ОТЗЫВ

официального оппонента на диссертационную работу
Бородина Алексея Евгеньевича
«Межпроцедурный контекстно-чувствительный статический анализ для поиска ошибок в исходном коде программ на языках Си и Си++»,
представленную к защите на соискание ученой степени кандидата физико-математических наук по специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Актуальность темы исследования

Современный этап развития сферы информационных технологий характеризуется взрывным ростом размеров программного обеспечения. Многие современные проекты насчитывают сотни тысяч и миллионы строк исходного кода. Пропорционально размерам ПО выросло и количество ошибок в программном коде. Таким образом, изменились условия для статических анализаторов ошибок: с одной стороны, требуется анализировать больший объём кода, с другой – увеличилась вычислительная мощность современных компьютеров. Однако правила и эмпирики, заложенные в старые анализаторы, ориентированы на меньшую вычислительную мощность и меньший объём кода, что привело к падению точности таких анализаторов.

В работе Бородина А. Е. рассматривается задача поиска ошибок в исходном коде программ с помощью методов статического анализа. К анализатору предъявляются следующие требования:

1. Масштабируемость на миллионы строк кода.
2. Высокий уровень истинных срабатываний (50-70%).
3. Поддержки популярных языков программирования (C, C++).
4. Возможность расширения анализатора новыми типами исследуемых ошибок.
5. Высокая скорость анализа: миллионы строк кода должны анализироваться за время порядка 4-6 часов.

Существующие в настоящее время свободно доступные статические

анализаторы не позволяют в полной мере решить данную задачу из-за неприемлемого времени работы, низкого уровня истинных срабатываний и высокого уровня ложных срабатываний. Именно поэтому поставленная в работе задача является актуальной.

Следует отметить также актуальность представленных и обоснованных в диссертации методов, основанных на известных алгоритмах статического анализа кода, но видоизменённых применительно к новым условиям.

Степень обоснованности научных положений, выводов и рекомендаций, сформулированных в диссертации

Предлагаемые в диссертации методы и алгоритмы сопровождаются строгим математическим обоснованием. Так, для промежуточного языка svace0 приведена структурная операционная семантика, сформулировано понятие корректности и обоснована корректность внутрипроцедурного анализа функции.

Обоснованность выносимых на защиту научных положений и выводов достигается также публикациями основных результатов работы: по результатам диссертационного исследования опубликовано 6 статей в рецензируемых журналах. Результаты исследования также представлены на 3 научных конференциях.

Сделанные выводы и рекомендации базируются на большом объёме тестовых данных: тестирование инструмента проводилось на 37 проектах с открытым исходным кодом объёмом в миллионы строк. В частности, экспериментально подтвержден один из наиболее важных практических результатов работы: реализованные алгоритмы масштабируются на проекты с миллионами строк кода при сохранении высокого процента истинных срабатываний (около 60%) и приемлемой скорости работы. Таким образом, выводы и рекомендации, представленные А. Е. Бородиным, строго аргументированы и логичны.

Достоверность результатов

Достоверность результатов диссертации подтверждается работающей программной реализацией и проведенными с ее помощью тестовыми исследованиями. Среди исследованных проектов – операционная система Tizen 2.3 и операционная система Android 5.0.2. Кроме того, было обеспечено внешнее тестирование: инженеры компании Samsung использовали предложенный инструмент для тестирования внутреннего проекта. В приложении Б представлены результаты, свидетельствующие о том, что большинство видов дефектов обнаруживается с вероятностью более 70%.

Достоверность результатов также подтверждается использованным в работе современным научным аппаратом (анализ графа потока управления, внутри- и межпроцедурный анализ, анализ на основе резюме и проч.), адаптированным к объекту исследования.

Новизна исследования

Новизна исследований и разработок А. Е. Бородина подтверждается анализом аналогичных свободных разработок для поиска ошибок в программном коде методами статического анализа. В работе убедительно показано, что относительно старые анализаторы кода не могут обеспечить приемлемый уровень обнаружения дефектов при приемлемом быстродействии.

Кроме того, новизна исследования заключается в использовании сочетания стандартных и новых методов анализа, а также ряда эвристик – именно данное сочетание позволило достичь хороший уровень выявляемости ошибок за приемлемое время.

