РЕЦЕНЗИЯ
на выпускную квалификационную работу обучающегося СПбГУ
Мисонижника Александра Владимировича
по теме Композициональные частично определенные типы в символьном интерпретаторе для платформы .NET Framework

ВКР мотивирована задачей формальной верификации объектно-ориентированных программ путем сведения к формулам первого порядка и поиску их моделей, используя автоматические SMT-решатели (от англ. Satisfiability Modulo Theories – задача выполнимости формул в теориях). На сегодняшний день существуют достаточно производительные SMT-решатели, с помощью которых можно находить ошибки (или доказывать их отсутствие) в программах в десятки и сотни тысяч строк кода. Однако, непременным условием для этого являются принадлежность формулы к разрешимому фрагменту логики и адекватность процесса трансляции программы в формулу. Формула должна корректно отражать всю функциональность программы, которая может воздействовать на потенциальные ошибки.

В работе затронута проблема неопределенности динамических типов объектов, повсеместно используемых при программировании на платформе .NET. Во время трансляции программы в формулу верификатор использует так называемое символьное исполнение, в ходе которого анализируется код на неопределенных входных данных и выводятся факты о достижимости или недостижимости определенных состояний программы. Однако, в случае динамических типов данных и полиморфизма эффективность символьного исчисления поставлена под сомнение, поскольку возникает необходимость обработать все возможные интерпретации неопределенных данных, что экспоненциально увеличивает область анализа. На данный момент ни один из существующих инструментов не поддерживает символьное исполнение кода с динамическими (открытыми) типами в полном объеме. ВКР Мисонижника Александра Владимировича ставит амбициозную задачу разработать алгоритм символьного исполнения функций с учетом открытых типов и полиморфизма, а также реализовать его на основе V#, верификатора для .NET. Предполагается, что результат работы не будет иметь аналогов не только среди верификаторов .NET, но и среди верификаторов других объектно-ориентированных языков, таких как Java, C++ и Scala.

ВКР структурирована следующим образом: после введения и обзора литературы, описывается архитектура подсистемы V#, ответственной за символьное исполнение. Далее в работе приводятся детали реализации символьного исполнения с учетом неопределенности открытых типов. Последняя и самая объемная часть работы посвящена формализации системы ограничений на типы, которая адекватна спецификации .NET, и алгоритму сведения задачи выполнимости системы таких ограничений к задаче выполнимости формулы логики первого порядка.

Результаты ВКР состоят из двух больших частей: теоретической и практической. Теоретическая часть, связанная с системой ограничений на типы, сама по себе является весомым результатом, достойным публикации на международной конференции ранга POPL, PLDI или OOPSLA. Ключевая идея построения формулы первого порядка, которая описывает все необходимые ограничения .NET, состоит в том, чтобы сузить область поиска моделей формулы на этапе ее построения. Система ограничений основана на отношении подтипирования, и поскольку известен факт неразрешимости подтипирования, SMT-решатели неэффективны (а на практике вообще не работают) при наивно построенных формулах. Чтобы обойти это препятствие, ВКР представляет алгоритм построения формулы в терминах EPR (от англ. Effectively PRopositional logic

– эффективно- пропозициональная логика). С помощью предложенных ограничений на систему типов задача становится разрешимой и SMT-решатели гарантировано находят модель (или доказывают ее отсутствие), тратя при этом незначительное время (на практике – секунды). ВКР формально обосновывает адекватность такой кодировки, а также предоставляет способ декодировки модели, найденной SMT-решателем. Теоретические результаты описаны в ВКР математически точно и последовательно, включая все необходимые обозначения и предпосылки. Текст доступен читателям с практически любой компетенцией в области компьютерных наук.

Вторая часть результатов (практическая) связана с реализацией всей подсистемы символьного исполнения в V#, которое включает в себя поддержку не только типов, но и композициональности, которая позволяет использовать уже проинтерпретированные участки кода повторно. Эта часть работы, к сожалению, описана в тексте недостаточно подробно, но она без сомнения представляет огромный практический потенциал. Исходный код подсистемы со всей заявленной функциональностью доступен в репозитории V# и служит платформой для дальнейших исследований в области верификации, тестирования и автоматического синтеза кода программ. К примеру, стоит отметить что символьное исполнение используется не только для трансляции в формулу для дальнейшей верификации, но и для реального выполнения отдельных .NET-функций, когда не все типы до конца известны. С помощью типов, синтезированных во время символьного исполнения, становится возможным выполнять такие NET-функции в изоляции от контекста вызова.

Текст ВКР написан технически грамотным языком, хотя содержит незначительное количество опечаток. Список литературы оформлен по стандарту и содержит ссылки на актуальные работы и ресурсы. Важно отметить, что вся указанная литература – англоязычная. ВКР не содержит неправомерных заимствований. В итоге, была проделана объемная работа, которая, выходит далеко за рамки бакалаврской. Без сомнений, ВКР заслуживает оценки «отлично», а ее исполнитель – присуждения звания бакалавра.

«20» мая 2018 г.  *Федюкович Григорий Геннадьевич*

Подпись *ФИО*