}{2}, \\ \left(\frac{2 \cdot 3^k}{n} + o(1)\right) f(n), & \frac{3^{k+1}}{2} < n < 2 \cdot 3^k \end{cases}$$

и глубины не более $d'(\lfloor \log_3(2n + o(n)) \rfloor - 1) + o(\log n)$.

В первом случае конструкция та же, что и в следствии 2. В третьем случае используется схема умножения многочленов суммарной степени не более $2 \cdot 3^k - 1$. Во втором случае произведение вычисляется по всем модулям $x^{2 \cdot 3^i} + x^{3^i} + 1$, где $i = k - s, \ldots, k - 1$. Отдельно вычисляются l младших и u старших коэффициентов произведения, доопределяющих систему уравнений. Для их вычисления достаточно перемножить многочлены, образованные младшими коэффициентами исходных сомножителей (суммарная степень этих многочленов не превосходит $2l - 2$), и соответственно многочлены, образованные старшими коэффициентами (суммарной степени не выше $2u - 2$). Числа $2l$ и $2u$ при этом можно выбрать из отрезков $[2 \cdot 3^i, 3^{i+1})$.

Автор выражает благодарность научному руководителю С. Б. Гашкову за внимание к работе.

Работа выполнена при финансовой поддержке РФФИ, проекты 08–01–00863 и 08–01–00632–а, программы «Ведущие научные школы», проект НШ–4470.2008.1, и программы фундаментальных исследований Отделения

математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Синтез и сложность управляющих систем»).

Список литературы

1. Ахо А., Хопкрофт Дж., Ульман Дж. Проектирование и анализ вычислительных алгоритмов. — М.: Мир, 1979.

2. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд. МГУ, 1984.

3. Bernstein D. J. Fast multiplication and its applications // Algorithmic Number Theory, MSRI Publ. — 2008. — V. 44. — P. 325–384.

4. Crandall R., Fagin B. Discrete weighted transforms and large-integer arithmetic // Math. of Comput. — 1994. — V. 62. — P. 305–324.

5. von zur Gathen J., Gerhard J. Modern computer algebra. — Cambridge University Press, 1999.

6. van der Hoeven J. Notes on the truncated Fourier transform. Tech. report. — Univ. Paris-Sud, Orsay, France, 2005.

7. Mateer T. Fast Fourier algorithms with applications. Ph. D. Thesis. — Clemson University, 2008.

8. Schönhage A. Schnelle multiplikation von polynomen über körpern der charakteristik 2 // Acta Inf. — 1977. — V. 7. — P. 395–398.

9. Schönhage A. Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients // Proc. EuroCAM. LNCS. — 1982. — V. 144. — P. 3–15.

10. Schönhage A., Strassen V. Schnelle multiplikation großer zahlen // Computing. — 1971. — V. 7. — P. 271–282. (Русский перевод: Шёнхаге А., Штрассен В. Быстрое умножение больших чисел // Кибернетический сборник. Вып. 10. — М.: Мир, 1973. — С. 87–98).

ОБ ИСПОЛЬЗОВАНИИ ЛОГИЧЕСКИХ МЕТОДОВ ПРИ УСТРАНЕНИИ ПРОТИВОРЕЧИЙ В БАЗАХ ДАННЫХ

Е. Е. Трифонова (Москва)

Для сложных распределенных баз данных актуальны вопросы целостности и устранения противоречий в данных. Противоречивость является общим феноменом в мире баз данных сегодня. Даже посредством ограничений целостности, успешно передающих семантику данных, действительные

данные в базе данных часто не удовлетворяют подобным ограничениям. Это происходит из-за того, что данные записываются из множества независимых источников (как в объединении данных) или возникают в результате комплексных длительных деятельностей, подобных технологическому процессу.

Традиционный путь избавления от противоречивости данных — это не позволять базе данных становиться противоречивой путём отмены обновлений или транзакций, которые приводят к нарушению целостности. Однако в настоящий момент этот сценарий действий становится всё более непрактичным. Поскольку, во-первых, если нарушение встречается из-за того, что данные поступают из множественных независимых источников, сливающихся в одно, то не существует отдельного обновления, ответственного за нарушение. Во-вторых, данные могут иметь противоречие в результате выполнения некоторой комплексной активности, так что не представляется возможным отследить специфическое действие, приведшее к возникновению противоречия.

В настоящий момент в большинстве случаев для работы с противоречиями в базах данных принят так называемый технично-человеческий подход. Этот подход основан на том, что на часть хранящихся данных накладываются средствами СУБД некоторые ограничения (тип данных, логические выражения, допустимый диапазон значений), выполнение этих ограничений проверяется автоматически средствами системы, а принятие решений об обработках ошибок предоставляется человеку.

Существуют три основные варианта работы с ошибочными данными: удаление, промежуточное хранение и преобразование для устранения ошибки. Поскольку при этом решение о выборе того или иного варианта оставляется человеку, то следует отметить, что назрела необходимость разработки автоматических процедур, которые согласно введённым критериям могли бы обеспечить механизм добавления непротиворечивых данных или устранения противоречий.

Для разработки такого подхода описание предметной области и правил для данных нужно рассматривать как аксиоматику над определенной сигнатурой. Если таблицы баз данных рассматривать как предикаты, а правила для данных как формулы первого порядка, то непротиворечивое состояние базы данных (восстановление) представляет собой модель для заданных аксиом (в качестве аксиом подразумеваем множество правил).

Ограничения целостности можно условно разделить на три базовых класса, которые представляют собой замкнутые формулы первого порядка, составленные из символов отношений — $P, P_1, \ldots, P_m$, кортежей переменных и констант $\bar{x}_1, ..., \bar{x}_m$ — и конъюнкций атомарных формул, касающихся

встроенных предикатов, обозначаемых через φ.

1. Отрицающие ограничения:

$$\forall \bar{x}_1, ..., \bar{x}_m \neg \left[P_1(\bar{x}_1) \wedge ... \wedge P_m(\bar{x}_m) \wedge \varphi(\bar{x}_1, ..., \bar{x}_m) \right].$$

Если $m \leqslant 2$, то ограничение является бинарным.

2. Функциональные зависимости (FDs):

$$\forall \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \neg \left[P(\bar{x}_1, \bar{x}_2, \bar{x}_4) \wedge P(\bar{x}_1, \bar{x}_3, \bar{x}_5) \wedge \bar{x}_2 \neq \bar{x}_3 \right],$$

где $\bar{x}_i$ — последовательности отдельных переменных. Более общая формулировка функциональных зависимостей это $X \to Y$, где X — множество атрибутов P, соответствующих $\bar{x}_1$ и Y — множество атрибутов, соответствующих $\bar{x}_2$ и $\bar{x}_3$. По сути это некоторое подмножество отрицающих ограничений. Однако они очень часто используются, поэтому целесообразно вынести их в отдельный класс.

3. Зависимости включения (INDs):

$$\forall \bar{x}_1 \exists \bar{x}_3 \left[Q(\bar{x}_1) \subseteq P(\bar{x}_2, \bar{x}_3) \right],$$

где $\bar{x}_i$ — последовательности отдельных переменных с $\bar{x}_2$, содержащимся в $\bar{x}_1$ и P, Q — отношения базы данных. Это часто записывается как $Q[Y] \subseteq P[X]$, где X (соответственно Y) есть множество атрибутов $P(Q)$, соответствующих $\bar{x}_2$. Если P и Q ясны из контекста, то их опускают и просто записывают $Y \subseteq X$. Полные зависимости включения выражаются без кванторов существования.

Будем рассматривать базу данных D, в которую в течение промежутка времени T_0 вносят большой объём данных D_1. По истечении времени T_0 получившийся объём данных $D + D_1$ необходимо проверить на противоречия с устранением обнаруженных противоречий. При этом отдельные кортежи данных мы не можем изменять, а только удалять для устранения противоречия. То есть если мы считает какой-то элемент наших данных ложным, то мы не редактируем его, а удаляем.

Таким образом, при автоматизации процесса устранения противоречий необходимо выбирать, какие именно означивания (кортежи) необходимо оставить, а какие удалить. Автоматизация невозможна без использования логических методов. Основная идея автоматического построения моделей состоит в том, что сначала мы строим ядро модели или минимальную модель. Выделим для противоречивого состояния БД те ее подмножества, для которых одно из ограничений целостности истинно (при этом не будем рассматривать те кортежи, которые участвуют в противоречии, то есть если два кортежа противоречат друг другу, то оба выкидываем) и определим

их пересечение. Данные из пересечения будем считать ядром восстановления БД, которые должны быть включены в любое восстановление БД. Это ядро представляет собой минимальную модель — основу восстановления непротиворечивой БД. Затем для получения восстановления (модели, которая будет включать в себя больше информации) необходимо произвести добавления данных. При этом для добавляемых данных будем требовать выполнение следующих свойств.

1. Ни один добавленный кортеж не нарушает свойство минимальной модели быть моделью.

2. При последовательном добавлении по одному кортежу для каждого из противоречивых ранее ограничений целостности получаем непротиворечивую конструкцию.

3. Добавление по одному кортежу (группе кортежей) для каждого из противоречивых ранее ограничений является аналогичным использованию гиперграфа противоречий.

