УТВЕРЖДАЮ
Директор Федерального государственного учреждения «Федеральный исследовательский центр «Информ управление» Российской ака

И.А. Соколов
«28» 01 2018 г.

ОТЗЫВ

ведущей организации на диссертацию Белеванцева Андрея Андреевича на тему «Многоуровневый статический анализ исходного кода для обеспечения качества программ», представленную на соискание ученой степени доктора физико-математических наук по специальности 05.13.11 (математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей)

Диссертационная работа Белеванцева А.А. посвящена проблеме построения методологии разработки статического анализа программ, используемого для поиска дефектов и уязвимостей в исходном коде в условиях постоянного применения анализаторов в цикле разработки программного обеспечения, в том числе в парадигме непрерывной разработки. **Актуальность** этой проблемы бесспорна и связана с современным уровнем развития программных систем, при котором их повсеместная доступность через сети, широкое применение библиотек с открытым исходным кодом делает необходимым снижение количества дефектов в программах всеми возможными способами. Быстрый рост размера программ не позволяет искать дефекты вручную, и

общепризнано, что одним из привлекательных подходов к автоматизации поиска является статический анализ исходного кода. Однако для адекватного ответа на описанные особенности программ требуется непрерывное комплексное развитие методов анализа.

В диссертации получены следующие **актуальные результаты**, обладающие **научной новизной**:

- новая методология проведения статического анализа исходного кода программ для поиска дефектов, которая состоит в комплексном наборе алгоритмов анализа и моделей программы и поддерживает анализ абстрактного синтаксического дерева программы, внутрипроцедурный анализ, межпроцедурный контекстно-чувствительный анализ, чувствительный к путям выполнения анализ, масштабируемый до программ в десятки миллионов строк исходного кода на языках Си, Си++, Java, C#;
- новые алгоритмы поиска конкретных дефектов в программе на основе разработанных методов, поддерживающие десятки популярных классов дефектов и уязвимостей. Среди поддерживаемых критических типов дефектов можно упомянуть дефекты разыменования некорректного указателя, дефекты управления памятью и ресурсами, использование неинициализированных переменных, дефекты переполнения буфера, дефекты многопоточных примитивов и др.;
- новая архитектура программной системы и ее программная реализация в наборе анализаторов Svace, которая содержит программное воплощение всех указанных методов анализа, а также обеспечивает их работу путем полностью автоматической подготовки всех необходимых входных данных и обработке и хранению результатов их работы;
- разработанные компоненты программной системы анализа, позволяющие единообразно управлять работой коллекцией анализаторов в рамках единой системы, в том числе запускать,

просматривать результаты, выполнять инкрементальный анализ измененной части программы.

Достоверность результатов, полученных в диссертационной работе, обеспечивается обоснованностью математических доказательств, согласованностью результатов с другими подходами, представленными в литературе, успешным функционированием реализованных программных средств, апробацией на конференциях различного уровня.

Теоретическая значимость работы вытекает из сделанного автором математически обоснованного развития области статического анализа программ, которое заключается в предложенной совокупности методов анализа над общей моделью памяти программы; подхода к увеличению точности анализа за счет использования концепции классов значений и параметризации результатов анализа процедур внешними значениями; поддержке уровней анализа различной глубины (от участков кода процедуры до всей программы).

Практическая значимость работы заключается в использовании программной системы Svace в качестве основного средства анализа в компании Samsung, в том числе для проверки всех изменений операционной системы Tizen, выполненных публично через систему непрерывной интеграции на сайте developer.tizen.org, а также в других российских компаниях. Компоненты программной системы используются в Институте системного программирования Российской академии наук для построения других инструментов статического анализа. Результаты диссертации могут быть интересны сертификационным центрам и компаниям, внедряющим жизненный цикл безопасной разработки программного обеспечения согласно ГОСТ Р 56939-2016. Разработанные алгоритмы используются и в учебном процессе при чтении лекций на факультете ВМК МГУ.

Диссертационная работа А.А. Белеванцева состоит из введения, пяти глав и заключения, изложена на 229 страницах.

