"УТВЕРЖДАЮ"
ерального Исследовательского
«Информатика и Управление»
академик РАН И.А.Соколов
2016 г.

Отзыв

ведущей организации на диссертацию Бородина Алексея Евгеньевича «Межпроцедурный контекстно-чувствительный статический анализ для поиска ошибок в исходном коде программ на языках Си и Си++», представленной на соискание ученой степени кандидата физико-математических наук по специальности 05.13.11 (математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей)

Ошибки в программном обеспечении во многом определяют качество ПО. Вместе с тем множество ошибок остается не найденным в течении многих лет после выпуска ПО. Одним из методов поиска ошибок, широко используемым в индустрии, является статический анализ программ. Вычисление необходимых свойств программы является алгоритмически неразрешимой задачей. Поэтому статические анализаторы осуществляют поиск приближенного решения, что приводит к выдаче ложных предупреждений.

Статический анализатор Svace, разрабатываемый в ИСП РАН, продемонстрировал высокое качество анализа проектов размером в сотни тысяч строк кода. Но при переходе к анализу проектов в миллионы строк кода (ОС Android, Tizen) качество анализа сильно упало из-за недостаточной точности анализатора. Актуальной является задача разработки алгоритмов анализа, достаточно точных для обеспечения высокого качества анализа для сверхбольших проектов.

В ставится задача поиска ошибок в исходном коде для использования в жизненном цикле разработки ПО, производится обзор существующих подходов и описывается инструмент статического анализа Svace, где предполагается реализовывать анализ. На основе обзора существующих подходов сделаны выводы о возможности пропуска части дефектов для достижения высокого уровня истинных срабатываний и необходимости поддерживать чувствительный к путям и контексту анализ.

Автором работы разработан язык svace0, являющийся внутрипроцедурным представлением анализа. Для анализа программ, написанных на языках Си и Си++, используется компилятор Clang, генерирующий биткод LLVM. Компиляция осуществляется без использования оптимизаций, и сгенерированный код содержит только подмножество инструкций LLVM. Файлы LLVM преобразуются во внутреннее представление анализатора,

и язык svace0 описывает упрощённое внутреннее представление, достаточное для описание анализа. Семантика языка описывается с помощью структурной операционной семантики.

Ядро анализа функции вычисляет базовые данные про память и значения переменных. Описываемый анализ можно рассматривать как комбинацию символьного выполнения и анализа потока данных. Символьное выполнение не используется из-за проблемы экспоненциального роста возможных путей. Анализ потока данных не используется из-за большего времени, требующегося на анализ всех возможных путей выполнения. Вместо этого используется символьное выполнение с объединением состояний анализа в точках слияния путей. Для определения классов эквивалентности значений переменных вводятся объекты, называемые идентификаторами значений. Идентификаторы значений похожи на номера значений, используемые оптимизирующими компиляторами, но имеют ряд отличий. В частности на разных итерациях цикла для переменных создаются новые идентификаторы значения, если не доказано, что переменные не изменяются внутри цикла. Для анализа указателей используются ссылки, являющиеся подмножеством идентификаторов значений и обозначающие непересекающиеся области памяти. Анализ циклов представляет собой обход N итераций циклов (по умолчанию N=3) с последующим объединением абстрактных состояний для отдельных итераций. В разделе 3.5 доказывается корректность анализа для заданного набора путей (путей, которые можно обойти за N итераций). У анализа нет задачи рассмотреть все возможные пути внутри функции.

Разработаны детекторы для поиска внутрипроцедурных ошибок. Детекторы реализуются как расширения анализа. Во время обхода графа потока управления ядро анализа оповещает детекторы о всех событиях, при этом все детекторы оповещаются одновременно. После анализа инструкции не требуется хранить входное состояние, что позволяет уменьшить потребление памяти. В разделе 4.2 описывается критерий выдачи предупреждений. Предупреждение выдаётся, если существует ребро в графе потока управления такое, что все пути, проходящие через него, содержат ошибку. Критерий позволяет анализировать неполные программы, а также производить анализ функций без информации о контексте вызова. В конце главы на примере анализа нулевых указателей описывается три вида возможных реализаций детекторов: использующих прямой анализ, обратный анализ и имеющих чувствительность к путям. Анализ, чувствительный к путям, различает пути, по которым управление могло достигнуть некоторой точки. Для реализации свойства анализа представляются в виде формул логики высказываний. В качестве литеральных формул используются сравнения констант и идентификаторов значений.

За основу межпроцедурного анализа взят анализ на основе резюме, при котором после анализа функции формируется её резюме, кратко описывающее поведение функции. Резюме

строится на основе состояния анализа в единственной точке выхода из функции. Резюме содержит как информацию о значении переменных и форме памяти, так и специфичную для детекторов информацию. Для реализации межпроцедурного поиска дефектов детекторам требуется подписаться на события создания резюме и трансляции резюме в контекст вызывающей функции. Трансляция резюме происходит отдельно для каждого контекста вызова, благодаря чему обеспечивается чувствительность к конкретному контексту вызова.

Разработанные алгоритмы реализованы в инструменте Svace. В процессе трансляции биткода LLVM, который подаётся на вход анализатору, для сохранения максимальной информации об оригинальной программе и поддержания множества диалектов и расширений Си и Си++ было сделано более 500 модификаций компилятора Clang. В разделе 6.4 описывается параметризация алгоритма создания резюме, описываются итерации над состоянием анализа для создания резюме. Максимальное количество элементов в резюме ограничено значением 250. Резюме обрезаются с целью минимизации размеров. В конце главы производится оценка скорости и качества анализа, подтверждающие масштабируемость созданного анализатора.

В рамках работы были получены следующие результаты:

- Разработан алгоритм анализа функции, обеспечивающий возможность поиска широкого класса дефектов.
- Разработан алгоритм межпроцедурного анализа функций, основанный на использовании резюме — краткого описания поведения функции.
- Разработанные алгоритмы реализованы в подсистеме статического анализатора Svace для поиска ошибок в исходном коде программ на Си и Си++. Экспериментальные результаты анализа ОС Andoroid и Tizen подтвердили масштабируемость алгоритмов при высоком качестве выдаваемых предупреждений (более 60% истинных срабатываний).

По работе могут быть сделаны следующие замечания:

1. В работе не описывается, каким образом разрываются циклы в графе вызовов, что затрудняет понимание возможной потери точности анализа для рекурсивных вызовов.

2. Межпроцедурный анализ параметризован максимальным количеством хранимых элементов M и количеством обходов элементов N для создания резюме. В шестой главе описывается выбор этих параметров, но не приводится сравнение эффективности разных сочетаний этих параметров.

Перечисленные замечания не влияют на положительную оценку диссертационной работы и не ставят под сомнение полученные в ней результаты. Результаты диссертации своевременно опубликованы. Автореферат полно и правильно отражает содержание

диссертации.

Диссертация Бородина А. Е. является законченной научно-исследовательской работой и удовлетворяет всем требованиям ВАК, предъявляемым к диссертациям на соискание ученой степени кандидата физико-математических наук по специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, а Бородин Алексей Евгеньевич заслуживает присуждения ученой степени кандидата физико-математических наук по специальности 05.13.11.

Отзыв на диссертацию составлен доктором физико-математических наук Серебряковым Владимиром Алексеевичем и обсужден на семинаре отдела систем математического обеспечения Вычислительного центра им. А.А.Дородницына ФИЦ ИУ Российской академии наук 16.05.2016 протокол №01.05.16.

Зав. отделом ВЦ РАН
д.ф.м.н., профессор В. А. Серебряков