ОТЗЫВ

официального оппонента доктора технических наук,
профессора Шалыто Анатолия Абрамовича
на диссертацию Ермакова Антона Дмитриевича
«Автоматные методы и алгоритмы синтеза тестов для программного обеспечения с использованием подходов формальной верификации»,

представленную на соискание ученой степени
кандидата технических наук по специальности
05.13.01 – Системный анализ, управление и обработка информации

Актуальность работы. Задача верификации и тестирования программного обеспечения (ПО) была и остается актуальной задачей при проектировании и эксплуатации программных или программно-аппаратных систем. Качество используемых тестов в значительной степени определяет корректность последующей работы системы, и практически невозможно построить качественные тесты для современных систем без применения формальных моделей. Одной из широко используемых моделей при построении качественных тестов для дискретных и гибридных, в том числе, программных систем, являются модели с конечным числом состояний/переходов, известные как автоматные модели. Большинство методов синтеза тестов с гарантированной полнотой разработаны для детерминированных автоматов, тогда как в настоящее время достаточно часто используются недетерминированные спецификации, которые появляются по различным причинам, таким как опциональность, неполные наблюдаемость и управляемость, уровень абстракции и т. п. Более того, классический конечный автомат для реальных программных систем оказывается слишком большим, и поэтому тесты с гарантированной полнотой желательно строить для неклассических автоматных моделей, таких, например, как расширенные или магазинные автоматы, описание которых достаточно близко к программным реализациям в языках высокого уровня. Соответственно, тесты с использованием таких моделей часто используют различные критерии покрытия. Однако для расширенных автоматов известно, что тесты, основанные на покрытии путей, переменных, условий и т. д., не обнаруживают большое число функциональных ошибок в ПО, поведение которого описано соответствующим расширенным автоматом. Поэтому необходимы модели неисправностей, более адекватно описывающие функциональные ошибки, появляющиеся в проектируемом ПО. При этом, как обычно, при обнаружении ошибки хотелось бы локализовать ее хотя бы с точностью до множества «подозрительных инструкций».

Следует также отметить, что в настоящее время активно развивается тестирование ПО относительно нефункциональных требований, например, таких, как проверка свойств его безопасности. Проверка безопасности ПО осуществляется как статическими, так и динамическими методами, и интерес представляет создание программных комплексов, позволяющих оценить безопасность, проектируемого ПО на различных языках программирования, в том числе, на языках высокого уровня, таких как язык *C*. При этом для повышения качества проверки ПО методы синтеза тестов целесообразно разрабатывать с учетом подходов формальной верификации.

Исходя из изложенного, можно утверждать, что диссертации А.Д. Ермакова «Автоматные методы и алгоритмы синтеза тестов для программного обеспечения с использованием подходов формальной верификации» весьма актуальна.

Краткий анализ диссертации. Диссертация состоит из введения, четырех глав, заключения и списка использованной литературы.

Во введении определяются цели и задачи диссертации и формулируются основные положения, выносимые на защиту.

В первой главе приводятся определения и понятия, которые используются далее в работе, относящиеся к конечным и расширенным автоматам, а также к мутационному тестированию. Кроме того, первая глава содержит краткий обзор литературы по синтезу тестов для программных систем. Показано, что поставленные в работе задачи актуальны и соответствуют современным тенденциям в области тестирования и верификации программного обеспечения.

Вторая глава диссертации посвящена описанию разработанного автором метода повышения полноты тестов, построенных по расширенному автомату с использованием мутационного тестирования. Автор отмечает, что тесты, построенные с использованием расширенных автоматов на основе различных покрытий (условий, путей, операторов и т. д.) достаточно часто оставляют в программных реализациях не обнаруженными функциональные ошибки. Отмечается, что процесс построения различающих последовательностей по двум программам (исходной и мутированной) является достаточно сложной задачей. Предложенные автором метод и алгоритм позволяют повысить качество теста путем добавления к нему различающих последовательностей, построенных по автомату-спецификации и автомату-мутанту. Для построения множества неисправности автор реализует автомат-спецификацию в виде разработанного шаблона на языке *Java* и далее генерирует множество компилируемых модификаций программы с

помощью генератора мутантов *muJava*. Такие модификации, не обнаруженные первоначальным тестом, достаточно просто отображаются в мутант автомата-спецификации. Это дает возможность нахождения различающих последовательностей на основе сравнения двух автоматных моделей. Последнее является более простой задачей по сравнению с построением различающей последовательности для двух программных реализаций. Соответствующие различающие последовательности добавляются в первоначальный тест, тем самым повышая его качество относительно обнаружения функциональных ошибок в проектируемом ПО. Проведенные автором эксперименты иллюстрируют, что в ряде случаев такой подход к повышению полноты тестов является достаточно эффективным. В этой же главе автором предлагается достаточно простой алгоритм для обнаружений неисправной компоненты в автоматной сети, эффективность которого иллюстрируется на примере ПО, содержащего несколько методов сортировки.