Гиперграф противоречий — это графический способ представления нарушений целостности. Вершины гиперграфа — это кортежи. Рёбрами в таком гиперграфе соединяются те кортежи, которые между собой противоречат. Причём на каждое нарушение — по одному ребру, то есть если четыре кортежа нарушают одно ограничение целостности, то они будут соединены одним ребром. Тогда восстановление базы данных — это максимальное независимое подмножество вершин графа, то есть внутренне устойчивое множество.

Состояние изначально противоречивой базы данных $D + D_1$ с удалёнными противоречиями называем восстановлением и обозначаем как D_0. То есть, D_0 получается из $D + D_1$ путём удаления находимых противоречий. Очевидно, что для каждого противоречивого состояния базы данных $D + D_1$ существует множество состояний D_0, каждое из которых будет являться восстановлением. То есть, каждое из которых будет являться моделью для исходного множества правил.

На самом деле нам нужна не минимальная модель, а какая-то модель, обладающая определенными свойствами. Исходя из каких параметров можно оценивать модель, и исходя из чего выбирать то или иное восстановление в качестве текущего состояния базы данных? Основными будут являться следующие критерии:

1. Минимальные потери информации (удалять минимальное количество кортежей для того, чтобы информации было потеряно как можно меньше).

2. Весовая функция кортежей (из практических соображений и предметной области)

Остановимся на каждом из них подробнее. Если нам необходимо, чтобы потери информации были минимальны, то это означает, что мы хотим минимальное количество кортежей удалить. Для исправления некоторых противоречий бывают возможны несколько вариантов удалений: возможно исправить, удалив 1 кортеж, а возможен вариант с удалением сразу двух. Тогда в данном случае мы всегда будем выбирать первый вариант, то есть этот подход особенно хорош, когда у нас нет приоритета одних данных перед другими, а кортежи (означивания) равнозначны.

Весовую функцию для каждого из кортежей мы можем получить, учитывая следующие параметры:

1. *Востребованность кортежа* (выдаётся ли кортеж, который мы хотим удалить, пользователю и как часто).

2. *Время появления кортежа* (был ли кортеж в «последнем принятом восстановлении» или нет).

3. *Приоритет хранимой информации* (есть таблицы из которых данные можно удалять только в крайнем случае

4. *Приоритет отдельных кортежей* (в рамках одной таблицы).

5. *Взвешенный параметр обращений к таблице* (чем чаще обращаются к данных, тем более достоверная информация).

6. *Достоверность источника данных* (можем прослеживать насколько часть отвергаются данные из конкретного источника).

7. *Количество источников информации* (некоторые кортежи могут быть составными и это, соответственно, может понижать надёжность рассматриваемой информации).

При этом в зависимости от того, какой из параметров будет более или менее приоритетным, возможно изменять величину весовой функции. Величина весовой функции в тот или иной момент времени будет различна.

Возможен и другой подход. Ограничения, накладываемые на БД, суть формулы исчисления предикатов первого порядка. Но интерпретация предикатов в данном случае несколько иная, нежели в классическом исчислении, когда при означивании своих аргументов предикат принимает только

два значения: истина/ложь. В данном случае означивание каждого предиката приводит к тому, что предикат принимает бесконечное множество значений из интервала [0, 1].

При этом 0 (ложь) интерпретируется как недостоверное (невероятное) событие, данные абсолютно ложны, невыполнимая формула. Значение между 0 и 1 характеризует вероятность того, что событие случится, вероятность того, что формула будет истинной. Тем самым промежуточное между 0 и 1 значение свидетельствует о том, что наши представления о предметной области не полные. Поэтому в ряде случаев при одном означивании предикат может быть истинным, а в ряде — ложным. И это определяется тем контекстом (здесь формулой), в которой данный предикат встречается. Значение же 1 (истина) — достоверное событие, достоверные данные, тождественно истинная формула.

Как видно, промежуточные значения могут быть интерпретированы как отнесение конкретного означивания предиката к истине или лжи с некоторой достоверностью. На эту достоверность влияет сложность предиката (его количество атрибутов), контекст (формульное окружение) и другие факторы (например то, как часто редактируются данные для этого предиката). Неопределенность возникает из способа описания объектов реального мира.

Таким образом, при автоматизации процесса построения модели (восстановления базы данных) возникают различные варианты для оценки того, какое означивание необходимо оставить. Можно вводить оценочную функцию «ценности» того или иного означивания исходя из источника данных или иных вспомогательных сведений, можно говорить о силе связи данного означивания с остальными, в том смысле, что удаление некоторого количества информации повлечёт за собой удаление данного означивания. Наконец, возможно введение информационной функции, позволяющей «взвешивать» каждое означивание. В итоге приходим к необходимости сочетания логических методов с использованием весовых целочисленных функций. Либо же необходимо рассматривать подход, затрагивающий истинностные значения не как две величины, а как бесконечное множество значений из интервала. Таким образом, представляется перспективным разработать математический аппарат и эффективные алгоритмы для построения восстановления с использованием различных критериев на автоматической основе.

На настоящий момент рядом авторов получено, что задача получения восстановления для произвольных отрицательные ограничений с минимальным количеством удаляемых кортежей является PTIME (т.е. может быть решена за полиноминальное время с помощью детерминированной

машины Тьюринга).

Список литературы

1. Marcinkowski Jerzy, Chomicki Jan. Minimal-change integrity maintenance using tuple deletions. // Information and Computation. — 2005. — Т. 1–2, вып. 197. — С. 90–121.

О СИНТЕЗЕ ИГРОВЫХ ПРОГРАММ С ПОМОЩЬЮ ЛОГИКИ ВЫСКАЗЫВАНИЙ

Р. В. Хелемендик (Москва)

1. Введение

Игровая программа (ИП) представляет собой специальный граф, который описывает выигрышную стратегию при взаимодействии двух сторон. Рассматривается задача синтеза ИП для заданных условий: начальных значений переменных, набора функций («ходов»), типа взаимодействия и цели, записываемой формулой логики ветвящегося времени (лвв). При этом стратегия, описываемая ИП (в случае её существования), считается выигрышной, если для этой ИП выполнены все компоненты зафиксированного условия.

В работе [3] дано определение ИП и установлена эквивалентность выразительных возможностей языков ИП и лвв. В то же время язык ИП предоставляет более широкие средства для структуризации записи задачи в рамках следующего подхода: указание начальной ситуации; определение правил её изменения и типа взаимодействия; постановка цели. Таким образом, многочисленные и сложные фрагменты формул лвв, кодирующие ходы сторон, и особенности их взаимодействия, могут быть достаточно просто записаны векторными k-значными функциями.

Однако при создании алгоритмов решения задачи синтеза ИП возникает некоторое множество «избыточных» ИП, что приводит к неэффективным решениям и затрудняет доказательство полноты алгоритмов. В связи с этим в настоящей работе уточнено определение ИП с исключением избыточных выигрышных стратегий и сохранением эквивалентности классов задач, решаемых путём распознавания выполнимости формул лвв и путём синтеза ИП. Особенностью данного определения ИП является достаточно жёсткая «привязка» цели и её подформул к вершинам ИП, что с одной стороны позволяет минимизировать число разбираемых формул на каждом

шаге, а с другой — на ранней стадии отсекать тупиковые попытки построения ИП. Для введённого определения ИП в настоящей работе построено сведéние задачи синтеза ИП к распознаванию выполнимости формул лв (логики высказываний), что позволяет решать актуальные теоретические и прикладные задачи в хорошо изученном языке алгебры логики. При построении сведéния некоторые фрагменты формул лв представляют собой модификацию формул ограниченной глубины из работы [1].

2. Игровая программа

Определение. Обозначим через $Y = \{y_1, \ldots, y_n\}$, $n \geqslant 4$, конечное множество переменных, $\overline{y} = \langle y_1, \ldots, y_n \rangle$ — набор переменных $y_1, \ldots, y_n$; $\mathrm{A} = \{0, \ldots, \mathrm{k}-1\} \cup \{2, 3\}$, $k \geqslant 2$, — конечную область значений переменных из множества Y, $\overline{\alpha}^{\delta} = \langle \alpha_1^{\delta}, \ldots, \alpha_n^{\delta} \rangle$ — набор значений этих переменных, $0 \leqslant \alpha_1^{\delta} \leqslant 2$, $0 \leqslant \alpha_j^{\delta} \leqslant 3$, $2 \leqslant j \leqslant 3$, $0 \leqslant \alpha_j^{\delta} \leqslant k-1$, $4 \leqslant j \leqslant n$; W, B — конечные множества частичных n-мерных функций называемых, соответственно, *множествами ходов белых и чёрных*. Через F_{δ}^{w} (F_{δ}^{b}) обозначим множество n-мерных функций f_t^{w} (f_t^{b}) из W (B), определённых на наборе $\overline{\alpha}^{\delta}$. Выделенный набор $\overline{\alpha}^{\delta(0)}$ назовем *начальным*. Пятёрку $R = \langle Y, \mathrm{A}, \overline{\alpha}^{\delta(0)}, \mathrm{W}, \mathrm{B} \rangle$ будем называть *игровыми правилами* или *R-правилами*. Переменная y_1 в данном определении управляет очерёдностью ходов белых и чёрных, а переменные y_2 и y_3 — типами взаимодействия, которые называются типами «доверия», «просчитывания» и «максимального выбора» (см. [3]). Пара $\mathcal{U} = \langle R, \Theta \rangle$ называется условием для задачи синтеза ИП. В этом условии Θ называется *целью* и является формулой логики ветвящегося времени (см. [2]), в которой пропозициональными переменными являются утверждения вида $(y_j = l)$, где $y_j \in Y$, $1 \leqslant j \leqslant n$, $0 \leqslant l \leqslant (k-1)$.

