

МАТЕРИАЛЫ
II Всероссийской молодежной
научной конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 16–17 мая 2014 г.

Под общей редакцией
кандидата технических наук И.С. Шмырина

Томск
Издательский Дом Томского государственного университета
2014

щения диполей на внешней вспомогательной поверхности $N_i = 260$, число разбиений осевого тока $N = 30$, число точек коллокации $L = 120$.

Заключение

Таким образом, в данной работе представлены алгоритм и программа расчета компонент электромагнитного поля, рассеянного тонким диэлектрическим цилиндром. Исследована зависимость сечения обратного рассеяния цилиндра от угла падения плоской волны относительно оси цилиндра. Установлено, что сечение обратного рассеяния монотонно возрастает с возрастанием угла падения (в интервале от 0° до 90°).

ЛИТЕРАТУРА

1. *Richmond J. H.* Digital computer solutions of the rigorous equations for scattering problems // Proc. IEEE. – 1965. – V. 53, № 8. – P. 796–804.
2. *Дмитренко А. Г., Гольцварт Е. П.* Решение задачи электромагнитного рассеяния на тонком диэлектрическом цилиндре методом вспомогательных источников // Радиотехника и электроника. – 2011. – Т. 56, № 5. – С. 600–607.

ПОСТРОЕНИЕ МАКСИМАЛЬНО СОВМЕСТИМЫХ ПОДМНОЖЕСТВ ТРОИЧНЫХ ВЕКТОРОВ С ПОМОЩЬЮ ДЕРЕВА РАЗЛОЖЕНИЙ

Т. П. Тарновская

Томский государственный университет
E-mail: tarnovskayat@mail.ru

Введение

Важным этапом проектирования является этап тестирования. Оно может быть внешним и внутренним. [2] Для проведения тестирования используется проверяющий тест, обеспечивающий полноту покрытия множества рассматриваемых неисправностей.

Сжатие тестов может быть динамическим или статическим.[4] Динамические методы выполняют сжатие одновременно с процессом генерации тестов и часто требуют модификации генератора тестов. Статическое сжатие используется после процесса генерации тестов.

В данной работе рассматривается статический метод сжатия тестовой последовательности, который реализуется с помощью дерева разложений. Он относится к категории алгоритмов, основанных на модификации тестовых векторов, основной операцией которых является совмещение. К этой же категории относится метод, предложенный в работе: «Модификация метода сокращения тестовой последовательности, основанного на построении дерева разложений», недостатком которого является использование «жадного» алгоритма на этапе формирования максимально совместимых подмножеств. Использование «жадного» алгоритма приводит к тому, что близость решения к точному решению зависит от места расположения векторов в исходной последовательности. Этот алгоритм не во всех случаях позволяет рационально распределить вектора по совместимым подмножествам, что способствует увеличению числа совместимых подмножеств, то есть увеличению длины решения. Чтобы избежать этого и приблизить решение к точному решению в данной работе предлагается алгоритм построения максимально совместимых подмножеств на основе разложения подмножеств ДС и анализа матриц смежности.

1. Постановка задачи

В данной работе исходная тестовая последовательность представляется в виде множества троичных векторов.

Троичный вектор – это вектор, компоненты которого могут принимать значения 0, 1, «-». Символ «-» – это неопределенная компонента вектора, которая может принимать значения как 0, так и 1.

Говорят, что вектор w подходит совместимому подмножеству W_i или вектор может быть добавлен в совместимое подмножество, если вектор, являющийся покрытием совместимого подмножества, совместим с w .

Множество совместимых векторов W_i из тестового набора W называется максимальным, если в него больше нельзя добавить ни один вектор из W .

Вес столбца тестовой последовательности определяется количеством определенных компонент в этом столбце.

Пусть дана проверяющая тестовая последовательность $W = \{w_1, w_2, \dots, w_n\}$ в виде троичных векторов. Имеется множество неисправностей $F = \{f_1, f_2, \dots, f_n\}$. Связь между W и F заключается в том, что вектора из W обнаруживают неисправности из F . Основной задачей является сокращение длины последовательности W , но такое сокращение, чтобы ни один из векторов w_1, w_2, \dots, w_n не был утрачен. Это условие является необходимым, так как каждый вектор w_i отвечает за обнаружение соответствующей неисправности f_i .

Так как исходная тестовая последовательность представляется в виде множества троичных векторов, то задачу будем решать с помощью совмещения векторов последовательности W . Операция совмещения применяется к троичным векторам по следующему правилу:

компоненты первого вектора	0 0 1 1 - - -
компоненты второго вектора	0 - 1 - 0 1 -
компоненты результирующего вектора	0 0 1 1 0 1 -

Операция совмещения для решения данной задачи выбрана не случайно, а по причине того, что она позволяет сокращать длину теста, не утрачивая значимые компоненты (0 или 1) векторов w_i , позволяющие обнаружить неисправности.