В третьей главе автором предлагается метод построения адаптивной проверяющей последовательности для случая, когда спецификация представлена недетерминированным конечным полностью определенным автоматом, а проверяемая реализация описывается детерминированным автоматом. Иными словами, предполагается, что недетерминизм в спецификации является, в первую очередь, следствием опциональности – в ПО могут быть реализованы различные опции. Одним из примеров могут служить *RFC*-спецификации телекоммуникационных протоколов. В этом случае поведение тестируемой реализации должно содержаться в поведении спецификации – проверяющий тест строится относительно редукции. При этом конформная реализация должна «делать» только то, что предписано спецификацией. Более того, автор отмечает, что в некоторых случаях надежный сигнал СБРОС, переводящий автомат в начальное состояние является достаточно «дорогим» (например, включение/выключение компьютера), и в такой ситуации имеется смысл строить одну последовательность, позволяющую выяснить, является ли проверяемая реализация редукцией спецификации.

Глава разделена на три части по уровню требований, предъявляемых к спецификации.

В первом разделе этой главы автором предлагается алгоритм построения адаптивной проверяющей последовательности, основанный на том, что автомат-спецификация обладает разделяющей последовательностью, позволяющей различить любые два состояния спецификации, и детерминированным сильно связным подавтоматом. Эти условия гарантируют, что только автомат, изоморфный некоторому подавтомату спецификации с тем же числом состояний, может быть редукцией автомата-

спецификации. Соответственно, в алгоритме автор предлагает строить проверяющую последовательность, которая содержит две части. В первой части устанавливается взаимно однозначное соответствие между состояниями спецификации и тестируемой реализации. Если такого соответствия не существует, то выдается вердикт, что реализация не является конформной спецификации. Во второй части проверяющей последовательности устанавливается взаимно однозначное соответствие между состояниями спецификации и тестируемой реализации.

Во втором разделе третьей главы на простых примерах автор иллюстрирует, как можно разделяющую последовательность заменить различающим тестовым примером и предлагает соответствующий алгоритм.

В третьем 3 этой главы требование о наличии сильно связного детерминированного подавтомата в спецификации ослабляется. При этом автор предлагает вместо детерминированной достижимости состояний рассматривать адаптивную достижимость. Несмотря на то, что последний раздел дает возможность строить проверяющую последовательность для большего числа спецификаций, автор отмечает, что первые два алгоритма могут быть использованы и для частичных спецификаций.

Четвертая глава посвящена методу проверки наличия уязвимостей в программах, написанных на языке *C/C++*. Проведен обзор известных программных продуктов поиска уязвимостей в программном коде. Над многими программными продуктами проведены компьютерные эксперименты, которые показали, что большинство из них не обнаруживают ошибки типа переполнения буфера и другие подобные ошибки, имеющие в ряде случаев большое значение для безопасности разрабатываемого ПО.

Автором разработан комплекс прикладных программ на основе верификатора *JavaPathFinder*, описание которого представлено в этой же главе. Эффективность данного программного комплекса продемонстрирована на основе компьютерных экспериментов.

Степень обоснованности научных положений, выводов и рекомендаций, сформулированных в диссертации, их достоверность, новизна.

В диссертации А.Д. Ермакова на основании выполненных им исследований разработан метод построения адаптивной проверяющей последовательности для недетерминированного конечного полностью определенного автомата относительно редукции. Также разработан ряд алгоритмов:

- поиска уязвимостей типа переполнения буфера в ПО на языке *C/C++*;
- повышения полноты тестов для ПО, построенных по модели расширенного автомата, с использованием мутантов, генерируемых программным инструментом *μJava*;
- локализации неисправной компоненты в многокомпонентной автоматной композиции на основе трассировки исполнения композицией проверяющего теста.

Результаты, выносимые на защиту, являются **новыми.** Это подтверждается соответствующими публикациями, как в журналах из Перечня ВАК, так и в трудах ведущих международных конференций. Все положения, формулируемые в диссертации, **доказываются** с использованием аппарата дискретной математики. **Эффективность** предложенных методов подтверждается компьютерными экспериментами.

