ОТЗЫВ ОФИЦИАЛЬНОГО ОППОНЕНТА НА ДИССЕРТАЦИЮ
ГРЕЧАНИКА Сергея Александровича
"Доказательство свойств функциональных программ методом насыщения равенствами"
представленную на соискание учёной степени кандидата физико-математических наук
по специальности 05.13.11 - математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

1. Актуальность темы

Преобразование, анализ и оптимизация программ являются ключевыми аспектами системного и теоретического программирования. Несмотря на то, что исследования в этой области имеют давнюю историю и многие вопросы — теория вычислимости, схематология, семантика и другие — хорошо изучены, практика ставит всё новые задачи, которые требуют осмысления и теоретического обоснования. В качестве одного из примеров таких задач можно привести следующее. Традиционно в формальных вычислительных моделях предполагается, что каждая функция имеет ровно одно определение. Однако, скажем, многие компиляторы создают несколько версий функции, используемых в зависимости от условий применения. Аналогично, при поливариантных смешанных вычислениях, суперкомпиляции, рефакторинге и других методах преобразования программ, цели которых могут быть весьма различны, появляются альтернативные варианты одной и той же функции. Именно этот феномен является центральным вопросом в рассматриваемой работе. Предложенная и теоретически обоснованная в работе модель вычислений легла в основу системы автоматического доказательства теорем, что может быть использовано для верификации программ, важность которой очевидна ввиду растущих требований к надёжности и безопасности программного обеспечения. С этой точки зрения, несмотря на то, что исследования носят теоретический и фундаментальный характер, диссертация С.А.Гречаника имеет и непосредственный выход в практику, что подтверждает **актуальность** работы

2. Содержание работы

Диссертация состоит из введения, шести глав, списка использованной литературы и трёх приложений.

Во введении даётся общий контекст исследований. Обосновывается выбор функциональных языков как предмета исследования, а также кратко излагаются те идеи, которые послужили непосредственной отправной точкой. Во-первых, это метод насыщения равенствами Тейта и др., изначально предложенный как способ эффективного представления программ в рамках оптимизирующего компилятора, позволяющий оперировать не отдельными конструкциями программы, а целыми классами конструкций, полученных одна из другой в результате применения эквивалентных преобразований. Во-вторых, это различные методы суперкомпиляции (в частности, так называемая многорезультативная суперкомпиляция), давно и успешно развиваемые в секторе анализа и преобразования программ отдела инструментального и прикладного программного обеспечения ИПМ им. Келдыша, в котором работает автор. В качестве основной цели выбирается и обосновывается доказательство эквивалентности функций.

Первая глава посвящена обзору смежных работ. Автор включил в него либо основополагающие работы (как, например, работа Бёрстала и Дарлингтона) по трансформационному подходу, либо

работы, непосредственно связанные с данным исследованием: о насыщении равенствами, о системах автоматического доказательства и о суперкомпиляции. Очевидно, что ввиду обширности общей тематики преобразования программ этот обзор вряд ли может быть полным, и наверняка можно найти примеры работ, где что-то похожее делалось, но несколько иначе или в другом контексте. Однако, сделанный обзор даёт достаточно полное и точное представление о текущем состоянии исследований.

Вторая глава также является по существу вводной. Здесь определяется формальная нотация для записи функций, кортежей, вхождения выражений и т.п. Затем вводится "традиционный" язык первого порядка с конструкторами и сопоставлением с образцом. Простота языка отнюдь не означает какого-либо содержательного упрощения — язык универсальный и является базовым для широкого класса функциональных языков программировани. В частности, отмечается, что отсутствие функций высшего порядка может быть компенсировано применением известных методов. Далее во второй главе вводится понятие полипрограммы, состоящее просто в снятии ограничения на количество определений для конкретной функции. Нетривиальность такого допущения демонстрируется большим количеством примеров, анализ которых приводит к заключению, что необходимо существенно изменить семантику языка и вместо рассмотрения единственной наименьшей неподвижной точки следует рассматривать все возможные интерпретации функциональных символов, что делает формализм близким к логике предикатов первого порядка. Новое понимание семантики даёт возможность формально определить понятие преобразования программы как замену одного подмножества определений на другое. Далее приводятся примеры правил преобразования и неформально обосновывается их корректность. Наиболее сильным является слияние по бисимуляции, которое требует глобального анализа программы и может одновременно затрагивать большое подмножество определений. Следует отметить, что целью данной главы является не столько формальное определение и доказательство корректности, сколько то, чтобы у читателя сложилось общее впечатление о предлагаемом методе. Всё изложенное в этой главе будет переформулировано в последующих главах в рамках ещё более жёсткого, но более эффективного как с теоретической, так и практической точки зрения формализма.