Во **введении** обосновывается актуальность работы, приводятся цель и задачи исследования, формулируются его результаты, даются сведения об апробации.

Первая глава представляет собой аналитический обзор основных современных методов статического анализа, устройства инструментов анализа и их использования, а также перспективных направлений исследований. Разделы 1.1-1.3 посвящены методам, соответствующим предложенным автором уровням анализа – обходам абстрактного синтаксического дерева и внутрипроцедурному анализу, межпроцедурному анализу, чувствительному к путям выполнения. Важно отметить, что для разбора программ в таких анализаторах используются промышленные компиляторы с открытым исходным кодом. Среди межпроцедурных методов подходом, доставляющим масштабируемость для сверхбольших программ, является применение аннотаций функций, записывающих результаты анализа для некоторого контекста. В чувствительных к путям выполнения методах существенным является использование SMT-решателей. Раздел 1.4 представляет попытки формализовать определение ошибочной ситуации – участка кода, который требуется находить анализаторам. Часто используемым методом является поиск противоречий в собранной информации о потоке данных программы в предположении о существовании реальных входных данных, на которых эти противоречия реализуемы. Раздел 1.6 посвящен опыту применения промышленных статических анализаторов на практике, а разделы 1.5 и 1.7 – новым методам анализа, в частности, использованию подходов машинного обучения для ранжирования выданных предупреждений об ошибках и автоматическому исправлению некоторых ошибок.

Во **второй** главе представляется и математически обосновывается методология построения статического анализа, состоящая в применении совокупности методов анализа разных уровней сложности над различными представлениями программы, но на основе единой модели памяти программы. Раздел 2.1 представляет анализ абстрактного синтаксического дерева и таблицы символов программы с помощью обходов дерева или внутрипроцедурного анализа процедур программы. Предлагается классификация детекторов, выполняющих обход дерева, соответствующие алгоритмы обхода, и доказывается теорема 2.1 об их линейной сложности. Строится модель памяти на основе ячеек памяти и их основных свойств, которые могут быть вычислены с помощью отслеживания целочисленных операций и операций с указателями. Приводится основной алгоритм 2.2 построения модели памяти и доказываются его корректность и сложность (теоремы 2.2-2.3). Интересной в плане развития математической теории анализа представляется раздел 2.2, где вводится понятие классов значений – абстрактных значений, позволяющих отслеживать эквивалентность значений между присваиваниями и основными арифметическими операциями, а также приводится алгоритм построения аннотаций для межпроцедурного анализа, параметризующий результаты анализа внешними ячейками памяти и внешними по отношению к процедуре классами значений, доказывается его корректность (теорема 2.5). Раздел 2.3 развивает далее предлагаемые формализмы на чувствительные к путям выполнения анализы, предлагая отслеживать предикаты, при равенстве истине которых точно известны конкретные значения для классов значений и множества потенциальных целей указателей. Раздел 2.4 содержит принципы построения детекторов – алгоритмов поиска конкретных типов дефектов.

Третья глава заключает в себе описание программного средства Svace, начиная от предлагаемой архитектуры, обеспечивающей функционирование всей совокупности методов анализа. Раздел 3.1 посвящен методам организации контролируемой сборки программы, а раздел 3.2 – вопросам создания трансляторов на основе открытых компиляторных инфраструктур, которые могут уверенно использоваться в промышленности для разбора различных диалектов языков программирования. Наибольший интерес представляет раздел 3.3, где затронуты вопросы организации параллельного межпроцедурного анализа согласно алгоритмам главы 2 – алгоритм 3.1 представляет все необходимые входные данные. Рассматривается специфика обработки программ на языке Java и C#, в особенности алгоритмов девиртуализации, необходимых для построения возможно более полного графа вызовов. Важным свойством является поддержка удаленного и инкрементального анализа. Раздел 3.4 представляет компоненты хранения всех артефактов контролируемой сборки и анализа и показа результатов.

