

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Языки программирования и компиляторы — 2017

Труды конференции
(3–5 апреля 2017 г.)

Под редакцией Д. В. Дуброва

Ростов-на-Дону
2017

УДК 004.4(063)
ББК 32.973я43
Я411

*Конференция проводится при финансовой поддержке
ООО «СиПроВер» (генеральный директор Е. А. Рыжков), разработчика
статического анализатора кода PVS-Studio для поиска ошибок
в программах на C, C++ и C# на Windows и Linux
(<https://www.viva64.com>)*

*Сборник издан при поддержке Российского фонда фундаментальных
исследований (проект 17-01-20061)*

Я411 **Языки программирования и компиляторы — 2017 :**
труды конференции / Южный федеральный университет ;
под ред. Д. В. Дуброва. — Ростов-на-Дону : Издательство
Южного федерального университета, 2017. — 282 с.
ISBN 978-5-9275-2349-8

В сборнике представлены труды конференции «Языки программирования и компиляторы» (Южный федеральный университет, Институт математики, механики и компьютерных наук им. И. И. Воровича, 3–5 апреля 2017 г.).

Конференция посвящается памяти известного ростовского учёного Адольфа Львовича Фуксмана, основоположника технологии вертикального слоения программ, заведующего Вычислительным центром РГУ с 1964 по 1978 годы.

УДК 004.4(063)
ББК 32.973я43

ISBN 978-5-9275-2349-8

© Южный федеральный университет, 2017

Краткая история суперкомпиляции в России

Климов А. В., klimov@keldysh.ru
Романенко С. А., romansa@keldysh.ru
Институт прикладной математики
им. М. В. Келдыша РАН*

Аннотация

Суперкомпиляция — это метод преобразования программ, предложенный В. Ф. Турчиным в 1970-е годы и далее развиваемый его учениками и последователями. В докладе дается краткое введение в суперкомпиляцию, справка о работах В. Ф. Турчина, подробнее о работах последних десятилетий в России, отмечены современные проблемы метавычислений и дальнейшие направления развития.

Ключевые слова: суперкомпиляция, специализация программ, верификация программ, метавычисления.

Введение *Суперкомпиляция* — это метод преобразования программ, предложенный Валентином Федоровичем Турчиным (1931-2010) в 1970-е годы, основанный на идее развертки графа путей вычисления программы (*прогонка*) и выполнения над ним операций, обеспечивающих в конце концов получение конечного графа, являющегося представлением остаточной программы, эквивалентной исходной (*конфигурационный анализ*). История работ В. Ф. Турчина с библиографией описана в статье [10]. Популярное изложение метода суперкомпиляции можно найти в работе [9].

*Работа поддержана грантом РФФИ № 16-01-00813-а.

Из-за ограниченности объема здесь не обозрываются многочисленные зарубежные работы по суперкомпиляции, приведен сокращенный список литературы и многие работы упоминаются без ссылок на публикации.

Суперкомпиляторы для языка Рефал В. Ф. Турчин начал разрабатывать суперкомпилятор для языка Рефал в середине 1970-х годов, пользуясь реализацией Рефала-2 на БЭСМ-6, но лишь после эмиграции в США смог опубликовать описание метода и завершить первые работающие версии суперкомпилятора Рефала. Библиография его работ приведена в [10].

С 1993 года дальнейшим развитием суперкомпилятора Рефала занимался А. П. Немытых. К началу 2000-х годов суперкомпилятор, названный SCP-4, приобрел законченный вид, и А. П. Немытых подготовил и опубликовал его подробное описание [14].

Исследовательские и экспериментальные суперкомпиляторы Параллельно шел поиск более простых систем метавычислений и суперкомпиляторов. Первой такой системой был реализованный С. А. Романенко в конце 1970-х годов простой специализатор для Рефала-2. Его принципы были позднее описаны в [11].

