МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ОПТИКИ АТМОСФЕРЫ СО РАН им. В.Е. ЗУЕВА

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАНИИ СЛОЖНЫХ СТРУКТУР

МАТЕРИАЛЫ
ДВЕНАДЦАТОЙ КОНФЕРЕНЦИИ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ
4–8 июня 2018 г.

Мероприятие проведено при финансовой поддержке
Российского фонда фундаментальных исследований (проект № 18-07-20033)

Томск
Издательский Дом Томского государственного университета
2018

временем покупки; проверка уведомлений об ошибке; пользователь с дополнительной работой; индивидуальный предприниматель; страница с результатами покупки; рассылка информации на почту. Далее был создан тестовый набор регрессионного тестирования, который содержал проверки заявленной функциональности (11 тестов). В результате поиска кратчайшего покрытия матрицы $\boldsymbol{R}$ оказалось, что первоначальный тестовый набор можно сократить на 14,3% (2 теста) без потери процента покрытия функционала. Аналогично данный метод был апробирован для другого проекта – «Сайта страхования недвижимости». По реализуемой задаче оба приложения похожи и имеют небольшое количество отличительных черт. Поэтому и тестовый набор во многом одинаков. Во втором случае удалось добиться уменьшения первоначального тестового на 21,4%.

Вторая задача относится к поиску минимального диагностического теста, позволяющего проводить классификации дефектов в программном продукте. Это полезно при автоматизации тестирования больших проектов, когда тестировщики разделяются на определенные группы. Рассмотрим множества $D = \{D_1, D_2, \ldots, D_n\}$ и $B = \{b_1, b_2, \ldots, b_n\}$. Множество D является множеством возможных дефектов, множество B – множеством внешних признаков (симптомов), которые возникают вследствие обнаружения дефекта. Симптом – свойство дефекта, позволяющее классифицировать дефекты по их типичному проявлению. Строится булева диагностическая матрица $\boldsymbol{C}$, которая показывает, какими признаками характеризуется тот или иной дефект. Затем строится матрица различий $\boldsymbol{R}$, строки которой соответствуют парам строк диагностической матрицы и показывают, какими компонентами отличаются строки в этих парах, при этом будем использовать покомпонентную операцию сложения по модулю два, выполняя ее над всеми парами строк диагностической матрицы.

Пусть {1, 2, 3, 4, 5, 6, 7, 8, 9} – множество симптомов. Дефект D_1характеризуется признаками 1,8 и отрицает остальные. Дефект D_2 характеризуется признаками 2, 6, 7, 8 и отрицает остальные. Тогда первой строке матрицы различий будет соответствовать вектор D_1D_2=(1 1 0 0 0 1 1 0 0).

Для полученной матрицы различий находится кратчайшее столбцовое покрытие. Множество признаков, соответствующее столбцам из найденного покрытия, будет одним их решений задачи поиска минимального безусловного диагностического тест. При этом каждый дефект однозначно определен соответствующей строкой полученной подматрицы. Это позволяет наиболее рациональным образом классифицировать дефекты на момент их возникновения по данным признакам Данная задача может оказаться полезной при организации автоматизации тестирования больших проектов, когда необходимо в разделении специалистов по тестированию на определенные группы, появляется задача о классификации дефектов.

Литература

1. *Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д.* Логические основы проектирования дискретных устройств. М. : ФИЗМАТЛИТ, 2007. 592 с.

О ПОВЫШЕНИИ ЗАЩИЩЁННОСТИ ЛОГИЧЕСКИХ СХЕМ ОТ ВНЕДРЕНИЯ ВРЕДОНОСНЫХ ПОДСХЕМ*

В.А. Провкин, А.Ю. Матросова

Национальный исследовательский Томский государственный университет, г. Томск, Россия
prowkan@mail.ru, mau11@yandex.ru

При синтезе современных интегральных схем разработчики всё чаще прибегают к услугам сторонних фирм для реализации тех или иных компонент системы с целью снижения её стоимости. В компонентах, изготовленных сторонними фирмами, могут быть спрятаны вредоносные подсхемы (Trojan circuits) с целью разрушения системы или извлечения из неё конфиденциальной информации [1]. Вредоносные подсхемы обычно действуют в ситуациях, которые возникают в работающей системе чрезвычайно редко, поэтому они трудно обнаружимы как в процессе верификации схемы, так и в процессе её тестирования. Вредоносные подсхемы необходимо обнаруживать и, по возможности, нейтрализовать их действие.

Дана комбинационная схема C, состоящая из вентилей. В ней рассматриваются линии с низкой наблюдаемостью, исходящие из точек ветвления. Из каждой точки ветвления может выходить только одна такая линия. Предполагается, что линии с низкой наблюдаемостью («подозрительные» линии») могут быть использованы для включения в них Trojan circuits (TCs). Такие линии предлагается закрывать программируемыми блоками памяти look up tables (LUTs), число входов которых на единицу больше максимального числа входов вентилей в исходной схеме[2], чтобы исключить возможность включения в них TCs.

