УТВЕРЖДАЮ
Проректор по научной работе
Федерального государственного бюджетного
образовательного учреждения высшего
образования
«Санкт-Петербургский государственный
университет»
С. В. Аплонов

_____ 2016 г.

ОТЗЫВ
ведущей организации
Федерального государственного бюджетного образовательного
учреждения высшего образования
«Санкт-Петербургский государственный университет»
на диссертацию Мандрыкина Михаила Усамовича
«Моделирование памяти Си-программ для инструментов статической верификации на основе SMT-решателей», представленную на соискание ученой степени
кандидата физико-математических наук по специальности
05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Актуальность

Проблема эффективного моделирования операций с указателями в Си-программах в контексте формальной верификации является широко известной. Для решения этой задачи создано множество подходов. В частности, широко известна сепарационная логика (separation logic) и её различные расширения (например, для поддержки массивов). Различные вариации сепарационной логики были использованы при верификации кода ядер ОС, например, в проекте по верификации микроядра L4. Существуют также методы автоматизации некоторых практически важных фрагментов сепарационной логики, таких как теория списков, применённая для верификации фрагмента ядра ОС Windows. Методы моделирования памяти, основанные на сепарационной логике, применяются в интерактивной верификации и характеризуются использованием мощных правил вывода, позволяющих за небольшое число шагов доказывать истинность большого числа условий. Такие правила, однако, плохо подходят для полной автоматизации, например, в контексте применения автоматических решателей для формул в теориях (SMT-решателей). В контексте использования сепарационной логики с помощью SMT-решателей, как правило, вводятся существенные ограничения на синтаксис и семантику верефицируемых программ,

а также на допустимый вид спецификаций корректности, что позволяет сформулировать систему правил вывода, подходящую для полной автоматизации.

В рамках некоторых проектов по верификации системного Си-кода применяются методы моделирования памяти, позволяющие генерировать условия верификации, оптимизированные для SMT-решателей. Эти методы, как правило, отличаются большей полнотой как в смысле поддержки возможностей языка Си, так и в смысле допустимых спецификаций, однако не гарантируют полноту в смысле разрешимости получаемых условий верификации. Вместо этого при разработке метода ставится цель минимизации числа неразрешенных автоматическим путём условий, а также времени верификации. Основным преимуществом этого класса методов является отсутствие необходимости написания и последующей поддержки кода интерактивных доказательств для большинства генерируемых условий верификации. Один из таких методов, основанный на использовании простого предварительного статического анализа синонимичности указателей, показал свою практическую эффективность. При этом время, требуемое для автоматического разрешения генерируемых условий, оказалось наименьшим среди методов того же класса. Данный метод, однако, накладывал существенные ограничения на допустимые к использованию возможности языка Си, в частности, запрещал использование вложенных структурных типов. Ослабление данных ограничений является актуальной задачей в контексте применения методов моделирования памяти для верификации кода ядер ОС, существенно использующих такие возможности Си как вложенные структуры и др., оказавшиеся за рамками данного метода. Снятию этих ограничений в рамках данного метода посвящена работа М. У. Мандрыкина.

Так как формализация и обоснование корректности существующего метода моделирования памяти в значительной степени опирается на введённые ограничения, для расширения области применимости метода потребовалась переработка как формализации метода, так и соответствующих доказательств. Новая формализация также включает ранее отсутствующие доказательства полноты метода (как для случая использования дополнительных спецификационных конструкций, так и для ограничения семантики входного языка без дополнительных конструкций). При этом удалось сохранить эффективность метода, порождая эквивалентные результирующие условия верификации для ранее поддерживаемого фрагмента языка Си.

Предложенный в работе метод моделирования памяти Си-программ совмещает использование простого предварительного автоматического анализа синонимичности указателей с использованием специальных спецификационных конструкций, позволяющих изменять или ослаблять предположения о синонимичности указателей как на уровне отдельных функций (соответствует унификации регионов), так и на уровне отдельных линейных участков кода (соответствует переинтерпретации участков памяти). При этом проверка корректности моделирования памяти совмещена с проверкой корректности использования механизма защиты памяти, предоставляемого соответствующей частью ядра ОС.

В работе также предложен относительно простой метод моделирования памяти для инструментов автоматической проверки достижимости ошибочных состояний на основе предикатной абстракции с уточнением на основе интерполяции Крейга.

Оба предложенных метода были реализованы и апробированы в используемых на практике инструментах верификации.

Структура работы

Диссертационная работа М. У. Мандрыкина состоит из введения, четырёх глав, заключения, списка литературы, включающего 174 наименования, списка свидетельств о государственной регистрации программ для ЭВМ и одного приложения. Общий объем работы – 207 страниц.

