УДК 004.021 DOI: 10.31114/2078-7707-2022-1-21-25

Ортогонализация системы ДНФ булевых функций

С.Н. Кардаш

Объединенный институт проблем информатики Национальной Академии Наук Беларуси, Республика Беларусь, kardash77@gmail.com

***Аннотация* — Для решения многих задач синтеза, диагностики и анализа надежности технических систем используется представление булевых функций в виде дизъюнктивных нормальных форм (ДНФ). На практике бывает полезно иметь такие системы ДНФ, в которых все входящие в них элементарные конъюнкции взаимно ортогональны. Для получения таких систем необходимо проводить ортогонализацию исходных систем ДНФ. В настоящей работе приводится оригинальный алгоритм решения задачи ортогонализации. Сообщается о разработке компьютерной программы, решающей задачу ортогонализации системы ДНФ. Приводятся результаты экспериментального исследования, подтверждающие эффективность разработанного алгоритма.**

***Ключевые слова* — булева функция, дизъюнктивная нормальная форма, ортогональность элементарных конъюнкций.**

I. Введение

Для решения многих задач синтеза, диагностики и анализа надежности технических систем используется представление булевых функций в виде дизъюнктивных нормальных форм (ДНФ). Часто бывает полезно иметь такие системы ДНФ, в которых все входящие в них элементарные конъюнкции взаимно ортогональны. Для получения таких систем необходимо проводить ортогонализацию исходных систем ДНФ. В работах [1-3] даны как необходимые понятия, так и идеи, способствующие решению этой задачи. В настоящей работе приводится алгоритм, на основе которого разработана компьютерная программа, решающая задачу ортогонализации системы ДНФ, и результаты ее экспериментального исследования.

В случае небольшого числа переменных задачу ортогонализации системы ДНФ можно решить, разложив дизъюнктивно каждую элементарную конъюнкцию по всем отсутствующим в ней переменным, от которых зависят функции, и после приведения подобных получить в результате совершенную ДНФ. Однако такой способ может оказаться неприемлем, когда переменных много. В частности, для системы ДНФ булевых функций, зависящих от n переменных, число конъюнкций в ортогонализованной системе может достигать 2^n.

II. Основные определения

Представим исходную систему ДНФ в матричном виде парой матриц – U (троичной) и S (булевой). Столбцы матрицы U соответствуют аргументам системы, а строки задают элементарные конъюнкции. Столбцы матрицы S соответствуют функциям системы, а единичные значения элементов в матрице S отмечают вхождения соответствующих конъюнкций в ДНФ функций исходной системы.

Строки троичных матриц будем задавать троичными векторами, а строки булевых матриц – булевыми.

Строке ***u*** матрицы ***U*** соответствует строка ***s*** матрицы ***S***.

Троичный вектор ***u*** будем представлять парой булевых векторов *U0* и *U1*, где вектор *U0* своими единичными компонентами задает нулевые компоненты вектора ***u***, а вектор *U1* своими единичными компонентами задает единичные компоненты вектора ***u***.

Аналогично, троичный вектор ***w*** представляется парой булевых векторов *W0* и *W1*.

Булевы векторы ***g*** и ***s*** представляются булевыми векторами *G1* и *S1* соответственно.

В рассмотрение вводится «нулевой» булев вектор ***Z*** размерности, равной размерности векторов *G1* и *S1*.

Определим следующие бинарные отношения на множестве векторов одинаковой размерности.

Ортогональность. Троичные векторы ***u*** и ***w*** ортогональны по *i*-й компоненте, если и только если *i*-я компонента имеет значение 0 в одном из этих векторов и 1 – в другом. Троичные векторы ортогональны, если они ортогональны хотя бы по одной компоненте.

Пересечение. Если векторы ***u*** и ***w*** не ортогональны, то они находятся в отношении пересечения. Это понятие согласуется с понятием пересечения множеств: пересекающиеся троичные векторы представляют пересекающиеся интервалы.

Примером пересекающихся векторов являются векторы (1 – – 0 – 1) и (1 0 – – 1–), которые имеют общий элемент (1 0 – 0 1 1).

