ОТЗЫВ

официального оппонента Галатенко Владимира Антоновича на диссертационную работу Белеванцева Андрея Андреевича «Многоуровневый статический анализ исходного кода для обеспечения качества программ», представленную к защите на соискание ученой степени доктора физико-математических наук по специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Актуальность

Статический анализ исходного кода программ является эффективным методом поиска ошибок в программах, широко применяемым в промышленных циклах разработки. Его популярность обусловлена возможностью внедрения на ранних этапах разработки, относительной простотой использования, а также наибольшей проработанностью по сравнению с другими методами, в частности, динамическим анализом и верификацией.

Современные инструменты статического анализа должны удовлетворять целому ряду требованиям, таких как поддержка широкой номенклатуры ошибок, эффективная обработка больших объемов кода в сочетании с высоким качеством анализа, масштабируемость, настраиваемость и другие. Постоянный рост требований к инструментам анализа, вытекающий из развития программной отрасли в целом и тенденций к усложнению программ, делает необходимым совершенствование методов анализа, развитие стратегий совместного применения различных методов, а также улучшение интеграции инструментов анализа в процесс разработки. Таким образом, тема диссертационной работы Белеванцева А.А., посвященной развитию методов статического анализа программ, является актуальной.

Структура работы

Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы из 196 наименований.

Во введении обосновывается актуальность исследований, приводятся цель и задачи работы, перечисляются использованные методы исследований,

формулируются научная новизна и практическая значимость полученных результатов, приводятся данные об апробации работы.

В первой главе дается обзор современных методов статического анализа и известных инструментов анализа, обрабатывающих абстрактное синтаксическое дерево программы, а также использующих внутрипроцедурные и межпроцедурные алгоритмы анализа, в том числе различающих контексты вызова функций и пути выполнения программы. Обсуждаются вопросы определения ошибок, методы ранжирования полученных предупреждений об ошибках по вероятности их истинности, сведения о применении анализаторов в промышленной разработке.

Во второй главе описывается предлагаемая методология статического анализа исходного кода, включающая совокупность методов анализа трех уровней, алгоритмы поиска конкретных классов ошибок и архитектуру программной системы, обеспечивающей работу указанных методов и алгоритмов. *Первый уровень* анализа реализует обход абстрактного синтаксического дерева программы, а также внутрипроцедурный анализ программы, для которого разработана модель памяти, поддерживающая языки программирования Си и Си++. Предложены алгоритмы построения этой модели и последующего выполнения анализа, математически обоснована их корректность и сложность (теоремы 2.1 – 2.3). *Второй уровень* выполняет межпроцедурный контекстно-чувствительный анализ программы с использованием аннотаций функций, содержащих результаты внутрипроцедурного анализа. Вводится понятие классов значений, объединяющих одинаковые значения в модели памяти и программы в классы эквивалентности, рассматриваются необходимые модификации алгоритмов анализа (теорема 2.4). Предложен подход, который использует аннотации функции, описывающие результаты ее анализа по отношению к внешним для нее участкам памяти; обосновывается корректность предложенного алгоритма построения аннотаций (теорема 2.5). *Третий уровень* анализа добавляет чувствительность к путям выполнения с помощью символьного выполнения с объединением состояний, где при вычислении нужных значений отслеживаются предикаты, соответствующие путям выполнения, на которых реализуются эти

значения. Обосновывается корректность алгоритмов анализа с предикатами (теорема 2.6).

В третьей главе представлена архитектура программной системы и ее реализация в семействе инструментов Svace, которые обеспечивают работу методов анализа, рассмотренных во второй главе. Описываются методы контролируемой сборки программы, которая позволяет получить нужные представления программы прозрачным для пользователя образом, методы построения собственных компиляторов, генерирующих эти представления, алгоритмы параллельного анализа, реализующие разработанные методы, а также способы хранения результатов анализа и способы их отображения в удобном для пользователя виде.

В четвертой главе рассмотрены алгоритмы поиска конкретных типов ошибок, так называемые детекторы, для языков программирования Си, Си++, Java и C#. Рассмотрены примеры детекторов всех представленных уровней анализа, в том числе алгоритмы поиска ошибок разыменования нулевого указателя, утечек памяти и ресурсов, переполнения буфера и другие.