Большое число ссылок на литературные источники, 9 публикаций результатов в научных изданиях дают дополнительное подтверждение новизны и востребованности результатов исследований, относящихся к тематике диссертации.

Содержание диссертации и ее теоретическая и практическая значимость

В диссертационной работе представлено описание вклада автора в сложную и актуальную задачу поиска ошибок в больших и сверхбольших программных проектах методами статического анализа кода.

Диссертация состоит из введения, шести глав и заключения. Объем диссертации составляет 137 страниц, включая 12 таблиц и 26 рисунков.

Во введении сформулирована цель диссертационной работы, дана постановка задач; обоснована актуальность и научная новизна выбранного направления исследования; сформулированы основные положения, выносимые на защиту; описана структура диссертации.

В первой главе производится обзор существующих статических анализаторов для поиска ошибок в исходном коде и описывается инструмент статического анализа Svace, разрабатываемый в ИСП РАН. Инструмент Svace позволяет производить поиск ошибок в исходном коде программ, написанных на языках C/C++, Java и C#. Работа посвящена улучшению семантического анализатора для поиска ошибок в исходном коде C/C++. Существующий анализатор соответствовал всем предъявляемым требованиям для проектов среднего размера (в сотни тысяч строк кода), но имел недостаточную скорость и качество анализа для проектов в миллионы строк кода. На основе анализа существующих подходов в качестве основы был выбран межпроцедурный анализ на основе резюме, при котором для анализа функции программы используется резюме, представляющее собой краткое описание интересующего поведения функции.

Во второй главе описывается язык svace0, используемый в качестве внутреннего представления.

Анализ разделен на ядро, выполняющее общие действия, и детекторы. Ядро анализа описывается в третьей главе, а детекторы в четвертой. Ядро выполняет нумерацию значений и анализ указателей. Для анализа строится граф потока управления, с ребрами графа ассоциируется абстрактное состояние, описывающее множество конкретных состояний, которые возможны при

выполнении программы. Анализ заключается в продвижении абстрактных состояний по графу потока управления. С целью ускорения анализа внутри функции не рассматриваются все возможные пути выполнения, а только часть путей. Также используется предположение, что среди входных параметров процедуры нет псевдонимов. Доказывается, что для заданных начальных состояний и рассматриваемых путей обхода внутри функции абстрактные состояния будут описывать возможные при выполнении конкретные состояния. При обходе каждой инструкции нумерация значений, анализ указателей и анализ отдельных детекторов выполняется одновременно. Благодаря этому после анализа инструкции входное абстрактное состояние больше не требуется и может быть удалено. Реализация отдельных детекторов показана на основе поиска ошибок разыменования нулевых указателей. В работе приведено три реализации: на основе прямого анализа, с использованием обратного анализа и с чувствительностью к путям. Реализация с чувствительностью к путям более сложная и требует больше процессорного времени и оперативной памяти, но позволяет находить больше ошибок.

Межпроцедурный анализ описывается в пятой главе. На основе абстрактного состояния в единственной точке выхода из функции формируется резюме — описание интересующих эффектов функции. В дальнейшем резюме используется при анализе инструкции вызова функции. Все функции обходятся по графу вызовов снизу вверх таким образом, чтобы вызываемые функции анализировались до вызывающих. Циклы в графе вызовов разрываются. В работе предложен механизм повышения контекстной чувствительности, при которой объединение анализируемых свойств в точках слияния путей внутри функции откладывается до точки вызова функции. В конце главы рассматривается модификация анализа, позволяющая находить неконсистентность конструкторов и деструкторов C++. Для этого в начало тела деструктора вставляется вызов конструктора. В точке выхода из деструктора производится анализ, зависящий от детектора. В частности, выполняется поиск утечек памяти, позволяющий обнаружить отсутствие освобождения в деструкторе памяти, выделенной в конструкторе.