Для этого выполняются следующие шаги: вместо элемента, в который ведёт «подозрительная» линия связи, ставится программируемый блок, сама линия связи удаляется, а к программируемому блоку подводятся

* Исследование выполнено за счет гранта Российского научного фонда (проект № 14-19-00218).

линии, которые получаются дублированием линий связи, идущих в элемент, из которого исходит «подозрительная» линия. Опишем способ программирования LUT. Пусть элемент, в который ведёт «подозрительная» линия, реализует функцию $f(x_1, x_2, \ldots, x_n)$, а элемент, из которого исходит «подозрительная» линия, реализует функцию $g(y_1, y_2, \ldots, y_l)$. Пусть «подозрительная» линия соединена с входом x_i элемента, реализующего функцию f, тогда в LUT программируется следующая функция: $f(x_1, x_2, \ldots, x_{i-1}, g(y_1, y_2, \ldots, y_l), x_{i+1}, \ldots, x_n)$. Новая функция зависит от $(l + n - 1)$ переменной. Если m – максимальное число входов вентилей в схеме, то для обеспечения покрытия элемента программируемым блоком должно выполняться следующее условие:

$$l + n - 1 \le m + 1.$$

Рассмотрим пример, иллюстрирующий работу алгоритма (рис. 1).

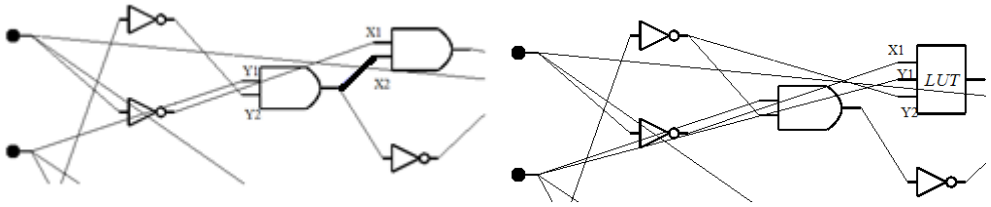

Рис 1. Покрытие LUT'ом двухвходового элемента

Здесь «подозрительная» линия исходит из элемента И. элемент, к которому ведёт эта линия, заменяется программируемым блоком, к нему подводятся все линии (кроме исключаемой), который вели в замещаемый элемент, а также линии, которые вели в элемент И (здесь новые линии получаются путём дублирования). В блоке памяти для рассматриваемого примера программируется функция:

$$f = f(x_1, x_2) = x_1 \wedge x_2 = x_1 \wedge g(y_1, y_2) = x_1 \wedge (y_1 \wedge y_2).$$

Литература

1. *Матросова А.Ю., Останин С.А., Николаева Е.А.* Синтез частично программируемых схем, ориентированный на маскирование вредоносных подсхем (Trojan Circuits) // Труды Института системного программирования РАН. 2017. Т. 29, № 5. С. 61–74.
2. *Jo S., Matsumoto T., Fujita M.* SAT-based automatic rectification and debugging of combinational circuits with LUT insertions // Proc. Of IEEE Asian Test Symposium. 2012. P. 19–24.

К ПОСТРОЕНИЮ ПАРАЛЛЕЛЬНОЙ КОМПОЗИЦИИ РАСШИРЕННЫХ АВТОМАТОВ*

Е.В. Дарусенкова, С.А. Прокопенко, Н.В. Шабалдина

Национальный исследовательский Томский государственный университет, Томск, Россия
darusenkova@gmail.com, s.prokopenko@sibmail.com, nataliamailbox@mail.ru

В работе рассматривается задача описания взаимодействия компонент веб-сервисов, которые, помимо общения друг с другом, также взаимодействуют и с внешней средой (рисунок 1). Адекватной математической моделью для описания веб-сервисов является расширенный автомат [1], и взаимодействие компонент также хотелось бы описать расширенным автоматом. В случаях, когда области значений входных/выходных параметров и контекстных переменных конечны, мы можем полностью промоделировать поведение расширенного автомата и перейти к хорошо изученной модели конечного автомата [2]. Однако, если данные области значений слишком велики или бесконечны, то построить конечный автомат не представляется возможным. В этом случае можно промоделировать поведение расширенного автомата на входных последовательностях длины не больше заданного числа l (так называемый l-эквивалент, который также является конечным автоматом) [3].

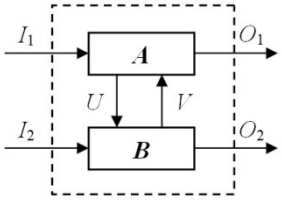

Рис. 1. Параллельная композиция расширенных автоматов

* Работа выполнена при поддержке Российского научного фонда, проект № 16-49-03012.