В пятой главе приводятся результаты применения разработанной программной системы Svace к программам с открытыми исходными текстами различного объема. Демонстрируется работоспособность программных компонентов, описанных в третьей и четвертой главах.

В заключении формулируются основные результаты диссертационной работы и возможные направления дальнейших исследований.

Научная новизна и практическая значимость

В диссертационной работе представлены следующие новые научные результаты:

- методология статического анализа исходного кода программ для поиска ошибок в программах, включающая многоуровневый статический анализ, основанный на разработанном автором наборе моделей программы и методах анализа с общей моделью памяти;
- новые методы, применяемые на уровнях статического анализа абстрактного синтаксического дерева программы, внутрипроцедурного анализа, межпроцедурного контекстно-чувствительного анализа,

чувствительного к путям выполнения анализа для языков программирования Си, Си++, Java, C#;

- формулировки и доказательства теорем о корректности предложенных методов и их сложности;
- алгоритмы поиска некоторых распространенных классов ошибок с помощью разработанных методов анализа;
- архитектура программной системы, обеспечивающая автоматическую работу предложенных методов на протяжении всего процесса анализа и управление набором анализаторов для различных языков, а также единый способ отображения их результатов. Поддерживается автоматическое построение внутренних представлений для анализа, единое переносимое хранилище собранной для анализа информации и результатов анализа, просмотр и разметка результатов анализа, инкрементальный анализ, выполняемый только для изменившейся части программы.

Предложенная в диссертационной работе методология и составляющие ее методы анализа положены в основу программного средства Svace, выполняющего анализ программ на языках Си, Си++, Java, C#. Система Svace внедрена в цикл промышленной разработки компании Самсунг, а также используется в НИЦ «Курчатовский институт». Анализаторы Svace могут быть применены и в других компаниях и организациях.

Достоверность и обоснованность научных положений и выводов работы

Достоверность проведенных исследований подтверждается апробацией результатов на 9 международных и российских научных конференциях и семинарах. По теме диссертационной работы автором опубликовано 12 работ, из них 10 работ в журналах из перечня ВАК РФ и 4 – в изданиях, индексируемых Web of Science. Получено 9 свидетельств о государственной регистрации программ для ЭВМ.

Замечания

По диссертационной работе имеются следующие замечания:

1. В диссертации не приводится оценка сложности алгоритмов построения предикатов пути и классов значений (раздел 2.3).
2. Результаты тестирования инкрементального анализа (стр. 206 диссертации) представлены недостаточно подробно.
3. Недостаточное внимание уделено сравнению результатов, полученных анализатором SVACE, с результатами, полученными другими средствами анализа. Например, с результатами проверки ряда проектов с открытыми исходными текстами при помощи сканера Coverity Scan (https://scan.coverity.com/projects/) или с результатами проверки операционной системы Tizen при помощи PVS-Studio (https://www.viva64.com/ru/b/0508/).
4. Вызывает вопросы одно из требований к методам анализа (стр. 45): "Не менее 60% истинных срабатываний". Не ясно как определить процент истинных срабатываний анализатора в целом, а не для конкретного эталонного примера, и как обосновать выполнение данного требования.
5. В тексте диссертации имеются стилистические погрешности. В частности, словосочетание "доставлять качество" (на стр. 7 и 9) не является принятым в русском языке.

Приведенные замечания не влияют на общую положительную оценку работы.

Заключение

Диссертация Белеванцева Андрея Андреевича «Многоуровневый статический анализ исходного кода для обеспечения качества программ» является завершенной работой, в которой разработаны теоретические положения и получены практические результаты, совокупность которых можно квалифицировать как решение научной проблемы, имеющей важное значение в области разработки программного обеспечения. Основные результаты

диссертации полностью и своевременно опубликованы. Автореферат верно отражает основное содержание диссертации.

Диссертация отвечает требованиям ВАК РФ, предъявляемым к диссертациям на соискание ученой степени доктора физико-математических наук, а ее автор, Белеванцев Андрей Андреевич, заслуживает присуждения ему ученой степени доктора физико-математических наук по специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Доктор физико-математических наук, заведующий
сектором Федерального государственного
учреждения «Федеральный научный центр Научно-
исследовательский институт системных
исследований Российской академии наук»

В.А. Галатенко

Подпись руки В.А. Галат
Начальник отдела кадров

«28» января 2018 г.