Обозначим через $E_r^{\exists}$ ($E_r^{\forall}$) конечные множества формул вида $\exists \bigcirc \varphi$ (соответственно, $\forall \bigcirc \varphi$); пусть $D(\theta_r)$ — конечное непустое множество пар $\langle E_r^{\exists}, E_r^{\forall} \rangle$, получаемое по формуле θ_r; $K^1(E_r^{\exists}, E_r^{\forall})$, $K^2(E_r^{\exists}, E_r^{\forall})$, $K^3(E_r^{\exists}, E_r^{\forall})$ — соответственно, формула θ_r^1, и конечные множества формул θ_r^2 и θ_r^3, получаемые по $E_r^{\exists}$ и $E_r^{\forall}$.

Определим ИП $\mathcal{P}_{\mathcal{U}}$, удовлетворяющую условию $\mathcal{U}$.

Базис. Начальная вершина этой ИП есть $v^{\overline{\alpha}^{\delta(0)}, x_0}_{0, \theta_0, E_0^{\exists}, E_0^{\forall}}$, где θ_0 есть Θ, $\langle E_0^{\exists}, E_0^{\forall} \rangle \in D(\theta_0)$, а x_0 есть либо $\otimes$, либо определено по $\alpha_1^{\delta(0)}$, $\alpha_2^{\delta(0)}$, $\alpha_3^{\delta(0)}$ и таблице 1 из работы [3], причём $x_0 = \otimes \rightleftharpoons E_0^{\exists} = \emptyset$.

Индукционный переход. Пусть в ИП $\mathcal{P}_{\mathcal{U}}$ определена вершина $v^{\overline{\alpha}^{\delta(r)}, x_r}_{r, \theta_r, E_r^{\exists}, E_r^{\forall}}$. Тогда если $x_r = \otimes$, то из данной вершины не выходит дуг. Иначе возможны следующие случаи:

1) $x_r = f^{\mathrm{w}}_t$, где $f^{\mathrm{w}}_t \in F^{\mathrm{w}}_{\delta(r)} \subseteq \mathrm{W}$. Тогда вершина $v^{\overline{\alpha}^{\delta(r)},x_r}_{r,\theta_r,E^{\exists}_r,E^{\forall}_r}$ называется белым *преобразователем*. Из этого преобразователя выходит единственная дуга, помеченная символом f^{w}_t, и эта дуга входит в некоторую вершину $v^{\overline{\alpha}^{\delta(t(\delta(r)))},x_{t(\delta(r))}}_{t(\delta(r)),\theta_{t(\delta(r))},E^{\exists}_{t(\delta(r))},E^{\forall}_{t(\delta(r))}}$ со значением $\overline{\alpha}^{\delta(t(\delta(r)))} = f^{\mathrm{w}}_t(\overline{\alpha}^{\delta(r)})$, $\theta_{t(\delta(r))} = K^1(E^{\exists}_r, E^{\forall}_r)$, $\langle E^{\exists}_{t(\delta(r))}, E^{\forall}_{t(\delta(r))}\rangle \in D(\theta_{t(\delta(r))})$, а $x_{t(\delta(r))}$ определено по $\overline{\alpha}^{\delta(t(\delta(r)))}$ аналогично x_0. Отметим, что в случае $|F^{\mathrm{w}}_{\delta(r)}| > 1$ в качестве функции f^{w}_t допускается любой элемент $F^{\mathrm{w}}_{\delta(r)}$.

2) $x_r = F^{\mathrm{w}}_{\delta(r)}$, и $|F^{\mathrm{w}}_{\delta(r)}| \geqslant 1$. Тогда вершина $v^{\overline{\alpha}^{\delta(r)},x_r}_{r,\theta_r,E^{\exists}_r,E^{\forall}_r}$ называется белым *ветвителем*. Из этой вершины выходит $|F^{\mathrm{w}}_{\delta(r)}|$ дуг, помеченных соответственно символами f^{w}_t, где $f^{\mathrm{w}}_t \in F^{\mathrm{w}}_{\delta(r)} \subseteq \mathrm{W}$, и эти дуги входят соответственно в вершины $v^{\overline{\alpha}^{\delta(t(\delta(r)))},x_{t(\delta(r))}}_{t(\delta(r)),\theta_{t(\delta(r))},E^{\exists}_{t(\delta(r))},E^{\forall}_{t(\delta(r))}}$ со значениями $\overline{\alpha}^{\delta(t(\delta(r)))} =$ $= f^{\mathrm{w}}_t(\overline{\alpha}^{\delta(r)})$, при этом $\theta_{t(\delta(r))} \in K^2(E^{\exists}_r, E^{\forall}_r)$, $\langle E^{\exists}_{t(\delta(r))}, E^{\forall}_{t(\delta(r))}\rangle \in D(\theta_{t(\delta(r))})$, а $x_{t(\delta(r))}$ определено по $\overline{\alpha}^{\delta(t(\delta(r)))}$ аналогично x_0.

3) $x_r = F^{\mathrm{w}}_{\tau(\delta(r))}$, и $F^{\mathrm{w}}_{\tau(\delta(r))} \subseteq F^{\mathrm{w}}_{\delta(r)}$, $|F^{\mathrm{w}}_{\tau(\delta(r))}| \geqslant 1$. Тогда вершина $v^{\overline{\alpha}^{\delta(r)},x_r}_{r,\theta_r,E^{\exists}_r,E^{\forall}_r}$. называется белым *выбирателем*. Из этой вершины выходит $|F^{\mathrm{w}}_{\tau(\delta(r))}|$ видов дуг, которые могут быть продублированы конечное число раз, помеченных соответственно символами f^{w}_t, где $f^{\mathrm{w}}_t \in F^{\mathrm{w}}_{\tau(\delta(r))} \subseteq \mathrm{W}$, и эти дуги входят в вершины $v^{\overline{\alpha}^{\delta(t(\delta(r),g))},x_{t(\delta(r))}}_{t(\delta(r),g),\theta_{t(\delta(r),g)},E^{\exists}_{t(\delta(r),g)},E^{\forall}_{t(\delta(r),g)}}$, соответственно, со значениями $\overline{\alpha}^{\delta(t(\delta(r)))} = f^{\mathrm{w}}_t(\overline{\alpha}^{\delta(r)})$, при этом g зависит от $t(\delta(r))$, $F^{\mathrm{w}}_{\tau(\delta(r))}$, $F^{\mathrm{w}}_{\delta(r)}$ и $E^{\exists}_r$; $\theta_{t(\delta(r),g)} \in K^3(E^{\exists}_r, E^{\forall}_r)$, $\langle E^{\exists}_{t(\delta(r),g)}, E^{\forall}_{t(\delta(r),g)}\rangle \in D(\theta_{t(\delta(r),g)})$, а $x_{t(\delta(r),g)}$ определено по $\overline{\alpha}^{\delta(t(\delta(r)))}$ аналогично x_0.

4)–6) $x_r \in \{f^{\mathrm{b}}_t, F^{\mathrm{b}}_{\delta(r)}, F^{\mathrm{b}}_{\tau(\delta(r))}\}$. Эти случаи определяются аналогично случаям 1)–3)

Множество $D(\theta_r)$ в данном определении получается путём преобразования формулы θ_r к аналогу д.н.ф., в которой элементами конъюнкций являются константа $\top$, пропозициональные переменные и их отрицания; а также формулы вида $\exists \bigcirc \varphi$, $\forall \bigcirc \varphi$, множество которых образуют пары $\langle E^{\exists}_r, E^{\forall}_r\rangle$. Формула $\theta_{t(\delta(r))}$, определяемая преобразованием $K^1(E^{\exists}_r, E^{\forall}_r\rangle)$, является конъюнкцией всех формул из множеств $E^{\exists}_r$ и $E^{\forall}_r$, в которых удалены главные связки $\exists\bigcirc$ и $\forall\bigcirc$. Формула $\theta_{t(\delta(r))} = \theta^2_{t(\delta(r))}$ является одной из комбинаций, получаемых путем размещения в одну из $|F^{\mathrm{w}}_{\delta(r)}|$ ячеек некоторого подмножества формул из $E^{\exists}_r$ и всех формул из множества $E^{\forall}_r$, из которых также удалены главные связки $\exists\bigcirc$ и $\forall\bigcirc$. Формула $\theta_{t(\delta(r),g)}$, являющаяся одним из элементов множества $K^3(E^{\exists}_r, E^{\forall}_r\rangle$, получается аналогично форму-

ле $\theta^2_{t(\delta(r))}$ с заменой $|F^{\mathrm{w}}_{\delta(r)}|$ на $|F^{\mathrm{w}}_{\tau(\delta(r))}|$, а также возможностью дублирования вершин с одинаковыми значениями $\overline{\alpha}^{\delta(t(\delta(r)))}$, но разными формулами $\theta^3_{t(\delta(r),g)}$, зависящими от подмножеств множества $E^{\exists}_r$.