В **четвертой** главе описывается устройство разработанных для системы Svace детекторов – алгоритмов поиска конкретных типов дефектов. Нужно отметить, что реализовано более 200 детекторов, успешно используемых на практике. Глава структурирована по уровням анализа: детекторам, выполняющим обходы абстрактного синтаксического дерева, посвящен раздел 4.1 (рассмотрены детекторы всех поддерживаемых языков и всех предложенных в классификации раздела 2.1 типов); межпроцедурным детекторам – раздел 4.2, а чувствительным к путям детекторам – раздел 4.3. Выделение ядра анализа, обрабатывающего инструкции внутреннего представления и генерирующего события, на которые могут подписываться конкретные детекторы, позволяет детекторам легко выбирать нужные свойства

программы и вычислять их атрибуты. Рассматривается набор часто используемых атрибутов и их применение в детекторах. Важной частью анализатора являются детекторы переполнения буфера на основе символьного выполнения. Отдельно рассматриваются детекторы, собирающие и обрабатывающие статистическую информацию о программе и детекторы для программ на языке C#, т.к. они реализованы в отдельной компоненте анализа.

Пятая глава содержит экспериментальные результаты применения семейства анализаторов Svace к открытому исходному коду. Раздел 5.1 посвящен тестированию компонентов контролируемой сборки, которые демонстрируют замедление в единицы процентов. В разделе 5.2 приведены данные о времени сборки программ для анализатора и времени самого анализа, которые показывают скорость анализа в пределах 500-1500 строк в минуту. Раздел 5.3 демонстрирует качество реализованных детекторов, для которых процент истинных срабатываний колеблется между 50% и 80%, а в среднем по всем детекторам для Android 5 истинные срабатывания составляют около 70% для языков Си и Си++ и около 80% для Java.

В **заключении** приводятся основные результаты работы.

В диссертации могут быть отмечены следующие недостатки.

1. Стиль изложения диссертации зачастую привлекает слова и выражения, понятные лишь узким специалистам, без подробного их введения.

2. В обзоре диссертации недостаточно подробно описан инструмент Infer и сепарационная логика (раздел 1.7), которые являются одним из альтернативных способов организации статического анализа.

3. В модельном языке (рисунок 2.2 диссертации) желательно было бы рассмотреть все основные арифметические операции, а не только сложение и вычитание.

4. В теореме 2.5 (раздел 2.2.3 диссертации) сохранение полного дерева вычислений с участием внешних классов значений позволило бы обеспечить тот же порядок вычислений без привлечения ассоциативности передаточных функций.

5. В разделе 3.2.1 указано, что используются компиляторы на основе LLVM 3.4 и 3.8, тогда как в настоящий момент можно использовать и более новые версии LLVM.

Указанные замечания не влияют на общую высокую оценку диссертационной работы, которая в соответствии с требованиями Постановления Правительства Российской Федерации «О порядке присуждения ученых степеней» от 24.09.2013 г. №842. Диссертация полностью соответствует паспорту специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей. По теме диссертации автором опубликовано 12 работ. Автореферат правильно отражает содержание диссертации и ее основные результаты.

Белеванцев А.А. является одним из ведущих специалистов в области компиляторных технологий, методов оптимизации и анализа программ. Его научные результаты признаны в России и за рубежом: они были доложены на международных конференциях и опубликованы в российских и ряде зарубежных журналов.

Диссертация Белеванцева А.А. является законченной научно-квалификационной работой, в которой разработаны и математически обоснованы положения и получены практические результаты, которые могут быть в целом квалифицированы как решение крупной научной проблемы, направленной на повышение качества разрабатываемого программного обеспечения. Работа полностью удовлетворяет всем требованиям ВАК, предъявляемым к докторским диссертациям по

специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, а ее автор, Белеванцев Андрей Андреевич, заслуживает присуждения ему ученой степени доктора физико-математических наук по указанной специальности.

Отзыв обсужден и утвержден на заседании секции Ученого совета ИПИ РАН Федерального исследовательского центра «Информатика и управление» Российской академии наук, протокол № 1 от 25 января 2018 года

Главный научный сотрудник ФИЦ ИУ РАН,
Заслуженный деятель науки РФ,
доктор технических наук, профессор

_ И.Н. Синицин
25.01 2018 г.