# ОТЗЫВ

**Официального оппонента на диссертационную работу**
**Саргсяна Севака Сениковича**
**«Методы поиска клонов кода и семантических ошибок на основе семантического анализа программы»,**

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»

Стремительный рост объема и сложности программного обеспечения ведет к пропорциональному росту ошибок в программах. Размер современных широко используемых программ варьируется в диапазоне от нескольких сотен тысяч до десятков миллионов строк исходного кода. Одной из причин столь быстрого роста размера кода является часто встречающаяся практика дублирования или клонирования (дублирования с адаптацией) отдельных частей кода. Это ведет не только к разрастанию размера кода, но и к внесению в него ошибок при клонировании.

Диссертационная работа Саргсяна Севака Сениковича посвящена разработке методов и созданию масштабируемых и высокоточных инструментов поиска клонов в исходных кодах программ, а также семантических ошибок, возникающих при некорректной адаптации клонированных фрагментов кода. *Актуальность* работы заключается в том, что автором предложены и практически реализованы методы поиска клонов, которые могут применяться в цикле разработки программного обеспечения (ПО) для анализа качества кодов и поиска допущенных ошибок. Высокая точность предложенных средств и их масштабируемость до программ размером в несколько десятков миллионов строк исходного кода позволяют рассматривать результаты работы как *новые* и даже уникальные, не имеющие аналогов среди свободно доступных инструментов той же направленности.

Диссертация состоит из введения, пяти глав, заключения и списка литературы, который содержит 98 наименований.

**Во введении** обосновывается актуальность работы, определяются цели и задачи работы, формулируются основные положения, выносимые на защиту, обосновывается теоретическая и практическая значимость работы.

**Первая глава** посвящена *обзору методов и существующих инструментов*, решающих задачу поиска клонов в программах, а также обнаружения семантических ошибок. В ней дается классификация клонов по типам и приводится

критический анализ 5 распространенных методов поиска клонов (текстовый, лексический, синтаксический, метрический, семантический), а также их комбинаций. В результате делается вывод, что существующие инструменты либо не обладают достаточной точностью, либо необходимой масштабируемостью для поиска клонов в очень больших программах. Приводится основные типы семантических ошибок, выявляющиеся при адаптации клонированных фрагментов кода в программах.

**Во второй главе** рассматривается предложенный автором *новый четырехфазный метод поиска клонов кода*, основанный на семантическом анализе программы. Метод базируется на построении графа зависимостей программы (ГЗП), на базе которого предложено несколько принципиально новых алгоритма разбиения ГЗП на единицы сравнения (ЕС), отсеивания ненужных ЕС, поиска максимально схожих подграфов и фильтрации ложных клонов. Предложенный подход позволяет обнаруживать клоны кода в больших проектах с высокой точностью. Это практически подтверждено на базе анализа проектов Linux 2.6, Mozilla Firefox, LLVM/Clang, OpenSSL, размеры исходных кодов которых варьируются от 270 тыс. до 13,9 млн. строк. Проведенное сравнение с известными инструментами поиска клонов MOSS, CCFinder, CloneDR показывает высокую эффективность предложенных автором инструментов, как в части числа обнаруживаемых клонов, так и в части их масштабируемости к большим проектам.

**В третьей главе** описывается архитектура разработанного автором инструмента поиска клонов на базе компиляторной инфраструктуры LLVM. Это представление программы является основой для построения ГЗП и используется для реализации таких широко распространенных языков программирования как C и C++. Инструмент дополнен отдельным компонентом, *принципиально новым генератором клонов* для тестирования точности реализованных алгоритмов поиска клонов. Инструмент практически применен к тем же программным проектам, что и в 2-й главе. При этом точность обнаружения искусственно сгенерированных клонов с помощью алгоритма на основе слайсинга (самый точный, но и самый медленный алгоритм) превышает 90%.

**В четвертой главе** рассматриваются методы поиска клонов кода в программах, реализованных на языке JavaScript. Автором предложен и реализован *оригинальный метод* модификации динамического компилятора V8 для этого языка, который позволяет построить ГЗП для всей программы. Затем используются разработанные и представленные в главе 2 инструменты поиска клонов. Автор проверил эффективность поиска клонов на нескольких реальных программах, реализованных на языке JavaScript, а также продемонстрировал почти 10-кратное

преимущество предложенного в работе метода по сравнению с методом инструмента CloneDR по числу обнаруженных клонов на самой большой программе Octane.

**В пятой главе** предложен *новый комплексный метод обнаружения семантических ошибок*, возникающих при некорректной адаптации скопированного фрагмента исходного кода к контексту, в который он был вставлен. Инструмент одновременно использует лексический и семантический анализ программы, что позволяет обнаруживать ошибки, возникающие, как правило, из-за некорректного переименования переменных при копировании. Этот подход был применен к тем же большим проектам, что и в главе 2, а также к операционной системе Android 4.3 и эмулятору аппаратных платформ Qemu и позволил выявить в них более 150 семантических ошибок.

**В заключении** приводятся основные научные и практические результаты, полученные в ходе выполнения диссертационной работы, а также определяются направления дальнейшей работы по данной тематике, в частности поиск уязвимостей на основе существующих шаблонов, а также применение инструмента в задачах оптимизации размера исходного кода программ.

Несмотря на высокое качество работы в ней можно отметить ряд недостатков:

1. Во 2-й главе указано, что более 90% клонов кода, найденных с помощью более быстрых алгоритмов (изоморфизм деревьев и на основе метрик) попадают в клоны, обнаруживаемые самым эффективным алгоритмом на основе слайсинга, но не проведен анализ того, почему 10% оставшихся клонов не были обнаружены методом на основе слайсинга;

2. Алгоритмы поиска клонов и семантических ошибок дают ложные срабатывания, но не всегда объясняются их причины и не везде приведены методы их отсеивания;

3. Упоминаемый во 2-й и 4-й главах инструмент CloneDR не включен в анализ, проведенный в 1-й главе, поэтому не понятно, за счет чего предложенный в работе метод обнаружения клонов находит почти в 10 раз больше клонов в программе на языке JavaScript.

Отмеченные недостатки не влияют на общую положительную оценку работы.

Диссертационная работа Саргсяна С.С. является законченным научным исследованием, а реализованные и практически испытанные на больших реальных программах инструменты подтверждают ее *практическую значимость*. Все

основные новые алгоритмы, предложенные в работе, *научно обоснованы* с использованием математического аппарата. *Достоверность работы* полностью подтверждается проведенными экспериментами. Полученные автором результаты отражены в опубликованных им работах и *прошли апробацию* на нескольких конференциях.

*Автореферат* в сжатом виде, но при этом *полно, правильно* отражает содержание диссертационной работы.

Диссертационная работа Саргсяна Севака Сениковича соответствует всем требованиям ВАК РФ, предъявляемым к диссертациям на соискание ученой степени кандидата физико-математических наук, а ее автор, Саргсян Севак Сеникович, заслуживает присуждения ему ученой степени кандидата физико-математических наук по специальности 05.13.11 – «математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Официальный оппонент
кандидат технических наук Волконский В.Ю.

Подпись кандидата технических наук
Волконского В.Ю. заверяю
Зам. Ген. директора ПАО «ИНЭУМ им.
И.С.Брука» по науке Перекатов В.И.

"24" февраля 2016 г.