Метод сокращения тестовой последовательности, основанный на построении дерева разложений (ДР)

Алгоритм

1. Построение ДР из данной тестовой последовательности W .
2. Формирования максимально совместимых подмножеств.

Построение дерева разложений с учетом неопределенных компонент

Алгоритм

Вход: Тестовая последовательность в виде множества троичных векторов $W = \{w_1, w_2, \dots, w_n\}$.

Выход: ДР степени два или три.

Корню u_0 ДР сопоставляется исходная тестовая последовательность W .

Пока в ДР существует вершина u_i , которой сопоставляется несовместимое подмножество W_i выполняется его разложение на подмножества по правилу: Выбирается компонента разложения s_i , соответствующая ортогональному столбцу максимального веса. Вектора последовательности анализируются по компоненте разложения. Если компонента 0, то соответствующий вектор помещается в вершину u_{i+1} по ветви 0, если 1, то вектор помещается в вершину u_{i+1+1} по ветви 1, а если компонента «-», то соответствующий вектор отправляется в вершину по ветви «-» и помещается в множество DC_j (don't care).

Воспользуемся идеей предложенной в работе [6]. Если множества DC_j являются несовместимыми, то данные множества раскладываются по алгоритму построения ДР, подмножества векторов, соответствующие терминальным вершинам нового поддерева обозначаются W_i^{DC} .

После построения ДР в терминальных вершинах, помеченных 0 и 1, соответствуют совместимые подмножества W_i , из построения они являются попарно непересекающимися, поэтому они точно войдут в решение. Следовательно, количество векторов в решении будет не меньше, чем количество полученных вершин W_i в данном дереве разложений.

Свойства ДР

1. Если ДР, построенное из тестовой последовательности W имеет степень равную двум, то соответствующие конечным вершинам подмножества W_i векторов представляют все максимально совместимые подмножества относительно множества W . Применяя операцию совмещения для каждого W_i получим искомую тестовую последовательность.

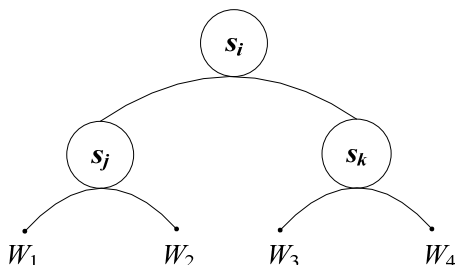


Рис. 1. ДР степени два. Все W_i являются максимально совместимыми

2. Совместимое подмножество W_i в ДР является максимально совместимым, если все узлы, находящиеся на пути от корня дерева до конечной вершины, которой соответствует подмножество W_i , имеют степень меньше трех.

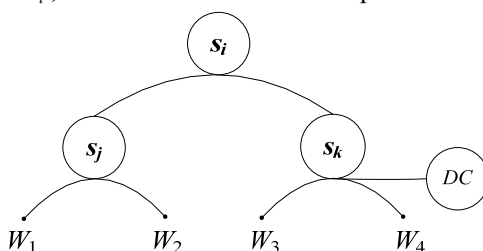


Рис. 2 ДР степени три. W_1, W_2 – максимально совместимые подмножества, а W_3, W_4 – совместимые

Если построенное ДР имеет степень три, то процесс поиска решения продолжается, т.к. при построении ДР некоторые вектора были отнесены к узлам по ветви « \leftarrow » и они не участвовали в формировании совместимых подмножеств W_i , но в решении должны участвовать все вектора исходной тестовой последовательности.

2. Формирование максимально совместимых подмножеств для ДР степени три

Вектора, которые при построении ДР не участвовали в формировании совместимых подмножеств должны быть отнесены к решению. Возможны различные исходы:

- вектор добавляется к совместимому подмножеству W_i ;

– вектор образует новое совместимое подмножество V_i ;

Из построения ДР следует, что вектора из подмножества W_j^{DC} соответствующего вершине DC_j имеют возможность быть добавленными в совместимые подмножества W_i , принадлежащие только терминальным вершинам поддерева с корнем, потомком которого является DC_j (рис. 3).

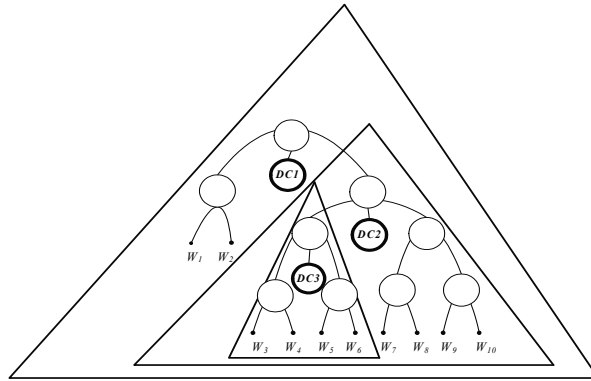


Рис. 3 Произвольное ДР, для каждого DC обозначены поддеревья, концевым вершинам которых могут быть присоединены вектора из соответствующего DC

Чтобы добыть дополнительную информацию о совместимости добавляемых векторов друг с другом выполняется следующая процедура.