Дополнительным требованием для $\mathcal{P}_{\mathcal{U}}$ является истинность в каждой её вершине $v^{\overline{\alpha}^{\delta(r)},x_r}_{r,\theta_r,E^{\exists}_r,E^{\forall}_r}$ формулы θ_r, $(\mathcal{P}_{\mathcal{U}}, v^{\overline{\alpha}^{\delta(r)},x_r}_{r,\theta_r,E^{\exists}_r,E^{\forall}_r} \models \theta_r)$, т.е., в частности, истинность цели Θ в ИП $\mathcal{P}_{\mathcal{U}}$, которая определяется аналогично истинности формулы логики ветвящегося времени в модели (см. [2]).

Пусть $\mathcal{U} = \langle R, \Theta\rangle$. Существует ли ИП $\mathcal{P}_{\mathcal{U}}$, удовлетворяющая условию $\mathcal{U}$? Если существует, то необходимо построить хотя бы одну такую ИП.

3. Алгоритм сведéния

Построение сведéния задачи синтеза ИП к распознаванию выполнимости формул лв вообще говоря состоит из трёх этапов и использует определения и методы работы [2]. На первом этапе происходит построение по условию $\mathcal{U}$ формулы лв Θ^{pq}_0, которое «моделирует» построение схемы модели из указанной работы.

Пусть $m = \lceil log_2 k \rceil$. Введём $m(n-3)+6$ пропозициональных переменных для кодирования равенств $(y_j = l)$. Тогда каждой такой переменной p_h сопоставим множество соответствующих ей пропозициональных переменных $\mathrm{p}_{h,i}$, мощность которого не превосходит числа вершин схемы модели. Аналогичным образом формулам вида $\forall \bigcirc \varphi_j$ и $\exists \bigcirc \varphi_j$. сопоставляются множества пропозициональных переменных $\mathrm{q}_{j,i}$. Кодирование равенств вида $\overline{\alpha}^{\delta(t(\delta(r)))} = f^{\mathrm{w}}_t(\overline{\alpha}^{\delta(r)})$ будем осуществлять с помощью специальных формул $\forall \bigcirc \varphi_j$ и $\exists \bigcirc \varphi_j$, которым также сопоставляются переменные $\mathrm{q}_{j,i}$. Более сложные формулы лвв с главной временнóй связкой, получающиеся из цели Θ, преобразовываются в д.н.ф., которые состоят из указанных выше пропозициональных переменных, называемых *базовыми*. Каждая основная вершина C_i (врéменная вершина D_i) схемы модели «кодируется» пропозициональной переменной $\mathrm{z}^{\mathrm{C}}_i$ $(\mathrm{z}^{\mathrm{D}}_i)$ и формулой лв θ^{C}_i (θ^{D}_i). При этом истинность $\mathrm{z}^{\mathrm{C}}_i$ $(\mathrm{z}^{\mathrm{D}}_i)$ эквивалентна истинности конечной конъюнкции (дизъюнкции) пропозициональных переменных $\mathrm{z}^{\mathrm{D}}_k$ $(\mathrm{z}^{\mathrm{C}}_k)$, каждая из которых кодирует сына D_k (C_k) вершины $(\mathrm{z}^{\mathrm{C}}_i)$ $(\mathrm{z}^{\mathrm{D}}_i)$. Поскольку истинность каждой пропозициональной переменной $\mathrm{z}^{\mathrm{C}}_i$ $(\mathrm{z}^{\mathrm{D}}_i)$ в свою очередь эквивалентна истинности конечной конъюнкции базовых переменных и их отрицаний, при последовательном раскрытии указанных выше эквивалентностей мы получаем формулы ограниченной глубины, обобщающие (в смысле работы [1]) дизъюнктивные и конъюнктивные нормальные формы. Кроме того, в формулу Θ^{pq}_0 добавляются пропозициональные переменные $\mathrm{r}^{\mathrm{C}}_{i,k}$, $\mathrm{r}^{\mathrm{D}}_{i,k}$, кодирующие дуги (C_i, D_k) и (D_i, C_k) графа схемы модели; а также дополнительные пропозициональные переменные $\mathrm{t}^{\mathrm{D}}_0$, $\mathrm{f}^{\mathrm{D}}_0$, $\mathrm{t}^{\mathrm{C}}_{i,\eta(i)}$, $\mathrm{t}^{\mathrm{D}}_{i,\eta(i),\kappa(i)}$ $(\mathrm{f}^{\mathrm{C}}_{i,\eta(i)}$, $\mathrm{f}^{\mathrm{D}}_{i,\eta(i)})$, соот-

ветствующие возможной $\top$-помеченности ($\bot$-помеченности) вершин схемы модели (см. [2]).

Далее возможны три случая. Если формула Θ_0^{pq} невыполнима, то полагаем Θ_1^{pq} равным Θ_0^{pq} и досрочно заканчиваем построение сведéния с ответом: «ИП $\mathcal{P}_{\mathcal{U}}$ для условия $\mathcal{U}$ не существует». Если формула $(\Theta_0^{\mathrm{pq}} \to \mathrm{t}_0^{\mathrm{D}})$ общезначима, то полагаем Θ_1^{pq} равным Θ_0^{pq} и переходим к третьему этапу сведéния. Если же не имеет места ни первое, ни второе, то на втором этапе строим по формуле Θ_0^{pq} формулу Θ_1^{pq}, моделируя процедуру фильтрования из работы [2]. Если формула Θ_1^{pq} оказалась невыполнимой, то построение сведéния заканчивается с ответом: «ИП $\mathcal{P}_{\mathcal{U}}$ для условия $\mathcal{U}$ не существует». Иначе следует переход к третьему этапу, имитирующему построение модели, на котором строится формула Θ_2^{pq}, описывается процедура построения ИП $\mathcal{P}_{\mathcal{U}}$ по формуле Θ_2^{pq}, после чего сведéние заканчивается с ответом: «Построена ИП $\mathcal{P}_{\mathcal{U}}$, удовлетворяющая условию $\mathcal{U}$».

Теорема 1. *Для каждого условия $\mathcal{U}$ алгоритм сведéния за конечное число шагов строит формулу Θ_1^{pq}, а в случае выполнимости Θ_1^{pq} — формулу Θ_2^{pq}.*

Теорема 2. *ИП $\mathcal{P}_{\mathcal{U}}$, удовлетворяющая условию $\mathcal{U}$, существует тогда и только тогда, когда выполнима формула Θ_1^{pq}.*

Теорема 3. *ИП $\mathcal{P}_{\mathcal{U}}$, построенная по формуле Θ_2^{pq}, удовлетворяет условию $\mathcal{U}$.*

Таким образом, поскольку задача распознавания выполнимости формул лв разрешима, построенное сведéние одновременно является и алгоритмом решения задачи синтеза ИП. Завершаемость, корректность и полнота этого алгоритма следует из теорем 1–3.

Работа выполнена при финансовой поддержке программы фундаментальных исследований Отделения математических наук РАН секция «Алгебраические и комбинаторные методы математической кибернетики» (проект «Задачи оптимального синтеза управляющих систем»).

Список литературы

1. Лупанов О. Б. О реализации функций алгебры логики формулами из конечных классов (формулами ограниченной глубины) в базисе $\&, \vee, \neg$ // Проблемы кибернетики. Вып. 6. — М.: Физматгиз, 1961. — С. 5–14.

2. Хелемендик Р. В. Алгоритм распознавания формул логики ветвящегося времени и эффективный алгоритм построения выводов общезначимых формул из аксиом. // Математические вопросы кибернетики. Вып. 15:

Сборник статей / Под редакцией О.Б.Лупанова. — М.: Физматлит, 2006. С. 217–266.

3. Хелемендик Р.В. О расширении типов игрового взаимодействия в языке игровых программ. // Материалы VI молодежной научной школы по дискретной математике и ее приложениям (Москва, 16–21 апреля 2007г.). Часть III. Под редакцией А.В.Чашкина. 2007. С. 30–35.

О ПРОГРАММЕ РАСПОЗНАВАНИЯ ВЫПОЛНИМОСТИ ФОРМУЛ ЛОГИКИ ВЫСКАЗЫВАНИЙ С ПОМОЩЬЮ МЕТОДА СЕМАНТИЧЕСКИХ ТАБЛИЦ

Р. В. Хелемендик (Москва)

1. Введение

Проблема распознавания выполнимости формул логики высказываний возникает в различных областях науки. При этом NP-полнота этой проблемы не препятствует успешному ее решению, если говорить о конкретных классах задач. В связи с этим представляет интерес применение различных методов к распознаванию выполнимости формул логики высказываний. В настоящей работе представлена программа распознавания выполнимости формул логики высказываний с помощью модифицированного метода семантических таблиц Бэта (см. [5]), один из вариантов которого был изложен в работе [3].