Поглощение. Троичный вектор ***w*** поглощает вектор ***u***, тогда и только тогда, когда все компоненты вектора ***w***, значения которых отличны от «–», совпадают с одноименными компонентами вектора ***u***, т.е. выполняется соотношение

$$\neg(W0 \vee W1) \wedge \neg(U0 \vee U1) = \neg(U0 \vee U1).$$

Булев вектор ***g*** поглощает булев вектор ***s***, если выполняется соотношение

$$\neg G1 \wedge S1 = Z.$$

Здесь знаки «¬», «∧», «∨» означают логические операции «инверсия», «конъюнкция» и «дизъюнкция» соответственно.

Склеивание булевых векторов. Два булевых вектора можно заменить одним вектором, у которого значения компонент определяются следующим образом: компонента, в которой соответствующая компонента хотя бы одного исходного вектора имела единичное значение, приобретает значение «1». Значения остальных компонент получают значение «0».

Склеивание соседних векторов. Троичные векторы ***u*** и ***v*** являются соседними, если по некоторой *i*-й компоненте они ортогональны, а значения остальных одноименных компонент совпадают. Два соседних вектора можно заменить одним, где значения компонент определяются следующим образом. Компонента, по которой исходные вектора ортогональны, приобретает значение «–». Значения остальных компонент совпадают со значениями соответствующих компонент исходных векторов.

Расщепление троичных векторов. Ключевой операцией для данного алгоритма является операция расщепления троичных векторов. Два троичных вектора ***u*** и ***w*** можно заменить тремя множествами *V1, V2, V3* троичных векторов, находящихся в следующих отношениях. Один из векторов (путь это будет ***u***) образует единственный элемент множества *V1*. Множество *V2* образует единственный элемент, являющийся пересечением векторов ***u*** и ***w***. Множество *V3* составляют вектора, полученные из вектора ***w*** следующим образом. Вектор ***w***, имеющий значение «–» в *i*-й компоненте, можно заменить парой векторов, один из которых получается из ***v*** присвоением его *i*-й компоненте значения 0, другой – значения 1. Последовательное применение этой процедуры ко всем компонентам троичного вектора ***v***, имеющим значение «–» на тех позициях, у которых вектор ***u*** имеет определенные значения, и приводит к множеству векторов, образующих множество *V3*. Вектора этого множества попарно ортогональны и каждый из них ортогонален векторам из множеств *V1* и *V2*.

Например, если

u = (1 0 – 0 1 1),

w = (1 – 1 0 – 1),

то *V1* = (1 0 – 0 1 1),

V2 = (1 0 1 0 1 1),

V3 = {(1 1 1 0 – 1), (1 0 1 0 0 1)}.

III. Постановка задачи

Пусть в матричном виде задана система ДНФ булевых функций. Требуется построить эквивалентную ей ортогонализованную систему, содержащую минимальное число элементарных конъюнкций.

Для решения этой задачи предлагается приближенный эвристический алгоритм, основная идея которого состоит в поиске пар не ортогональных строк матрицы ***U***, их расщеплении и последующем склеивании или поглощении продуктов разложения.

Описание алгоритма

Результат ортогонализации представляется парой матриц – ***W*** (троичной) и ***G*** (булевой).

Строке ***w*** матрицы ***W*** соответствует строка ***g*** матрицы ***G***.

Результат разложения двух строк представляется троичной матрицей ***V***, строка ***v*** которой представляется парой булевых векторов *V0* и *V1*.

Матрица ***W*** полагается пустой.