Третья глава начинает формальное изложение с упрощения конструкций входного языка. Во-первых, в языке исключаются сложные выражения, поскольку любое сложное выражение можно разбить на части, используя дополнительные определения. Это даёт возможность исключить из рассмотрения понятие вхождения одного выражения в другое, что существенно упрощает дальнейшую формализацию. Кроме того, тут же выполняется выделение общих подвыражений, что даёт возможность не анализировать их многократно. Во-вторых, вводится понятие перестановки параметров, которое помогает решить ряд проблем. Так, исследуемые далее методы определения эквивалентности функций естественным образом распространяются на эквивалентность с точностью до перестановки параметров. Также оно автоматически покрывает традиционные преобразования введения неиспользуемых параметров (повышение арности) и удаление неиспользуемых (понижение арности). Оказывается, что перестановки параметров можно "просачивать" по программе и комбинировать, избегая тем самым необходимости введения большого количества вспомогательных функций и специальных правил работы с ними. Избежать комбинаторного роста при работе с перестановками позволяет наличие канонической формы.

Завершается третья глава доказательством того, что любую программу из исходного языка можно преобразовать к такой форме.

В четвёртой главе рассматриваются локальные преобразования программ, то есть такие, которые могут быть применены к нескольким связанным друг с другом определениям без знания глобального контекста. Здесь становятся наглядными достоинства предложенного в третьей главе формализма. Так, определения всех 12 упрощающих правил умещаются на одной странице. Большая часть этой главы состоит из формального доказательства корректности этих правил и того важного утверждения, что процесс их применения всегда завершается. Вторая группа локальных правил — насыщающие правила — также корректны, но могут привести к неограниченному росту программы и, следовательно, их применение должно дополнительно контролироваться стратегией применения.

Пятая глава, видимо, наиболее сложная с теоретической точки зрения, описывает средства для работы с рекурсивными определениями. Для этого используется метод бисимуляции, по-существу схожий с суперкомпиляцией: он пытается выяснить эквивалентность двух функций путём построения развёртки программы до тех пор, пока не найдёт противоречия или не достигнет ограничения на глубину развёртки, либо процесс остановится ввиду обнаружения циклов на всех ветвях. В последнем случае функции объявляются эквивалентными.

Шестая глава посвящена реализации и сравнению с другими системами. Собственно о реализации говорится весьма немного, поскольку проведённая формализация была исходно нацелена на поиск эффективного внутреннего представления программ. Для того, чтобы описанная система преобразований воплотилась в систему доказательства эквивалентности, осталось только определить стратегию применения преобразований, что, собственно, и делается в этой главе. Далее реализованная система, названная Graphsc, сравнивается с другими известными системами. Показывается, что Graphsc находится примерно на том же уровне. Детально разбираются случаи, в которых Graphsc проигрывает.

3. Новизна результатов и выводов

Данная работа несомненно вносит существенный вклад в теорию программирования. Из текста диссертации можно заключить, что в при решении поставленной задачи автором были получены следующие новые научные результаты.

1. Определено понятие полипрограммы, как программы, допускающей множественность определений одной и той же функции.
2. Предложена семантика полипрограмм, основанная на совпадении множеств подтверждающих интерпретаций.
3. Предложена система преобразования полипрограмм, для которой доказана корректность и изучены вопросы завершаемости и конфлюентности.
4. Реализована система доказательства эквивалентности функций.

Результаты были представлены на большом количестве научных семинаров и конференций, включая международные.

Считаю, что **наиболее важными** являются первые три из вышеперечисленных пунктов, поскольку система автоматического доказательства, хотя и важный, но лишь один из возможных инструментов, которые можно построить на разработанной инструментальной базе. Говоря "коммерческим" языком,"здесь предлагается платформа, а не отдельное решение".