2. Описание алгоритма

Определение формулы логики высказываний, интерпретации, истинности формулы в интерпретации, выполнимости и общезначимости стандартные, и их можно найти, например, в работе [2]. Мы будем рассматривать формулы в базисе $\{\neg, \to, \wedge, \vee\}$, хотя вообще говоря описываемый ниже алгоритм может быть легко модифицирован для любого полного базиса, состоящего из функций не более чем от двух переменных. Метод семантических таблиц фактически является прямой проверкой формулы на выполнимость по определению истинности формулы в интерпретации. А именно, предположив, что формула φ выполнима, путём декомпозиции этой формулы будем искать подходящую интерпретацию I, в которой она была бы

истинна. Если мы разобрали все возможные интерпретации I и удостоверились, что формула φ не может быть истинна ни в одной из них (или нашли такую, что в ней φ истинна), то формула φ невыполнима (соответственно, выполнима). Отметим, что в методе семантических таблиц интерпретации рассматриваются «блоками», зависящими от структуры φ, в которых фиксируются лишь некоторые значения пропозициональных переменных и подформул формулы φ, благодаря чему число этих блоков зачастую существенно меньше, чем 2^n — числа интерпретаций — числа строк в таблице истинности.

Опишем алгоритм семантических таблиц детально. Мы будем представлять формулу списком её подформул, в котором каждая пропозициональная переменная встречается ровно один раз. При этом произвольная подформула исходной формулы может встречаться как один, так и несколько раз. Таким образом, такой список подформул детально описанный в работе [4], эквивалентен заданию формулы в виде схемы из функциональных элементов, которые предложены в работе [1] для ускорения выполнения эквивалентных преобразований формул.

Пусть $S = \langle \mathcal{T}, \mathcal{F} \rangle$ — пара множеств $\mathcal{T}$ и $\mathcal{F}$, называемая в дальнейшем *таблицей*, в которых содержатся номера подформул исходной формулы φ, представленной списком – обозначим этот список через G. Множество $\mathcal{T}$ будет называться множеством *положительных* (предположительно истинных) формул, а множество $\mathcal{F}$ будет называться множеством *отрицательных* (предположительно ложных) формул, причём объединение этих множеств непусто. Таблица S называется *противоречивой*, если пересечение множеств $\mathcal{T}$ и $\mathcal{F}$ непусто. Таблица S называется *тупиковой*, если она не противоречива, и каждый элемент множества $\mathcal{T} \cup \mathcal{F}$ является пропозициональной переменной с номером из списка G. Пусть $\mathbf{S} = \{S_1, \ldots, S_i, \ldots, S_k\}$ — множество таблиц S_i, где $1 \leqslant i \leqslant k$. Тогда метод семантических таблиц заканчивает свою работу в том и только в том случае, когда каждая таблица является противоречивой — с ответом «φ невыполнима», либо когда существует тупиковая таблица S_i. В последнем случае даётся ответ: «φ выполнима» и по таблице S_i строится интерпретация I. для которой утверждается $I \models \varphi$. Если переменная p_j, точнее её номер, входит во множество $\mathcal{T}_i$, то полагаем $I(p_j) = \top$; если входит во множество $\mathcal{F}_i$ — полагаем $I(p_j) = \bot$; если же номер этой пропозициональной переменной не входит ни в $\mathcal{T}_i$, ни в $\mathcal{F}_i$ (в оба множества номер этой переменной входить не может, так как таблица S_i непротиворечива), то значение $I(p_j)$ определяется произвольным образом, либо остаётся неопределённым. В начале применения метода семантических таблиц к формуле φ полагем $\mathbf{S} = \{\mathbf{S_1}\}$, где $S_1 = \langle \mathcal{T}_1, \mathcal{F}_1 \rangle$, где $\mathcal{T}_1$ состоит из единственного номера, под

которым в таблице G записана формула φ, а $\mathcal{F}_1 = \emptyset$.

Для формул в базисе $\{\neg, \to, \wedge, \vee\}$ каждый шаг метода семантических таблиц состоит в применении одного из 8 правил, которые подразделяются на две группы: линейные правила — $R(\neg+)$, $R(\neg-)$, $R(\wedge+)$, $R(\vee-)$, $R(\to -)$ и правила ветвления — $R(\vee+)$, $R(\to +)$, $R(\wedge-)$. Пусть множество $\mathbf{S} = \{S_i, \ldots, S_i, \ldots, S_k\}$ не содержит тупиковых таблиц, а каждая из таблиц этого множества непротиворечива. Пусть формула ψ, отличная от пропозициональной переменной, имеет номер m в таблице G, а $S_i = \langle \mathcal{T}_i, \mathcal{F}_i \rangle$. Тогда если $m \in \mathcal{T}_i$, то возможны 4 случая, соответствующих связкам $\{\neg, \to, \wedge, \vee\}$. Рассмотрим для примера случай связки $\wedge$:

$R(\wedge+)$: $\psi = (\psi_1 \wedge \psi_2)$, т.е. подформула ψ с номером $[m]$ представлена в виде «$\wedge$, $[m_1]$, $[m_2]$», где m_1 — номер подформулы ψ_1, а m_2 – номер подформулы ψ_2 в таблице G. Тогда

1) удаляем элемент $[m]$ (или помечаем его символом $*$ — см. доказательство теоремы 1) из множества $\mathcal{T}_i$;

2) добавляем элемент $[m_1]$ (если его ещё там нет) во множество $\mathcal{T}_i$;

3) добавляем элемент $[m_2]$ (если его ещё там нет) во множество $\mathcal{T}_i$.

Если $m \in \mathcal{F}_i$, то возможны также 4 случая. Рассмотрим для примера случай связки $\wedge$:

$R(\wedge-)$: $\psi = (\psi_1 \wedge \psi_2)$, т.е. подформула ψ с номером $[m]$ представлена в виде «$\wedge$, $[m_1]$, $[m_2]$», где m_1 — номер подформулы ψ_1, а m_2 — номер подформулы ψ_2 в таблице G. Тогда

1) удаляем элемент $[m]$ (или помечаем его символом $*$ — см. доказательство теоремы 1) из множества $\mathcal{F}_i$;

2) добавляем во множество $\mathbf{S}$ таблицу S_{k+1} такую, что $S_{k+1} = S_i$;

3) добавляем элемент $[m_1]$ (если его ещё там нет) во множество $\mathcal{F}_i$;

4) добавляем элемент $[m_2]$ (если его ещё там нет) во множество $\mathcal{F}_i$.

Каждое из правил для остальных 6 случаев аналогично либо правилу $R(\wedge+)$, либо правилу $R(\wedge-)$.

Для правила $R(\wedge-)$ (а также правил $R(\vee+)$ и $R(\to +)$) в алгоритме предусмотрены эвристики, позволяющие, например, не заводить новую таблицу в случае ложности одного из аргументов конъюнкции.

3. Обоснование алгоритма

Алгоритм с.т. называется *корректным*, если его ответ «формула φ выполнима» является правильным, а также является правильным и его ответ «формула φ истинна в построенной интерпретации I». Отметим, что данный алгоритм не зацикливается и всегда завершает свою работу за конечное число шагов, так как длина формулы φ конечна, и на каждом шаге работы алгоритма длины разбираемых формул уменьшаются. Алгоритм с.т. является полным, если в случае выполнимости формулы φ он даёт ответ «φ выполнима», а также предъявляет интерпретацию I и даёт ответ «формула φ истинна в интерпретации I». В силу свойства завершаемости алгоритма с.т. определение его полноты может быть переформулировано эквивалентным образом: алгоритм с.т. называется полным если его ответ «формула φ невыполнима» является верным.

Теорема 1. *Алгоритм семантических таблиц является корректным.*

Доказательство. Индукцией по длине формулы φ докажем, что данная формула истинна в построенной модели (интерпретации) I. Поскольку данная интерпретация получилась из некоторой тупиковой таблицы S_i, конъюнкция всех пропозициональных переменных из $\mathcal{T}_i$, а также отрицаний пропозициональных переменных из $\mathcal{F}_i$ истинна в этой интерепретации. Далее, рассмотрим протокол алгоритма с.т. в обратном порядке. В нём каждая помеченная символом $*$ формула зависит от одной (для связки отрицания), либо двух (для $\to$, $\wedge$, $\vee$) подформул, истинность которых в интерпретации I по индукционному предположению уже доказана. Отсюда согласно определению в интерпретации I истинна и данная помеченная формула, что даёт в конце концов истинность в I формулы φ.

Теорема 2. *Алгоритм семантических таблиц является полным.*

Доказательство. Номера формул из $\mathcal{T}_i$ и $\mathcal{F}_i$ будем здесь отождествлять с самими формулами, записанными в таблице G под соответствующими номерами. Обозначим для S_i, где $S_i = \langle \mathcal{T}_i, \mathcal{F}_i \rangle$ через $\breve{\mathcal{T}}_i$ ($\breve{\mathcal{F}}_i$) конъюнкцию (конъюнкцию отрицаний) формул, входящих в $\mathcal{T}_i$ ($\mathcal{F}_i$), либо формулу $\top$, если множество $\mathcal{T}_i$ ($\mathcal{F}_i$) пусто. Обозначим через $\breve{S}_i$ формулу $(\breve{\mathcal{T}}_i \wedge \breve{\mathcal{F}}_i)$. Пусть $\mathbf{S} = \{S_1, \ldots, S_i, \ldots, S_k\}$. Тогда обозначим через $\breve{\mathbf{S}}$ формулу $\bigvee_{i=1}^{k} \breve{S}_i$ (порядок скобок в этой формуле не фиксируем). Докажем эквивалентность двух формул — формулы $\breve{\mathbf{S}}$, полученной по множеству таблиц до применения любого из указанных в алгоритме семантических таблиц 8 правил, и формулы $\breve{\mathbf{S}}'$, полученной по множеству таблиц, изменившихся после применения соответствующего правила, т. е. общезначимость формулы $(\breve{\mathbf{S}} \equiv \breve{\mathbf{S}}')$.