В начале 1990-х Анд. В. Климов и R. Glück выделили ядро суперкомпиляции для модельного функционального языка [1]. Затем С. М. Абрамов описал на Haskell'e другие компоненты суперкомпиляции, включая *окрестностый анализ*, и на его основе предложил метод *окрестностного тестирования*, а также реализацию *универсального решающего алгоритма* и методику *инверсного программирования* на основе суперкомпиляции [7].

Линию простых суперкомпиляторов продолжил И. Г. Ключников расширив суперкомпиляцию на языки с функциями высшего порядка [12].

Корректность суперкомпиляторов Первые подробные доказательства корректности отдельных алгоритмов метода суперкомпиляции были проведены С. М. Абрамовым [7]. Анд. В. Климов разработал формальные спецификации простых суперкомпиляторов с целью последующего использования их для доказательств корректности. Первое доказательство корректности простого суперкомпилятора в системе Coq, проверяемое на компьютере, разработал Д. Кръстев. Вслед за

ним И. Г. Ключников и С. А. Романенко реализовали альтернативный метод проверки на компьютере корректности суперкомпиляции, назвав его *сертифицирующей суперкомпиляцией*: вместе с остаточной программой суперкомпилятор порождает доказательство ее эквивалентности исходной на входном языке системы Coq [4].

Формальные свойства «свистков» (правил останковки суперкомпилятора) исследовала Антонина Н. Непейвода [6].

Приложения к верификации программ Первый практический результат по верификации коллекции моделей протоколов с помощью SCP-4 получили А. П. Немытых и А. П. Лисица [13]. Впоследствии эти эксперименты были повторены с помощью Java-суперкомпилятора [2] и простого суперкомпилятора, запрограммированного на языке Scala. Анализируя их, Анд. В. Климов определил компактный предметно-ориентированный суперкомпилятор для счетчиковых систем переходов [3].

И. Г. Ключников успешно применил суперкомпилятор функционального языка высшего порядка [12] для проверки эквивалентности программ и использовал его для построения *двухуровневого суперкомпилятора*.

Дальнейшего повышения мощности и эффективности проверки эквивалентности программ добился С. А. Гречаник путем синтеза методов *многорезультатной суперкомпиляции и насыщения равенствами* [8].

Многорезультатная суперкомпиляция В суперкомпиляции, как и во многих методах оптимизации программ, много степеней свободы. Вместо изобретения эвристик можно перебирать всевозможные решения, порождая множество остаточных программ, и выбирать из них «лучшую» по некоторому критерию или использовать их все для каких-то задач. Отсюда возникла идея и реализация *многорезультатной суперкомпиляции* [5].

Оказалось, что успех в применении суперкомпиляторов для верификации протоколов использовал многорезультатность путем варьирования эвристик руками, а специализированный алгоритм для счетчиковых систем [3] проводит автоматический частичный перебор с учетом специфики данного класса задач.

Суперкомпилятор языка Java В 1999–2003 годах Анд. В. Климов, Арк. В. Климов и А. Б. Шворин разработали суперкомпилятор JScp для языка Java (актуальной в то время версии 1.4) [2]. Прогонка была проработана достаточно глубоко с целью сохранения как можно большей информации о значениях ссылочных переменных. Конфигурационный анализ был упрощен: он обрабатывал только явные циклы в программе, а рекурсивными функциями не занимался. Всевозможные степени свободы в стратегиях суперкомпиляции были выведены на пользователя в виде опций.

Основная проблема, с которой столкнулись попытки практического применения Java-суперкомпилятора, — необходимость вручную подбирать опции стратегий суперкомпиляции, изучая остаточный код и догадываясь, как прошел процесс суперкомпиляции и как его направить по-другому.

Направления будущих работ

- разработка упрощенных суперкомпиляторов — как демонстрационных для теоретических исследований и преподавания, так и практических проблемно-ориентированных;
- реализация суперкомпиляторов для реальных функциональных языков (Haskell, SML и т.п.);
- перенос достижений с функциональных языков на практические объектно-ориентированные типа Java;
- теория суперкомпиляции и построение доказательств важных свойств суперкомпиляторов (особенно — проверяемых на компьютере), методы верификации суперкомпиляторов, сертифицирующей суперкомпиляции;
- приложения суперкомпиляции к верификации программ;
- многоуровневая суперкомпиляция;
- многорезультатная суперкомпиляция;
- суперкомпиляция с насыщением равенствами;
- интерактивные метавычисления.