Работа выполнена на хорошем математическом уровне, хорошо структурирована, написана понятным языком.

Предложенные методы и алгоритмы **могут быть использованы** при тестировании программного обеспечения, функциональные требования к которому описаны посредством расширенного автомата, в том числе, для тестирования программных реализаций телекоммуникационных протоколов, программ автоматизированного управления и т. д. Пакет программ для поиска уязвимостей в программном обеспечении на основе верификатора *JavaPathFinder* **может быть использован** при тестировании безопасности ПО. С помощью этого комплекса **был проверен** ряд производственных программ ОАО «ТомскНИПИнефть» на наличие уязвимостей типа «переполнение буфера». Разработанный автором метод синтеза тестов на основе расширенных автоматов с использованием мутационного тестирования позволил повысить качество функциональных тестов для штатного ПО.

Результаты работы могут быть также применены в учебном процессе, поскольку могут быть использованы в курсах лекций по теории автоматов, теории формальных моделей, верификации и тестирования.

Замечания по диссертации. Следует отметить, что диссертация не свободна от некоторых недостатков, некоторые из которых имеет смысл перечислить:

1) Результаты главы 2 по повышению полноты тестов на основе расширенных автоматов с использованием мутационного тестирования для шаблонной программной реализации представляются интересными. Однако автору, наверное, стоило в приложении привести пример не только плейера, но и для более «серьезной» системы.

2) Нам кажется, что обнаружение неисправной компоненты в автоматной сети требует дополнительных исследований. Вполне возможно, что впоследствии эти результаты окажутся очень интересными, но на данный момент они выглядят, скорее, как наблюдения, а не результаты исследования.

3) Автор утверждает, что первые два раздела главы 3 можно использовать для частичных автоматов, в то время как последний раздел на данный момент допустим только для полностью определенных автоматов. Не кажется однозначным принятое автором решение включить в диссертацию все три алгоритма. Однако, принимая во внимание, что диссертация является, в том числе и квалификационной работой, можно согласиться с его решением.

4) Интересно было бы увидеть результаты работы разработанного автором программного продукта по обнаружению уязвимостей не только для студенческих программ.

5) Несмотря на достаточно грамотное изложение материала, в работе отсутствуют некоторые знаки препинания и имеются опечатки.

Тем не менее, указанные недостатки не снижают научной и практической ценности диссертации.

По материалам диссертации **опубликовано** 12 работ, из них четыре статьи в журналах, включенных в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, а также три статьи в зарубежных изданиях, индексируемых *Web of Science* и *Scopus*. Эти результаты докладывались на ряде научных конференциях и семинарах, в том числе, международных. Материалы диссертации достаточно полно отражены в опубликованных работах.

Автореферат **полностью** соответствует содержанию диссертации.

Все результаты, выносимые на защиту, принадлежат соискателю.

Исходя из изложенного, можно утверждать, что диссертация А.Д. Ермакова является **научно-квалификационной работой**, в которой изложен новый научно обоснованный метод синтеза адаптивной проверяющей последовательности по модели неинициального недетерминированного автомата, предложены алгоритмы синтеза проверяющих тестов, повышающие полноту тестирования программных продуктов. Таким образом, в диссертации изложены новые научно обоснованные технические решения и разработки, имеющие существенное значение для развития подходов к

тестированию (встроенного) программного обеспечения, в том числе для тестирования программных реализаций телекоммуникационных протоколов, программ автоматизированного управления, встроенных программных продуктов, и т. .

Считаю, что диссертация А.Д. Ермакова соответствует всем требованиям, установленным Положением о присуждении ученых степеней к диссертациям на соискание ученой степени кандидата технических наук по специальности 05.13.01 – «Системный анализ, управление и обработка информации», а автор диссертации полностью заслуживает присуждения этой степени.

Официальный оппонент,
заведующий кафедрой технологий программирования федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»,
доктор технических наук (05.13.05 – Элементы и устройства вычислительной техники и систем управления), профессор

Анатолий Абрамович Шалыто

24 апреля 2017 г.

Сведения об организации:
Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»
197101, Санкт-Петербург, пр. Кронверский, д. 49;

7 (812) 232 97 04; od@mail.ifmo.ru; http://www.ifmo.ru

Подпись А.А. Шалыто заверяю

Начальник управления кадров О.В. Котусева