Во **введении** обоснована актуальность темы, сформулирована цель работы и определен перечень решаемых задач, указана научная новизна и отмечена теоретическая и практическая значимость полученных результатов, приведены ссылки на публикации и зарегистрированные программы по теме работы, перечислены конференции и семинары, на которых были апробированы полученные результаты.

В **первой** главе приведён обзор исследований в области моделирования памяти Си-программ в инструментах статической верификации, использующих SMT-решатели. Введено понятие модели памяти как формализации семантики подмножества языка программирования, обеспечивающего работу с адресуемыми данными в памяти программы, с использованием частично или полностью автоматизированных логик (комбинаций логических теорий). Также представлен обзор существующих моделей памяти, использованных ранее в контексте статической верификации Си-программ с применением автоматических инструментов проверки выполнимости логических формул. Наиболее важными выводами первой главы можно считать следующие:

- модели памяти Си-программ, используемые в инструментах статической верификации на основе автоматических решателей, существенно отличаются как от моделей памяти, используемых при описании его семантики (например, в различных версиях стандартов), так и от моделей памяти вычислительных машин, используемых, например, в контексте динамической верификации;
- модели памяти Си-программ, используемые в инструментах статической верификации, существенно отличаются по такими важными характеристикам, как поддерживаемое подмножество языка Си, используемые дополнительные предположения о проверяемых программах, число успешно автоматически разрешаемых формул и суммарного времени, требуемого на их разрешение;
- поддерживаемое подмножество языка Си для моделей памяти, показывающих наибольшую эффективность, не охватывает большую часть возможностей этого языка программирования, используемых в коде промышленных систем, таких как ядра ОС.

Во **второй** главе рассмотрены основные проблемы существующих моделей памяти для языка Си в контексте автоматической статической и дедуктивной верификации модулей ядер ОС. Выделены следующие проблемы:

- несовместимость существующих моделей памяти и интерполяции Крейга при совместном применении к результирующим логическим формулам;
- неприменимость существующих эффективных моделей памяти для ядер ОС, что связано с отсутствием поддержки вложенных структурных типов, а также ограниченностью поддержки объединений, приведений типов указателей и адресной арифметики в этих моделях памяти;
- несовместимость семантики, использованной в существующих доказательствах корректности и полноты моделей памяти, с семантикой соответствующего подмножества языка Си.

В **третьей главе** описана предложенная в работе модель для ограниченных областей памяти на основе теории неинтерпретируемых функций. Модель предназначена для использования в инструментах автоматической статической верификации, использующих предикатную абстракцию с уточнением по контрпримерам с помощью интерполяции Крейга. Основные результаты третьей главы заключаются в следующем:

- предложена модель памяти, совместимая с применением интерполяции Крейга, позволяющая анализировать программы, содержащие адресную арифметику и работающую с областями динамической памяти фиксированного размера;
- предложенная модель памяти была реализована в инструменте автоматической статической верификации CPAchecker и апробирована на наборе задач верификации для драйверов ОС Linux;

- результаты апробации предложенной модели памяти показали существенное уменьшение числа ложных срабатываний по сравнению с ранее используемой моделью памяти.

В **четвертой** главе предложена модель памяти с вложенными регионами для дедуктивной верификации Си-программ. Наиболее важными результатами четвертой главы следует считать:

- модель памяти с вложенными регионами, которая расширяет область применимости дедуктивной верификации Си-программ; при этом предложенная модель по сравнению с существующими поддерживает произвольную вложенность структурных типов, а также объединения и приведение типов указателей;
- доказательство корректности и полноты предложенной модели памяти;
- метод автоматического вывода дополнительных спецификаций, необходимых для обеспечения полноты предложенной модели памяти, а также соответствующую схему доказательства полноты предложенного метода для ограниченного класса Си-программ.

Предложенная модель памяти была реализована в инструменте дедуктивной верификации Jessie и позволила применить этот инструмент при верификации модуля безопасности для ядра ОС Linux.

Научная новизна результатов исследования

Научную новизну представленной диссертационной работы составляет следующее.

- Формализация метода моделирования памяти Си-программ для инструментов дедуктивной верификации на основе SMT-решателей, поддерживающая автоматизированное разбиение памяти на непересекающиеся области (регионы), произвольную вложенность структурных типов, объединения и приведение указателей.
- Доказательство корректности предложенного метода моделирования памяти Си-программ относительно формализованной семантики соответствующего подмножества языка Си.
- Доказательство полноты предложенного метода моделирования памяти Си-программ при использовании дополнительных семантических аннотаций.
- Схема доказательства полноты предложенного метода автоматического вывода дополнительных аннотаций при выполнении дополнительных семантических ограничений на используемое подмножество языка Си.

Новизна результатов подтверждается сравнением с известными результатами по тематике диссертационного исследования.