1. В матрицу ***W*** заносится первая строка ***u*** матрицы ***U***, а в матрицу ***G*** заносится первая строка ***s*** матрицы ***S***.
2. Из матрицы ***U*** выбирается очередная строка ***u***.
   Если все строки матрицы ***U*** просмотрены – переход на п. 3.
   Иначе –
   2.1. Выбирается очередная (выбор начинается с первой) строка ***w*** матрицы ***W***.
   Если все строки матрицы ***W*** просмотрены – переход на п. 3.
   Иначе – строки ***u*** и ***w*** сравниваются.
   Если ***u*** и ***w*** ортогональны – переход на п.2.1.
   Если ***u*** и ***w*** не ортогональны, то если строки ***u*** и ***w*** совпадают, то строки ***s*** и ***g*** склеиваются, и результат склеивания заменяет строку ***g*** в матрице ***G***. Переход на п.3.
   Если строки ***u*** и ***w*** не совпадают, то если ***w*** поглощает ***u***, а ***g*** поглощает ***s***, то переход на п.3.
   Если строки ***u*** и ***w*** не совпадают, и если ***u*** поглощает ***w***, а ***s*** поглощает ***g***, то строка ***w*** матрицы ***W*** заменяется строкой ***u***. Переход на п.3.
   Иначе – для строк ***u*** и ***w*** выполняется процедура разложения – строится троичная матрица ***V***, содержащая *n (n>1)* строк.
   2.2. Первые *n-1* строк матрицы ***V*** переносятся в матрицу ***W***, а соответствующие *n-1* строк матрицы ***G***, являются копиями строки *S1*.
   Последняя строка матрицы ***V*** добавляется в матрицу ***W***, строки ***s*** и ***g*** склеиваются, и

результат склеивания добавляется в матрицу ***G***. Переход на п.2.

3. Если на шаге 2 были склеивания или поглощения, то переход на п.2.
   Иначе – строка ***u*** добавляется в матрицу ***W***, а строка ***s*** – в матрицу ***G***. Переход на п.2.
4. Если на шаге 2 не было разложений, то переход на п.5.
   Иначе – в матрицу ***U*** в обратном порядке переносятся строки матрицы ***W***. Переход на п.1.

5. Для полученных матриц ***U*** и ***S*** производится процедура, направленная на сокращение числа строк, путем их склеивания.
6. Конец.

Процедура разложения троичных векторов

Матрица ***V*** полагается пустой.

В рассмотрение вводится булев вектор *Y*:

$$Y = (W0 \lor W1) \land (\neg(U0 \lor U1)).$$

1. До тех пор, пока в векторе *Y* имеются единичные компоненты, определяется номер *j* его крайней левой единичной компоненты. Если единичных компонент нет, то переход на п.3.
   Значение 0 присваивается *j*-й компоненте вектора *Y*.
   Полагается *V0 = U0, V1 = U1*.
   Значение *j*-й компоненты вектора *W1* присваивается *j*-й компоненте вектора *V0*.
   Значение *j*-й компоненты вектора *W0* присваивается *j*-й компоненте вектора *V1*.
   В матрицу ***V*** добавляется вектор ***v***. Полагается *k = j*. Переход на п.1.
2. Если п.1. выполнялся хотя бы один раз, то *k*-й компоненте вектора *V0* присваивается значение *k*-й компоненты вектора *W0* и *k*-й компоненте вектора *V1* присваивается значение *k*-й компоненты вектора *W1*.
   Иначе полагается *V0 = U0, V1 = U1*.
3. В матрицу ***V*** добавляется троичный вектор, образованный парой *V0* и *V1*.
4. Конец.

Процедура склеивания троичных векторов

Строки матрицы ***U*** полагается непомеченными. Рассматриваться будут только непомеченные строки, а помеченные – пропускаться.

1. Из матрицы ***U*** выбирается очередная (выбор начинается с первой) строка ***u*** и соответствующая ей строка s из матрицы ***S***.
2. Если строка ***u*** – последняя в матрице ***U***, то переход на п. 4.
3. Иначе из матрицы ***U*** выбирается строка ***w*** (выбор начинается со строки, следующей за строкой ***u***) и соответствующая ей строка ***g*** из матрицы ***S***.
   Если все строки матрицы ***U*** просмотрены, то переход на п. 1.
   Если не все строки матрицы ***U*** просмотрены, то если строки ***s*** и ***g*** не совпадают, то переход на п.3.
   Если строки ***s*** и ***g*** совпадают, то если строки ***u*** и ***w*** не являются соседними, то переход на п.3.
   Иначе – строки ***u*** и ***w*** склеиваются, и результат заменяет строку ***u*** в матрице ***U***. Строка ***w*** в матрице ***U*** помечается. Переход на п. 1.
4. Конец.

IV. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Предложенный алгоритм реализован на языке программирования С++ с использованием библиотеки классов «булев вектор» и «троичный вектор». Для проверки эффективности разработанной программы был проведен вычислительный эксперимент.