Выполняется обход ДР, и просматриваются подмножества векторов W_i^{DC} , сопоставляемые узлам DC_i .

Если подмножество W_i^{DC} является совместимым, то обход продолжается, иначе для подмножества W_i^{DC} выполняется алгоритм построения ДР.

Для формирования максимально совместимых подмножеств используются матрицы совместимости для рассматриваемых вершин.

Матрицы заполняются по одному правилу. В ячейку ij ставится единица, если i -й интервал совмещается с j -ым интервалом, иначе ноль.

Строятся две основные матрицы совместимости T_1 и T_2 , для терминальных вершин ДР W_i и вспомогательные локальные матрицы совместимостей, для локального дерева декомпозиции построенного для DC_i , число которых зависит от конкретного примера.

Матрица T_1 строится для поддерева P_0 , расположенного по ветви 0 от корня ДР, T_2 – для поддерева P_1 , расположенного по ветви 1.

Суть в том, что решение находится с помощью основных матриц совместимостей, прибегая к анализу содержимого строк и столбцов. Например, если строка состоит из нулей, то соответствующий этой строке интервал отправляется в решение, а если в строке содержится одна единица, то результат совмещения векторов, на пересечении которых находится данная единица, помещается в решение. В других случаях над матрицами выполняются операции, результат которых позволяет получить решение.

Для случая, когда поддеревья DC_i являются степени три, в дополнение используется анализ локальных матриц.

Заключение

В данной работе рассматривалась задача сокращения длины проверяющей тестовой последовательности. Решение задачи было предложено в работе: «Модификация метода сокращения тестовой последовательности, основанного на построении дерева разложений». Сокращение выполнялось следующими этапами: построение дерева разложения; формирование максимально совместимых подмножеств, используя «жадный» алгоритм распределения векторов по подмножествам. В некоторых случаях использование «жадного» алгоритма не гарантирует построения решения близкого к точному решению, так как качество решения при использовании этого алгоритма зависит от порядка добавления совместимых интервалов.

В данной работе на основе анализа свойств дерева разложений был предложен метод, который позволяет приблизить решение к точному решению. Это достигается с помощью использования дополнительного разложения множеств DC_i и с помощью анализа матриц совместимостей.

ЛИТЕРАТУРА

1. *Andreeva V.* Test set compaction procedure for combinational circuits based on decomposition tree Proc., East-West Design&Test international Symposium Sevastopol, Ukraine, 2011. pp. 251–254.
2. *Уэйкерли Дж. Ф.* Проектирование цифровых устройств. – М. : Постмаркет, 2002. – Т. 2. – 1041 с.
3. *Miyase K., Kajihara S., Reddy M.* A Method of Static Test Compaction Based on Don't Care Identification // Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications (DELTA.02). – 2002. – P. 392–395.
4. *Hamzaoglu I., Patel J. H.* Test Set Compaction Algorithms for Combinational Circuits // ICCAD–1998. – San Jose, 1998. – P. 283–288.
5. *Андреева В. В.* Обеспечение сокращения аппаратурных затрат в схемах логического управления со свойствами самопроверяемости, самотестируемости и отказоустойчивости : автореф. дис. ... канд. техн. наук. – Томск, 2009. – 22 с.
6. *Andreeva V., Sorudeykin K.* A Research of Heuristic Optimization Approaches to the Test Set Compaction Procedure Based On a Decomposition Tree for Combinational Circuits // Proceeding of IEEE East–West Design & Test Symposium 2012, pp. 382–387.
7. *Hochbaum S.* An optimal test compression procedure for combinational circuits // IEEE Trans. on CAD of Integrated Circuits and Systems 1996, 15(10) – P. 1294–1299.
8. *Kanad Basu, Prabhat Mishra.* Test Data Compression Using Efficient Bitmask and Dictionary Selection Methods // IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. 2010. Vol. 18, № 9. SEPTEMBER. – P. 1277–1286.
9. *Istvan Juhos, Attila Troth, Jano I. van Hemert.* Binary Merge Model Representation of the Graph Colouring Problem / Eds. by J. Gottlieb and G.R. Raidl. EvoCOP 2004, LNCS 3004, pp. 124–134.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ JAVA-ПРИЛОЖЕНИЙ

В. В. Томков, А. А. Мирютов

Томский государственный университет

E-mail: isirbis@gmail.com

Введение

Существующие на сегодня методы тестирования программного обеспечения не позволяют однозначно и полностью установить корректность функционирования анализируемой программы, но они могут доказать, что отсутствуют дефекты с точки зрения используемого кода.

1. Анализ автоматизации модульного тестирования

Как правило, автоматизируют модульное тестирование с помощью готовых программных инструментов, например набор XUnit [1], которых сейчас достаточно для каждого языка программирования, а некоторые языки, например D [2] и Cobra [3], имеют поддержку модульного тестирования на уровне синтаксиса.