Пусть $\mathbf{S} = \{S_1, \ldots, S_i, \ldots, S_k\}$, применяется правило $R(\wedge+)$ к формуле ψ_m, где $\psi_m = (\psi_{m_1} \wedge \psi_{m_2})$, $\mathcal{T}_i = \{\psi_m\} \cup (\mathcal{T}_i \setminus \psi_m)$; конъюнкцию формул из множества $\mathcal{T}_i \setminus \psi_m$ обозначим через $\breve{\mathcal{T}}_i^*$. После применения данного правила имеем $\mathbf{S}' = \{S'_1, \ldots, S'_i, \ldots, S'_k\}$, где $S'_j = S_j$ для всех $j \neq i$ и выполнено $S'_i = \langle (\mathcal{T}_i \setminus \{\psi_m\}) \cup \{\psi_{m_1}, \psi_{m_2}\}, \mathcal{F}_i \rangle$. Тогда

$$\breve{\mathcal{T}}_i = (\breve{\mathcal{T}}_i^* \wedge \psi_m) = (\breve{\mathcal{T}}_i^* \wedge (\psi_{m_1} \wedge \psi_{m_2})) = \breve{\mathcal{T}}'_i,$$

откуда следует общезначимость формул $(\breve{S}_i \equiv \breve{S}'_i)$, $(\breve{\mathbf{S}} \equiv \breve{\mathbf{S}}')$. Правила $(R\neg+)$, $(R\neg-)$, $(R \to -)$, $(R \vee -)$ рассматриваются аналогичным образом.

Пусть $\mathbf{S} = \{S_1, \ldots, S_i, \ldots, S_k\}$, применяется правило $R(\wedge-)$ к формуле ψ_m, где $\psi_m = (\psi_{m_1} \wedge \psi_{m_2})$, $\mathcal{F}_i = \{\psi_m\} \cup (\mathcal{F}_i \setminus \psi_m)$; конъюнкцию формул из множества $\mathcal{F}_i \setminus \psi_m$ обозначим через $\breve{\mathcal{F}}_i^*$. После применения данного правила имеем $\mathbf{S}' = \{S'_1, \ldots, S'_i, \ldots, S'_k, S'_{k+1}\}$, где $S'_j = S_j$ для всех $j \neq i$, $1 \leqslant j \leqslant k$, а $S'_i = \langle \mathcal{T}_i, (\mathcal{F}_i \setminus \{\psi_m\}) \cup \{\psi_{m_1}\} \rangle$, $S'_{k+1} = \langle \mathcal{T}_{k+1}, (\mathcal{F}_{k+1} \setminus \{\psi_m\}) \cup \{\psi_{m_2}\} \rangle$. Тогда

$$\begin{aligned}
&\breve{S}_i = (\breve{\mathcal{T}}_i \wedge (\breve{\mathcal{F}}_i^* \wedge \neg\psi_m)) = (\breve{\mathcal{T}}_i \wedge (\breve{\mathcal{F}}_i^* \wedge \neg(\psi_{m_1} \wedge \psi_{m_2}))) \equiv \\
&\equiv (\breve{\mathcal{T}}_i \wedge (\breve{\mathcal{F}}_i^* \wedge (\neg\psi_{m_1} \vee \neg\psi_{m_2}))) \equiv (\breve{\mathcal{T}}_i \wedge ((\breve{\mathcal{F}}_i^* \wedge \neg\psi_{m_1}) \vee (\breve{\mathcal{F}}_i^* \wedge \neg\psi_{m_2}))) \equiv \\
&\equiv ((\breve{\mathcal{T}}_i \wedge (\breve{\mathcal{F}}_i^* \wedge \neg\psi_{m_1})) \vee (\breve{\mathcal{T}}_i \wedge (\breve{\mathcal{F}}_i^* \wedge \neg\psi_{m_2}))) \equiv (\breve{S}'_i \vee \breve{S}'_{k+1}),
\end{aligned}$$

откуда следует общезначимость формулы $(\breve{\mathbf{S}} \equiv \breve{\mathbf{S}}')$. Правила $(R \to +)$ и $(R \vee +)$ рассматриваются аналогично.

В случае ответа «φ невыполнима» все семантические таблицы противоречивы, откуда следует, что соответствующая им формула-дизъюнкция невыполнима. Поскольку число шагов алгоритма семантических таблиц конечно, и для каждого шага — применения правила — доказана общезначимость формулы $(\breve{\mathbf{S}} \equiv \breve{\mathbf{S}}')$, мы получаем, что невыполнима и начальная формула, соответствующая единственной таблице с единственной предположительно истинной формулой φ, т. е. начальная формула, совпадающая с формулой φ невыполнима. Теорема доказана.

4. Описание программы

Программа распознаёт выполнимость произвольной формулы в базисе $\{\wedge, \vee, \to, \neg\}$ и строит в случае выполнимости некоторую модель. В программе предусмотрен ввод формулы как в виде строки, так и в виде списка подформул. На число переменных и длину строки ограничения не налагаются. Возможен выбор между исследованием формулы на выполнимость и исследованием на общезначимость, а также получение и сохранение протокола применения каждого правила к формуле и соответствующей таблице. В силу естественности правил метода семантических таблиц в случае придания пропозициональным переменным конкретных высказываний

данный протокол позволяет получить анализ исследуемой формулы в естественном языке. Предусмотрена возможность нахождения всех моделей. В программе предусмотрена генерация случайных списков и формул логики высказываний с заданием числа переменных и числа строк. Фрагмент статистики работы программы на таких формулах содержится в приводимой ниже таблице.

Программа прошла апробацию на практических занятиях по курсу математической логики в МАТИ-РГТУ имени К.Э.Циолковского на кафедре «Кибернетика».

Общее число разобранных формул	100000	100000
Число переменных в формулах	5	20
Средняя длина формул в символах	546,3	217,6
Среднее число строк в списках	73,2	76,3
Число выполнимых формул	87204	97456
Число невыполнимых формул	12796	2544
Среднее число проделанных шагов	630,8	118,8
Среднее время (в миллисекундах)	2,1	0,4
Среднее число шагов для выполнимых формул	215,0	42,2
Среднее число шагов для невыполнимых формул	3464,2	3051,1
Среднее время для выполнимых формул	0,5	0,1
Среднее время для невыполнимых формул	13,2	13,1

Работа выполнена при финансовой поддержке программы фундаментальных исследований Отделения математических наук РАН секция «Алгебраические и комбинаторные методы математической кибернетики» (проект «Задачи оптимального синтеза управляющих систем»).

Список литературы

1. Ложкин С. А. Лекции по основам кибернетики: Учебное пособие. — М.: Издательский отдел Факультета ВМиК МГУ им. М.В.Ломоносова, 2004.

2. Мендельсон Э. Введение в математическую логику: Пер. с англ. // Под ред. С.И.Адяна. — 3-е изд. — М.: Наука. Главная редакция физико-математической литературы, 1984.

3. Хелемендик Р. В. Алгоритм распознавания формул логики ветвящегося времени и эффективный алгоритм построения выводов общезначимых формул из аксиом. // Математические вопросы кибернетики. Вып. 15: Сборник статей / Под ред. О.Б.Лупанова. — М.: Физматлит, 2006. С. 217–266.

4. Хелемендик Р. В. Элементы математической логики и возможности ее применения // Учебное пособие. М.: МАТИ, 2009. В печати.

5. Beth E. W., The foundations of mathematics, Amsterdam, 1959; выдержки даны в русском переводе: Математическая теория логического вывода. М.: Наука, 1967. С. 191–199.

О НАДЕЖНОСТИ СХЕМ ИЗ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ В БАЗИСАХ, СОДЕРЖАЩИХ ФУНКЦИИ СПЕЦИАЛЬНОГО ВИДА

В. В. Чугунова (Пенза)

Впервые задачу синтеза надежных схем из ненадежных элементов рассматривал Дж. фон Нейман [1]. Он предполагал, что все элементы схемы независимо друг от друга с вероятностью ε ($\varepsilon < 1/2$) подвержены инверсным неисправностям на выходах, когда функциональный элемент с приписанной ему булевой функцией $e(\tilde{x})$ в неисправном состоянии реализует $\bar{e}(\tilde{x})$. Для повышения надежности схем Дж. фон Нейман использовал схему, реализующую функцию голосования $g_1(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3$. Позднее Алехина М. А. и Аксенов С. И. ввели в рассмотрение новые классы функций, корректирующих ошибки: $G_1 = \{x_1^{\delta_1}x_2^{\delta_2} \vee x_1^{\delta_1}x_3^{\delta_3} \vee x_2^{\delta_2}x_3^{\delta_3}\}$, $G_2 = \{x_1^{\delta_1}x_2^{\delta_2} \vee x_3^{\delta_3}x_4^{\delta_4}\}$, $G_3 = \{(x_1^{\delta_1} \vee x_2^{\delta_2})\&(x_3^{\delta_3} \vee x_4^{\delta_4})\}$ (где $\delta_i \in \{0, 1\}$ и $i = 1, 2, 3, 4$).