4. Значимость для науки и производства

Проведённые диссертантом исследования представляют средства анализа и преобразования программ и могут быть использованы для создания широкого спектра инструментов, поддерживающих надёжное программирование. Кроме этого, разработанная в рамках диссертационной работы система автоматического доказательства эквивалентности может быть использована как вспомогательный инструмент в различных системах верификации программ — важного средства обеспечения **информационной безопасности**. По-видимому, результаты диссертации могут быть успешно использованы в таких коллективах как ИПМ РАН, ИСИ СО РАН, ВЦ РАН, СпбГУ, а также во многих других организациях, разрабатывающих программное обеспечение.

5. Замечания

1. Первое замечание скорее можно отнести к достоинствам работы: многие проблемы (например, вопрос о конфлюэнтности упрощающих правил) остались открытыми, и автор явно это отметил. Остались вне рассмотрения вопросы полноты, т.е. описания класса программ, для которых эквивалентность будет разрешима с помощью предложенных правил, а также сложности процесса доказательства, которая может зависеть не только от размеров программы, но и, скажем, от ограничений на глубину поиска бисимуляции. Следует признать, что это не влияет на достоверность результатов диссертации, поскольку все необходимые утверждения формально доказаны.
2. Хотя в целом диссертация написана вполне хорошим языком, некоторые термины, такие как "прувер", "солвер", "инстанцируют", режут слух. Некоторые обороты являются калькой с английского, такие как "человеко-читаемая программа", "свежая переменная", и без труда могут быть улучшены. Также встречаются жаргонизмы, типа "движок". Представляются также неудачными термины "обобществление выражений", "соседние определения", "разворот циклов".
3. Иногда встречаются не объявленные ссылки вперёд по тексту. Так на стр. 69 используется обозначение unshared, которое определяется на рисунке 3.7, расположенном на следующей странице.
4. В тексте имеется некоторое количество пунктуационных ошибок и несколько орфографических.
5. К содержательным недостаткам можно отнести следующие: На стр 30 предлагается "добавить определение f(x) = bottom". Формально это неверно, поскольку bottom является элементом семантической области, а не синтаксической конструкцией. Возможно, соответствующий элемент следовало внести непосредственно в язык для обозначения никогда не завершающихся вычислений.
6. В разделе 2.3.1 на стр 39 основная задача преобразования программ сформулирована неудачно, поскольку решить её в такой постановке можно всегда, взяв в качестве результата

исходную программу. В определении должно быть указано наличие цели преобразования или требований к результату.

7. На стр. 44 в разделе "Понижение арности" отношение включения должно быть направлено в обратную сторону.
8. В таблице 4.1 на стр. 92 в строке (red-case-cons) должно быть Call(h,g) в правой части импликации.
9. Неясно, что означает характеристика "недостаточно фундаментальны" для насыщающих правил, поскольку они не выразимы через упрощающие правила.
10. В первом пункте определения несовместимых функций на стр. 126 следует уточнить, что C не равно D.
11. Неудачно оформлен первый пример в разделе 5.3.1 на стр 137, поскольку не определена функция f(x).

6. Заключение

Автореферат правильно отражает содержание диссератации: строго формулирует постановку задачи исследования и основные полученные результаты. Автореферат соответствует требованиям, предъявляемым ВАК к кандидатским диссертациям.

На основании материалов диссертации считаю, что диссертация Гречаника Сергея Александровича "Доказательство свойств функциональных программ методом насыщения равенствами" является законченным научно-квалификационным исследованием, решающим важную задачу теоретического и системного программирования, и удовотлетворяет требованиям Положения о порядке присуждения научных степеней, а её автор заслуживает присуждения учёной степени кандидата физико-матматических наук по специальности 05.13.11 "Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей".

Заведующий лабораторией смешанных вычислений Бульонков Михаил Алексеевич
Института систем информатики СО РАН
кандидат физико-математических наук
(специальности 05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей)

Сведения об организации:
Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук
630090, Российская Федерация, г. Новосибирск, проспект Академика Лаврентьева, 6.
тел. (383) 3308652, факс (383) 3323494,
вебсайт: www.iis.nsk.su,
электронная почта: mike@iis.nsk.su.

Личную подпись заверяю
Нач. отдела кадров Верейко
01.02.2018г.