Примеры матричных SF-описаний систем полностью определенных булевых функций были взяты из набора промышленных тестовых примеров, входящих в библиотеку примеров Berkeley PLA TestSet [5].

Исследовались два алгоритма ортогонализации – предложенный в [4] и представленный в настоящей работе, обозначаемые далее OLD и NEW соответственно.

Результаты эксперимента представлены в таблице 1, где *n* – число переменных, *m* – число функций, *k* – число элементарных конъюнкций исходной системы ДНФ булевых функций. Жирным шрифтом выделены лучшие решения.

Всего для каждого алгоритма рассматривалось три варианта ортогонализации. При первом производилось предварительное упорядочивание строк матриц ***U*** и ***S*** по возрастанию числа литералов в строках матрицы ***U***, при втором – по убыванию, а в третьем – упорядочивание не производилось.

В ходе вычислений замерялись следующие параметры:

C – число выполнений цикла из пунктов 1 – 4 описанного алгоритма;

M – максимальное число строк в матрице W, полученных при работе алгоритма;

R – суммарное число произведенных разложений;

S – суммарное число произведенных склеиваний;

P – суммарное число произведенных поглощений.

K – число элементарных конъюнкций в ортогонализованной системе.

В качестве иллюстрации сложности решаемых в ходе ортогонализации задач отметим результат ортогонализации для примера *intb* с параметрами: *n* = 15, *m* = 7, *k* = 664. За 10 часов работы программы была получена ДНФ с 7987конъюнкциями. При этом за 25 выполнений цикла было произведено 31707 разложений, 117599 склеиваний, 143844 поглощений, а число конъюнкций на пике вычислений достигло 22855.

V. Заключение

Предложенная в настоящей работе модификация представленного ранее алгоритма ортогонализации показала заметное преимущество в качестве получаемых решений над старым вариантом. Эксперимент показал, что для исследованного множества примеров использование новой программы во всех случаях обеспечивало нахождение лучшего решения. При этом преимущество нового алгоритма над старым достигало 80%.

Таблица 1

Результаты ортогонализации

№	Имя	*n*	*m*	*k*		По возрастанию		По убыванию		Без сортировки	
						OLD	NEW	OLD	NEW	OLD	NEW
1	TIAL	14	8	640	*C*	23	23	25	25	24	24
					M	22 011	22 011	26 686	26 686	29 403	29 403
					R	85 118	85 118	94 729	94 729	96 449	96 449
					S	105 760	105 760	120 047	120 047	124 603	124 603
					P	16 474	16 474	17 766	17 766	18 368	18 368
					K	9 218	**5 937**	10 385	6 405	10 584	6 461
2	B9	16	5	123	*C*	21	21	21	21	25	25
					M	12 142	12 142	9 813	9 813	12 956	12 956
					R	36 838	36 838	27 304	27 304	44 505	44 505
					S	40 604	40 604	30 126	30 126	49 682	49 682
					P	2 859	2 859	2 659	2 659	3 127	3 127
					K	11 954	6 319	9 793	**5 334**	12 703	6 763
3	Mp2d	14	14	123	*C*	15	15	14	14	20	20
					M	801	801	722	722	693	693
					R	2 707	2 707	2 083	2 083	2 553	2 553
					S	3 911	3 911	2 663	2 663	2 963	2 963
					P	209	209	199	199	202	202
					K	525	334	534	**328**	561	339
4	X6dn	39	5	121	*C*	12	12	13	13	13	13
					M	302	302	328	328	346	346
					R	384	384	409	409	446	446
					S	436	436	456	456	478	478
					P	206	206	176	176	227	227
					K	268	**219**	287	239	305	233
5	In2	19	10	137	*C*	15	19	14	12	16	18
					M	1 165	808	1 727	1 345	2 212	1 829
					R	1 711	1 220	2 827	2 177	3 502	3 094
					S	2 496	1 384	4 214	2 633	5 360	3 693
					P	429	645	523	785	544	1 005
					K	955	**448**	1 316	627	1 772	879
6	intb	15	7	664	*C*	25	25	25	24	25	29
					M	31 707	31 707	31 707	32 496	31 707	34 451
					R	117 599	117 599	117 599	118 077	117 599	107 197
					S	143 844	143 844	143 844	149 916	143 844	135 809
					P	22 855	22 855	22 855	21 605	22 855	22 278
					K	8 875	**7 281**	7 987	7 703	8 554	7 416

Литература

[1] Поттосин, Ю.В., Шестаков Е.А. Ортогонализация системы полностью определенных булевых функций / Ю.В. Поттосин, Е.А. Шестаков / Логическое проектирование, Вып.5. – Минск: Институт технической Кибернетики НАН Беларуси, 2000 г. – С. 107–115.