Аксенов С. И. показал [2], что при инверсных неисправностях на выходах элементов наличие любой из функций множества $G = G_1 \cup G_2 \cup G_3$ в заданном полном базисе Б гарантирует реализацию произвольной булевой функции схемой, функционирующей с вероятностью ошибки не больше $\varepsilon + c\varepsilon^2$, где $\varepsilon \leqslant d$, а c и d — некоторые положительные константы.

В работе [3] Алехина М. А. ввела новый класс функций M_k, повышающих надежность схем, и доказала для него теорему 1. Множество M_k — множество всех булевых функций $m(x_1, ..., x_k)$ ($k \geqslant 3$), обладающих свойством: найдется такой набор $(b_1, ..., b_k)$, что на нем и всех соседних с ним

наборах функция m принимает значение 0, а на наборе $(\bar{b}_1, ..., \bar{b}_k)$ и всех соседних с ним наборах — значение 1. Наборы $(b_1, ..., b_k)$ и $(\bar{b}_1, ..., \bar{b}_k)$ называются *характеристическими наборами* функции $m(x_1, ..., x_k)$.

Теорема 1 [3]. *Пусть $f(x_1, ..., x_n)$ — произвольная булева функция, а S — схема, ее реализующая с ненадежностью $P(S) \leqslant p$. Пусть схема S_m реализует функцию $m(x_1, ..., x_k) \in M_k$ и $P(S_m) \leqslant p$. Обозначим υ^1 и υ^0 — вероятности ошибок схемы S_m на характеристических наборах. Тогда существует схема $\phi(S)$, реализующая функцию f, такая что ее ненадежность удовлетворяет неравенству $P(\phi(S)) \leqslant \max\{\upsilon^1, \upsilon^2\} + cp^2$, где положительная константа $c \leqslant kC_k^{[k/2]}$.*

Следствие 1. *Пусть полный базис Б содержит функцию $m(x_1, \ldots, x_k)$ из класса M_k, а функциональные элементы с вероятностью ε подвержены инверсным неисправностям на выходах. Пусть f — произвольная булева функция, а S — схема, реализующая ее с ненадежностью $P(S) \leqslant s\varepsilon$ (s — положительная константа). Тогда функцию f можно реализовать такой схемой A над Б, что $P(A) \leqslant \varepsilon + c\varepsilon^2$, где c — константа, $0 < c \leqslant kC_k^{[k/2]}s^2$.*

Из рассмотренных выше результатов следует, что существуют такие булевы функции, наличие которых в рассматриваемом базисе при инверсных неиправностях на выходах позволяет реализовать почти все булевы функции асимптотически оптимальными по надежности схемами с ненадежностью ε (при $\varepsilon \to 0$).

Пусть функциональные элементы подвержены инверсным неисправностям на входах. Эти неисправности характеризуются тем, что поступающее на каждый вход элемента значение a ($a \in \{0, 1\}$) с вероятностью ε ($0 < \varepsilon < 1/2$) может превратиться в значение $\bar{a}$. Очевидно, что при инверсных неисправностях на входах с увеличением t — числа входов каждого элемента базиса Б, его ненадежность увеличивается до $t\varepsilon$. Возникает вопрос: можно ли при инверсных неисправностях на входах элементов реализовать произвольную булеву функцию схемой с ненадежностью порядка $\varepsilon^{[t/2]+1}$ (где $t \geqslant 3$)? Ответ на него получен в этой статье.

Пусть $P_{\bar{f}(\tilde{a})}(S, \tilde{a})$ — вероятность появления значения $\bar{f}(\tilde{a})$ на выходе схемы S, реализующей булеву функцию $f(\tilde{x})$, при входном наборе $\tilde{a}$. Ненадежность $P(S)$ схемы S определяется как максимальное из чисел $P_{\bar{f}(\tilde{a})}(S, \tilde{a})$ при всевозможных входных наборах $\tilde{a}$. Надежность схемы S равна $1 - P(S)$.

Рассмотрим множество H_{2k+1}, содержащее функции $h(x_1, ..., x_{2k+1})$, существенно зависящие от $2k+1$ (где $k = 1, 2, ...$) переменных и обладающие свойствами:

1) найдется такой набор значений переменных $\tilde{b} = (b_1, ..., b_{2k+1})$, что на нем и на всех наборах $\tilde{a} = (a_1, ..., a_{2k+1})$, таких, что

$$\rho(\tilde{a}, \tilde{b}) = \sum_{i=1}^{2k+1} |a_i - b_i| \leqslant k,$$

функция принимает значение 0, то есть $h(\tilde{b}) = h(\tilde{a}) = 0$.

2) на наборе $\bar{\tilde{b}} = (\bar{b}_1, ..., \bar{b}_{2k+1})$ и на всех наборах $\tilde{c} = (c_1, ..., c_{2k+1})$, таких, что

$$\rho(\tilde{c}, \bar{\tilde{b}}) = \sum_{i=1}^{2k+1} |c_i - b_i| \leqslant k,$$

функция принимает значение 1, то есть $h(\bar{\tilde{b}}) = h(\tilde{c}) = 1$.

Наборы $\tilde{b} = (b_1, ..., b_{2k+1})$ и $\bar{\tilde{b}} = (\bar{b}_1, ..., \bar{b}_{2k+1})$ назовем характеристическими наборами функции $h(x_1, ..., x_{2k+1})$.

Функции множества H_{2k+1} можно представить в виде ДНФ. Для этого фиксируем числа $\delta_1, \delta_2, ..., \delta_{2k+1} \in \{0, 1\}$ и получаем соответствующую им функцию:

$$h(x_1, ..., x_{2k+1}) = \bigvee_{\substack{i_1, i_2, ..., i_{k+1} \in \{1,2,...,2k+1\} \\ i_j \neq i_p}} x_{i_1}^{\delta_{i_1}} x_{i_2}^{\delta_{i_2}} ... x_{i_{k+1}}^{\delta_{i_{k+1}}},$$

где под знаком дизъюнкции стоят все возможные элементарные конъюнкции ранга $k+1$ от $2k+1$ переменных (их C_{2k+1}^{k+1} штук).

Число функций во множестве H_{2k+1} равно: $|H_{2k+1}| = 2^{2k+1}$.

Пример 1. При $k = 1$ множество рассматриваемых функций H_3 имеет вид: $h(x_1, x_2, x_3) = x_1^{\delta_1} x_2^{\delta_2} \vee x_1^{\delta_1} x_3^{\delta_3} \vee x_2^{\delta_2} x_3^{\delta_3}$, где $\delta_i \in \{0, 1\}, i = 1, 2, 3$.

Пример 2. При $k = 2$ и $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = 1$ функция из множества H_5 может быть задана СДНФ: $h(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3x_4x_5 \vee$
$\vee\, \bar{x}_1x_2x_3x_4x_5 \vee x_1\bar{x}_2x_3x_4x_5 \vee x_1x_2\bar{x}_3x_4x_5 \vee x_1x_2x_3\bar{x}_4x_5 \vee x_1x_2x_3x_4\bar{x}_5 \vee$
$\vee\, \bar{x}_1\bar{x}_2x_3x_4x_5 \vee \bar{x}_1x_2\bar{x}_3x_4x_5 \vee \bar{x}_1x_2x_3\bar{x}_4x_5 \vee \bar{x}_1x_2x_3x_4\bar{x}_5 \vee x_1\bar{x}_2\bar{x}_3x_4x_5 \vee$
$\vee\, x_1\bar{x}_2x_3\bar{x}_4x_5 \vee x_1\bar{x}_2x_3x_4\bar{x}_5 \vee x_1x_2\bar{x}_3\bar{x}_4x_5 \vee x_1x_2\bar{x}_3x_4\bar{x}_5 \vee x_1x_2x_3\bar{x}_4\bar{x}_5$. Минимизируя СДНФ, получим: $h(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3 \vee x_1x_2x_4 \vee x_1x_2x_5 \vee$
$\vee\, x_2x_3x_4 \vee x_2x_3x_5 \vee x_3x_4x_5 \vee x_2x_4x_5 \vee x_1x_3x_4 \vee x_1x_3x_5 \vee x_1x_4x_5$.

В случае $k = 2$ и произвольных $\delta_i \in \{0, 1\}$, рассуждая аналогично, получим: $h(x_1, x_2, x_3, x_4, x_5) = x_1^{\delta_1} x_2^{\delta_2} x_3^{\delta_3} \vee x_1^{\delta_1} x_2^{\delta_2} x_4^{\delta_4} \vee x_1^{\delta_1} x_2^{\delta_2} x_5^{\delta_5} \vee x_2^{\delta_2} x_3^{\delta_3} x_4^{\delta_4} \vee$
$\vee\, x_2^{\delta_2} x_3^{\delta_3} x_5^{\delta_5} \vee x_3^{\delta_3} x_4^{\delta_4} x_5^{\delta_5} \vee x_2^{\delta_2} x_4^{\delta_4} x_5^{\delta_5} \vee x_1^{\delta_1} x_3^{\delta_3} x_4^{\delta_4} \vee x_1^{\delta_1} x_3^{\delta_3} x_5^{\delta_5} \vee x_1^{\delta_1} x_4^{\delta_4} x_5^{\delta_5}$, где $\delta_i \in \{0, 1\}, i = 1, 2, 3, 4, 5$.