Поясним последний пункт. Мы пришли к убеждению, что автоматические методы преобразования и оптимизации программ подошли некоторому пределу, барьеру сложности. Многорезультатность и использование мощных компьютеров для перебора может несколько отодвинуть эту границу. Тем не менее, практическое применение суперкомпиляции и других методов метавычислений требует вовлечения человека и разработки систем с удобным человеко-машинным интерфейсом, дающим программисту информацию в понятном ему виде о том, как его программа анализируется и преобразуется, что получается на выходе и как связан остаточный код с исходным. Только сам программист знает, с какой целью он занимается преобразованиями своих программ и какие аспекты стоит оптимизировать, а какие не надо. Такие средства должны быть не слабее широко используемых диалоговых отладчиков. Это направление работ мы называем *интерактивными метавычислениями*.

Список литературы

1. *Glück R., Klimov A. V.* Occam's razor in metacomputation: the notion of a perfect process tree // *Static Analysis. Lecture Notes in Computer Science*. Т. 724. — Springer, 1993. — С. 112–123.
2. *Klimov A. V.* A Java Supercompiler and its application to verification of cache-coherence protocols // *Perspectives of Systems Informatics. PSI 2009. Lecture Notes in Computer Science*. Т. 5947. — Springer, 2010. — С. 185–192.
3. *Klimov A. V.* A simple algorithm for solving the coverability problem for monotonic counter systems // *Automatic Control and Computer Sciences*. — 2012. — Т. 46, № 7. — С. 364–370.
4. *Klyuchnikov I. G., Romanenko S. A.* Certifying supercompilation for Martin-Löf's type theory // *Perspectives of System Informatics. PSI 2014. Lecture Notes in Computer Science*. Т. 8974. — Springer, 2015. — С. 186–200.
5. *Klyuchnikov I. G., Romanenko S. A.* Multi-result supercompilation as branching growth of the penultimate level in metasystem transitions // *Perspectives of Systems Informatics. PSI 2011. Lecture Notes in Computer Science*. Т. 7162. — Springer, 2012. — С. 201–226.

6. *Непеивода А. Н.* Turchin's relation for call-by-name computations: a formal approach // Electronic Proceedings in Theoretical Computer Science. — 2016. — Т. 216. — С. 137–159.
7. *Абрамов С. М.* Метавычисления и их приложения. — М.: Наука, 1995.
8. *Гречаник С. А.* Доказательство свойств функциональных программ методом насыщения равенствами // Программирование. — 2015. — № 3. — С. 44–61.
9. *Климов А. В.* Введение в метавычисления и суперкомпиляцию // Будущее прикладной математики: Лекции для молодых исследователей. От идей к технологиям. — М.: КомКнига, 2008. — С. 343–368.
10. *Климов А. В.* О работах Валентина Федоровича Турчина по кибернетике и информатике // Труды SORUCOM-2011. — 2011. — С. 149–154.
11. *Климов А. В., Романенко С. А.* Метавычислитель для языка Рефал. Основные понятия и примеры: Препринт / М.: ИПМ им. М.В. Келдыша АН СССР. — 1987. — № 71.
12. *Ключников И. Г.* Суперкомпиляция функций высших порядков // Программные системы: теория и приложения. — 2010. — Т. 4, № 4. — С. 37–71.
13. *Лисица А. П., Немытых А. П.* Верификация как параметризованное тестирование (эксперименты с суперкомпилятором SCP4) // Программирование. — 2007. — № 1. — С. 22–34.
14. *Немытых А. П.* Суперкомпилятор SCP4: общая структура. — М.: Эдиториал УРСС, 2007.