Шестая глава посвящена реализации предложенных алгоритмов и оценке результатов анализа. Для более эффективного многопоточного анализа создан диспетчер выбора следующей функции для анализа, позволяющий минимизировать количество хранимых в памяти резюме и повысить локальность функций относительно модуля; что позволяет сократить потребление памяти и увеличить скорость анализа. Описываются вспомогательные алгоритмы: для анализа неизвестных функций; анализа значений, которые могли быть сохранены; анализа логических переменных. Эти алгоритмы позволяют повысить точность реализации отдельных детекторов. В конце главы производится оценка времени и качества анализа. В соответствии с оценкой реализованный инструмент удовлетворяет всем предъявленным к анализатору требованиям. Кроме этого, сравнивается старая версия анализатора с новой для проекта Android 4.4: при сохранённом уровне истинных срабатываний количество предупреждений увеличилось более чем в 4 раза.

Заключение содержит основные результаты диссертационной работы:

1. Разработан алгоритм внутрипроцедурного анализа функции. Доказана его корректность.
2. Разработан межпроцедурный контекстно- и потоково- чувствительный анализ, основанный на алгоритме создания резюме функции.
3. Разработанные алгоритмы реализованы в статическом анализаторе Svace.

Теоретическая значимость работы заключается в развитии методов и алгоритмов статического анализа программного кода, а также в обосновании корректности такого анализа.

Практическую значимость трудно переоценить. Значительный массив тестируемого кода и высокий процент истинных срабатываний (более 60%) за приемлемое время, а также внешняя проверка инженерами фирмы Samsung свидетельствуют о высокой практической значимости настоящего исследования.

Замечания по работе

1. Работа иллюстрируется значительным количеством примеров находимых ошибок. Тем не менее, отсутствует полная классификация ошибок, для поиска которых строится анализатор.
2. Алгоритм анализа циклов имеет экспоненциальную сложность в зависимости от глубины вложенности циклов. В разделе 3.4 говорится, что обычно в проектах не используются сильно вложенные циклы, но в анализируемых проектах не проводится исследование на максимальное количество вложенных циклов.
3. В название работы вынесен «межпроцедурный контекстно-чувствительный статический анализ», однако в качестве первого результата работы выделяется «алгоритм внутрипроцедурного анализа функции», что приводит к некоторому логическому противоречию.
4. В заключении приводятся заверения автора по поводу направлений дальнейшей работы. Видимо, неуместно формулировать заведомо не сделанное в таком виде в заключении. Однако данный недостаток легко исправить, написав перед предлагаемыми направлениями дальнейших исследований фразу «Результаты, полученные в настоящей работе, позволили выявить актуальные направления для дальнейшего исследования: …».
5. Работа содержит незначительное количество орфографических ошибок и описок. Так, на странице 28 слово «предупреждений» написано неправильно, на той же странице для слова «миллионов» выбрана неправильная форма, на стр. 30 в словосочетании «приводит к потере» выбрана неправильная форма.
6. На странице 28, очевидно, содержится логическая ошибка: вместо «размером сотни строк кода» следует скорее всего читать «размером сотни тысяч строк кода».

Заключение

Сделанные замечания не снижают общей научной и практической ценности результатов работы и не влияют на общую положительную оценку диссертации.

Диссертационная работа Бородина Алексея Евгеньевича является завершенным научным исследованием, а разработанный инструмент статического анализа, являющийся частью системы Svace, внедрен в коммерческой компании Samsung, что свидетельствует о практической значимости работы. Полученные научные результаты в опубликованы в 9 научных работах, 6 из которых входят в Перечень рецензируемых научных изданий. Автореферат полно и правильно отражает содержание диссертационной работы.

Диссертационная работа отвечает всем требованиям ВАК РФ, предъявляемым к диссертациям, представленным на соискание ученой степени кандидата физико-математических наук, а Бородин Алексей Евгеньевич заслуживает присуждения ему ученой степени кандидата физико-математических наук по специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

Доцент кафедры алгебры и дискретной математики
Института математики, механики и компьютерных наук
им. И.И. Воровича Южного федерального университета,
кандидат физико-математических наук
(специальность 01.01.02 Дифференциальные уравнения)

Станислав Станиславович
Михалкович

Почтовый адрес: ул. Мильчакова 8-А, оф. 204, г. Ростов-на-Дону, 344090
Тел.: +7(863)297-51-14 (доп.) 204
e-mail: miks@sfedu.ru