Теорема 2. *Пусть функция* $h(x_1, \ldots, x_{2k+1}) \in H_{2k+1}$ *содержится в полном базисе* B, *а функциональные элементы с вероятностью* ε *подвержены инверсным неисправностям на входах. Допустим, что произвольную булеву функцию* $f(\tilde{x})$ *можно реализовать такой схемой* S, *что* $P(S) \leqslant p$. *Тогда функцию* $f(\tilde{x})$ *можно реализовать такой схемой* $\varphi(S)$ *над* B, *что*

$$P(\varphi(S)) \leqslant a\varepsilon^{k+1} + (2k+1)ap^2, a = C_{2k+1}^{k+1}. \quad (1)$$

Доказательство. Пусть $f(\tilde{x})$ — произвольная булева функция, а S — схема, реализующая ее с ненадежностью $P(S) \leqslant p$ в базисе B, содержащем функцию $h(x_1, ..., x_{2k+1})$, удовлетворяющую условиям теоремы 2. Пусть элемент E_h реализует функцию $h(x_1, ..., x_{2k+1})$ и $P(E_h) \leqslant p$. Так как множество функций $H_{2k+1} \subset M_{2k+1}$, то для функций $h(x_1, ..., x_{2k+1})$ утверждение теоремы 1 справедливо.

Найдем вероятности ошибок на выходе функционального элемента E_h на характеристических наборах:

$$\upsilon^1 = \upsilon^0 = C_{2k+1}^{k+1}\varepsilon^{k+1}(1-\varepsilon)^k + C_{2k+1}^{k+2}\varepsilon^{k+2}(1-\varepsilon)^{k-1} + ... + C_{2k+1}^{2k+1}\varepsilon^{2k+1} \leqslant C_{2k+1}^{k+1}\varepsilon^{k+1}.$$

Используя теорему 1, по схеме S построим такую схему $\phi(S)$, ненадежность которой: $P(\phi(S)) \leqslant a\varepsilon^{k+1} + cp^2$ (где $a = C_{2k+1}^{k+1}$, $c \leqslant (2k+1)a$).

Схема $\phi(S)$ является искомой схемой $\varphi(S)$. Теорема 2 доказана.

Следствие 1. При $k = 1$ неравенство (1) принимает вид (2):

$$P(\varphi(S)) \leqslant 3\varepsilon^2 + 9p^2. \quad (2)$$

Пусть B^1 — произвольный конечный полный базис, содержащий хотя бы одну из функций множества H_3, а m — наибольшее число входов элементов базиса B^1 ($m \geqslant 3$). Тогда в базисе B^1 справедлива теорема 3.

Теорема 3. *При* $\varepsilon \leqslant 1/(648m^2)$ *любую булеву функцию* $f(\tilde{x})$ *в полном конечном базисе* B^1 *можно реализовать такой схемой* S, *ненадежность которой* $P(S) \leq 3\varepsilon^2 + \varepsilon^3$.

Для доказательства теоремы 3 используем леммы 1 и 2.

Лемма 1 [2]. *Если* B — *конечный полный базис, тогда функцию штрих Шеффера* $x|y$ *можно реализовать над* B *схемой, в которой не более шести функциональных элементов.*

Лемма 2 [4]. *Если схема* S^* *в произвольном базисе* B *реализует функцию штрих Шеффера* $x|y$ *с ненадежностью* $P(S^*) \leqslant \mu$, *то при* $\mu \leqslant 1/50$

любую булеву функцию $f(\tilde{x})$ в базисе B можно реализовать схемой S, ненадежность которой $P(S) \leqslant 4\mu$.

Доказательство теоремы 3. В базисе B^1, содержащем хотя бы одну из функций множества H_3, можно построить схему S^*, реализующую функцию штрих Шеффера $x|y$ и состоящую из не более шести функциональных элементов (лемма 1), то есть $P(S^*) \leqslant 6m\varepsilon$, тогда $\mu \leqslant 6m\varepsilon$, где m — наибольшее число входов элементов базиса B^1 ($m \geqslant 3$). Следовательно, используя лемму 2, получим: при $\varepsilon \leqslant 1/(300m)$ любую булеву функцию $f(\tilde{x})$ в базисе B^1 можно реализовать схемой $\tilde{S}$, ненадежность которой $P(\tilde{S}) \leqslant 24m\varepsilon$.

Используя следствие 1 из теоремы 2, по схеме $\tilde{S}$ построим схему $\varphi(\tilde{S})$, ненадежность которой $P(\varphi(\tilde{S})) \leqslant 3\varepsilon^2 + 9(24m\varepsilon)^2 \leqslant 9\varepsilon$ при $\varepsilon \leqslant \min\{1/(300m); 1/(24 \cdot 9 \cdot 3m^2)\} = 1/(648m^2)$ (по формуле 2). Применяя теорему 2 еще раз, по схеме $\varphi(\tilde{S})$ построим схему $\varphi^2(\tilde{S})$, для которой имеет место оценка $P(\varphi^2(\tilde{S})) \leqslant 3\varepsilon^2 + 9(9\varepsilon)^2 = 732\varepsilon^2 \leqslant \varepsilon$ при $\varepsilon \leqslant 1/(648m^2)$. На четвертом шаге итерации построим схему $\varphi^3(\tilde{S})$, ненадежность которой удовлетворяет неравенству $P(\varphi^3(\tilde{S})) \leqslant 3\varepsilon^2 + 9(\varepsilon)^2 = 12\varepsilon^2$ при $\varepsilon \leqslant 1/(648m^2)$. По схеме $\varphi^3(\tilde{S})$ построим схему $\varphi^4(\tilde{S})$, реализующую $f(\tilde{x})$ с ненадежностью $P(\varphi^4(\tilde{S})) \leqslant 3\varepsilon^2 + 9(12\varepsilon^2)^2 = 3\varepsilon^2 + 1296\varepsilon^4 \leqslant 3\varepsilon^2 + \varepsilon^3$ при $\varepsilon \leqslant 1/(648m^2)$. Схема $\varphi^4(\tilde{S})$ искомая, т. е. $S = \varphi^4(\tilde{S})$.

Теорема 3 доказана.

Теорема 4. *Пусть m — наибольшее число входов элементов в полном конечном базисе B, содержащем хотя бы одну функцию $h(x_1, ..., x_{2k+1})$ множества H_{2k+1} ($m \geqslant 3$), тогда любую булеву функцию $f(\tilde{x})$ в базисе B при $\varepsilon \leqslant \frac{1}{24am^2(2k+1)}$ можно реализовать схемой S, ненадежность которой $P(S) \leqslant a\varepsilon^{k+1} + \varepsilon^{k+2}$, где $a = C_{2k+1}^{k+1}$, $k = 1, 2, ...$.*

Доказательство. Проведем индукцией по числу k.

При $k = 1$ утверждение верно (см. теорему 3).

При $k = 2, 3, 4$ утверждение верно (доказывается как теорема 3).

Допустим, что при $k \geqslant 5$ утверждение теоремы 4 верно, то есть справедливо неравенство: $P(S) \leqslant a\varepsilon^k + \varepsilon^{k+1} \leqslant (a+1)\varepsilon^k$ при $\varepsilon \leqslant \frac{1}{24m^2 C_{2k-1}^k (2k-1)}$, тогда при $k+1 \geqslant 6$, по теореме 2, получим: $P(S) \leqslant a\varepsilon^{k+1} + (2k+1)a(a+1)^2\varepsilon^{2k} \leqslant$

$\leqslant a\varepsilon^{k+1} + \frac{1}{2^6}(1+\frac{1}{a})\frac{1}{a}\varepsilon^{2k-3} \leqslant a\varepsilon^{k+1} + \varepsilon^{2k-3} \leqslant a\varepsilon^{k+1} + \varepsilon^{k+1}$ при $\varepsilon \leqslant \frac{1}{24am^2(2k+1)}$, т.е. теорема 4 верна.

Таким образом, показано, что при инверсных неисправностях на входах элементов наличие хотя бы одной функции $h(x_1, ..., x_{2k+1}) \in H_{2k+1}$ в

полном конечном базисе B позволяет реализовать все булевы функции схемами с ненадежностью не более $a\varepsilon^{k+1}+\varepsilon^{k+2}$, где $a = C_{2k+1}^{k+1}$, $\varepsilon \leqslant \frac{1}{24am^2(2k+1)}$, m — наибольшее число входов элементов в базисе B.

Список литературы

1. фон Нейман Дж. Вероятностная логика и синтез надежных организмов из ненадежных компонент // Автоматы. М.: Изд-во иностр. лит., 1956. — С. 68–139.

2. Аксенов С. И. О надежности схем над произвольной полной системой функций при инверсных неисправностях на выходах элементов // Известия высших учебных заведений. Поволжский регион. Естественные науки. — 2005. — N 6 (21). — С. 42–55.

3. Алехина М. А. О функциях и схемах, корректирующих ошибки // Материалы XVI Международной школы-семинара «Синтез и сложность управляющих систем» — М.: Изд-во мех.-мат. ф-та МГУ, 2006. — С. 8–12.

4. Алехина М. А., Чугунова В. В. Об асимптотически наилучших по надежности схемах в базисе $\{\&, \vee, \bar{}\}$ при инверсных неисправностях на входах элементов // Дискретный анализ и исследование операций. — Новосибирск: Изд-во института математики. — Октябрь–декабрь 2006 г. — Сер. 1. — Т. 13. — N 4. — С. 3–17.