Практическая значимость результатов исследования

Методы моделирования памяти Си-программ, предложенные в диссертации, были реализованы в соответствующих инструментах верификации. Реализация метода моделирования памяти для предикатной абстракции в инструменте автоматической статической верификации CPAchecker позволила уменьшить число ложных срабатываний и расширить набор используемых правил корректности при анализе кода модулей ядра ОС Linux. Реализация модели памяти для инструментов дедуктивной верификации в инструменте Jessie позволила использовать его при верификации модуля безопасности для ядра ОС Linux. Разработанные программы входят в состав международных открытых проектов и могут использоваться в учебных, исследовательских и промышленных целях.

Достоверность и обоснованность научных положений и выводов

Достоверность полученных результатов подтверждается апробацией основных результатов работы на 5 научных конференциях и семинарах и публикацией в 10 научных трудах соискателя, в том числе 10 работ в журналах, рекомендованных ВАК Минобрнауки России.

Рекомендации по практическому использованию

Предложенные в работе методы представляется целесообразным использовать в проектах по верификации системного кода на языке Си, например, ядер операционных систем и встроенных систем, с использованием инструментов дедуктивной, а также автоматической статической верификации.

Замечания

1. В правиле вывода assign_deref_int на стр. 42 не введено обозначение $\cdot[\cdot,\ldots,\cdot]$. Также не введено обозначение $\hat{+}$, используемое на стр. 46.
2. При рассмотрении примера моделей памяти на стр. 59, 60, а также на стр. 63, 64 и 65 делается упрощающее предположение об отсутствии целочисленных переполнений при выполнении находящихся на рассматриваемом пути арифметических операций. Делается ли такое же предположение и в соответствующих инструментах верификации?
3. В разделе 3.3 "Оптимизации" (стр. 107) перечислены оптимизации, предлагаемые для использования при реализации описанного в главе 3 метода. Какие из перечисленных оптимизаций были использованы в реализации, представленной к защите?
4. При описании базового языка в разд. 4.1.1 отдельно вводится операция сравнения указателей на равенство ($t_v ::= p_1 == p_2$), хотя непосредственно перед этим уже определена операция вычитания указателей ($t_v ::= p_1 - p_2$), также имеется операция сравнения числа с нулем **assert** ($v \diamond 0$) и **assume** ($v \diamond 0$), позволяющая проверить результат вычитания указателей на равенство нулю. С учетом того, что предложенный базовый язык является минималистичным, операция сравнения указателей на равенство кажется избыточной. В работе не объясняется, зачем вводится такая избыточность.
5. В разделе 4.4 "Анализ регионов для базового языка с поддержкой вложенности" на стр. 155 утверждается следующее: "На практике, однако, предлагаемые ограничения позволяют разработать соответствующий алгоритм анализа регионов на основе структуры непересекающихся множеств, помещающий все термы, для которых отсутствуют соответствующие ограничения, в разные регионы". В тоже время в работе не описан такой алгоритм и не даются ссылок на статьи, предлагающие подобные алгоритмы.

В целом отмеченные недостатки не имеют принципиального значения, теоретическое исследование и разработанные программы являются полезным вкладом в область верификации программного обеспечения.

Заключение

На основании изложенного считаем, что диссертационная работа Мандрыкина Михаила Усамовича удовлетворяет требованиям ВАК Минобрнауки РФ, предъявляемым к кандидатским диссертациям, соответствует требованиям пункта 9 «Положения о

присуждении учёных степеней», утверждённого постановлением Правительства РФ от 24 сентября 2013 г. №842, а ее автор, Мандрыкин М. У. заслуживает присуждения учёной степени кандидата физико-математических наук по специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

Диссертация рассмотрена на семинаре кафедры Системного программирования Санкт-Петербургского государственного университета 14 ноября 2016 года и рекомендована к защите.

Отзыв заслушан и обсуждён на заседании кафедры Системного программирования СПбГУ, протокол № 38 от 18 ноября 2016 года.

Отзыв подготовили:

Доцент кафедры системного программирования, доктор технических наук, доцент Кознов Д.В., d.koznov@spbu.ru.

Доцент кафедры системного программирования, кандидат физико-математических наук, Булычев Д.Ю., dboulytchev@gmail.com.

Профессор, д. ф.-м.-н., профессор с возложенными обязанностями зав. кафедрой Системного программирования	Терехов А.Н.
Доцент, д. т. н., доцент кафедры Системного программирования	Кознов Д.В.
К. ф.-м.н., доцент кафедры Системного программирования	Булычев Д.Ю.

Сведения об организации.
ФГБОУ ВО «Санкт-Петербургский государственный университет»
199034, Санкт-Петербург, Университетская наб., 7/9. Сайт: www.spbu.ru
Кафедра Системного программирования СПбГУ, сайт http://se.math.spbu.ru/SE

Документ подготовлен в порядке исполнения трудовых обязанностей