[2] Закревский, А.Д. Основы логического проектирования. В двух книгах. Книга 1. Комбинаторные алгоритмы дискретной математики / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – Минск: ОИПИ НАН Беларуси, 2004. – 226 с.

[3] Закревский, А.Д. Основы логического проектирования. В двух книгах. Книга 2. Оптимизация в булевом пространстве / А.Д.Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – Минск: ОИПИ НАН Беларуси, 2004. – 240 с.

[4] Кардаш, С. Н. Ортогонализация системы ДНФ булевых функций / С. Н. Кардаш // Информационные технологии и системы 2020 (ИТС 2020) – Information Tehnologies and Systems 2020 (ITS 2020): материалы междунар. науч. конф., (Республика Беларусь, Минск, 18 ноября 2020 года) редкол.: Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2020. – С. 41–42.

[5] Berkeley PLA test set [Electronic resource]. Mode of access: http://www1.cs.columbia.edu/ ~cs6861/ sis/espresso-examples/. Date of access: 9.12.2015.

Ortogonalization of the DNF System of Boolean Function

S.N. Kardash

United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk,

kardash77@gmail.com

***Abstract* — To solve many problems of synthesis, diagnostics and analysis of the reliability of technical systems, the representation of Boolean functions in the form of disjunctive normal forms (DNF) is used. It is often useful to have such DNFs in which all elementary conjunctions included in them are mutually orthogonal. To obtain such DNFs, it is necessary to orthogonalize the original DNF systems. In this paper, we present a new algorithm for solving the orthogonalization problem. The development of a computer program that solves the problem of orthogonalization of the DNF system is reported. The results of an experimental study are presented that confirm the effectiveness of the developed algorithm.**

***Keywords* — Boolean function, disjunctive normal form, orthogonality of elementary conjunctions.**

REFERENCES

[1] Pottosin, YU.W., Shestakow E.A. Ortogonalizacija systemy polnost’ju opredelennych bulewych funkciy (Orthogonalization of a system of fully defined Boolean functions) / YU.W. Pottosin, E.A. Shestakow / Logitcheskoe proektirowanie, Wyp.5. – Minsk: Institut technicheskoy Kibernetiki HNAN Belarusi, 2000 g. – S. 107–115.

[2] Zakrewsky, A.D. Osnowy logicheskogo proektiriwanija. Kniga 1. Kombinatornyje algoritmy diskretnoy matematiki (Combinatorial algorithms for discrete mathematics) / A.D. Zakrewsky, YU.W. Pottosin, L.D. Sheremisinowa. – Minsk: OIPI NAN Belarusi, 2004. – 226 s.

[3] Zakrewsky, A.D. Osnowy logicheskogo proektiriwanija. Kniga 2. Optimizacija w bulewom prostranstwe (Boolean Space Optimization) / A.D. Zakrewsky, YU.W. Pottosin, L.D. Sheremisinowa. – Minsk: OIPI NAN Belarusi, 2004. – 240 s.

[4] Kardash S.N. Ortogonalizacija systemy DNF bulewyh funkcij (Orthogonalization of a DNF System of Boolean Functions) / S. N. Kardash // Informacionnyje tehnologii i systemy 2020 (ITS 2020) – Information Tehnologies and Systems 2020 (ITS 2020): materialy megdunar. nauch. konf., (Respublika Belarus, Minsk, 18 nojabr’ja 2020 g.) redkol.: L. YU. Shilin [i dr.]. – Minsk: BGUIR, 2020. – S. 41–42.

[5] Berkeley PLA test set [Electronic resource]. Mode of access: http://www1.cs.columbia.edu/ ~cs6861/ sis/espresso-examples/. Date of access: 9.12.2015. URL: http://www.site.com.