



Метавычисления



Направления исследований

- специализация программ по контексту использования;
- суперкомпиляция;
- методы анализа и оптимизации программ;
- методы автоматической верификации вычислительных систем;
- методы автоматизации отладки и тестирования;
- методы программного моделирования недетерминированных и инверсных вычислений;
- методы автоматической генерации компиляторов языков программирования из интерпретаторов этих языков;
- окрестностное тестирование программ;
- нестандартные семантики;
- нестандартные диалекты языков программирования;
- методы распараллеливания программ;
- реализация функциональных и логических языков программирования;
- аналитические вычисления и компьютерная алгебра.

Основные цели специализации программ

Технология программирования развивается в сторону оперирования понятиями задачи, которая стоит перед программистом, а не понятиями универсального прибора, на котором программа будет исполняться. Это стимулирует развитие языков программирования высокого уровня позволяющих адекватно отражать объектную область задачи. С другой стороны, аппаратная реализация современных широко используемых ЭВМ поддерживает фоннеймановскую модель вычислений; что приводит к неэффективной реализации таких языков — посредством интерпретации — более того, часто не прямой, а косвенной — через другую интерпретацию. К подобной неэффективности приводит и любое структурное программирование само по себе; ибо его целью является создание гибких, легко понимаемых и изменяемых программ. Всё чаще программы вычисляются другими программами, а потому естественно ожидать, что первые будут содержать простейшие структуры, ведущие к накладным расходам, которые никогда бы не допустил квалифицированный программист.

Методы автоматической специализации структурированных программ высокого уровня призваны предоставить свободу развития новым технологиям программирования.

Суперкомпиляция

Суперкомпиляция есть набор методов автоматической специализации программ. Основным механизмом суперкомпиляции — метаинтерпретация. Основополагающие идеи суперкомпиляции, как и сам термин, были предложены В.Ф. Турчиным в 70-х годах двадцатого века.

Суперкомпилятор SCP4

- первый реально работающий свободно распространяемый экспериментальный суперкомпилятор:
 - <http://www.botik.ru/pub/local/scp/refal5/>
 - демонстрация SCP4 доступна непосредственно через Интернет (в режиме on-line);
- название SCP4 отражает историю попыток построения суперкомпиляторов;
- разработан совместно с В.Ф. Турчиным, The City University of New York;
- специализирует программы на функциональном языке программирования РЕФАЛ-5
 - входной язык является реальным языком программирования;
- реализован на языке РЕФАЛ-5;
- может быть использован для верификации недетерминированных параметризованных распределенных вычислительных систем;
- принципы реализации описаны в монографии
 - А.П. Немытых «Суперкомпилятор SCP4: общая структура», издательство УРСС, г. Москва, 2007 г.



```
<SCP4 ..... ( )>
  <Int   e.data ( )> SourceInt
    <Go  !   >   Prog

MST-схема, описывающая использование суперкомпилятора
как компилятора из языка, семантика которого задана
интерпретатором Int, в язык программирования РЕФАЛ-5.
```

Верификация

К области метапрограммирования относится и автоматическое доказательство корректности вычислительных систем. Вычислительные системы, в отличие от «обычных» программ, могут быть недетерминированными и работать бесконечно долго. Под корректностью здесь понимается выполнение какого-то конкретного свойства вычислительной системы, которое, по тем или иным причинам, является важным для пользователя.

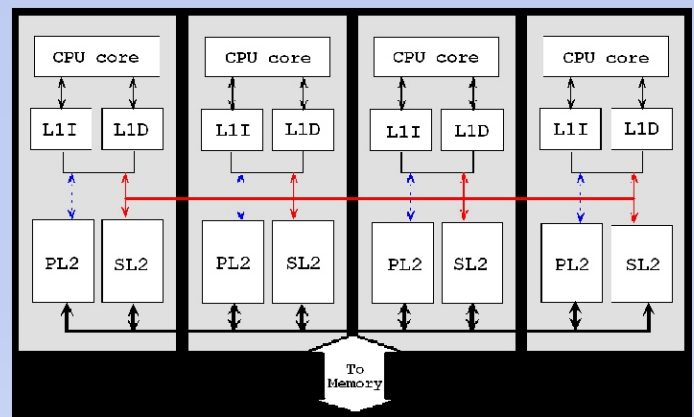


Верификация посредством суперкомпиляции

- Совместно с Лабораторией верификации Ливерпульского университета (University of Liverpool) разработан метод верификации параметризованных вычислительных систем посредством суперкомпиляции их программных моделей:
 - в параметризованной вычислительной системе количество участников взаимодействия в процессе эволюции системы конечно, но не известно сколько;
 - свойства корректности формулируются в терминах недостижимости определенного вида глобальных состояний рассматриваемой вычислительной системы;
 - суперкомпилятор специализирует программную модель этой вычислительной системы по свойствам начальных состояний системы;
 - удачная верификация модели выражается в преобразовании скрытой семантической информации об интересующем нас свойстве модели в явную синтаксическую информацию результата преобразований программной модели суперкомпилятором;
 - метод может быть охарактеризован также как параметризованное тестирование.
- Посредством суперкомпилятора SCP4 верифицированы:
 - параметризованные протоколы корректности когерентности кэша: Synapse N+1, MSI, MOSI, MESI, MOESI, The University of Illinois, Berkley, DEC Firefly, IEEE Futurebus+;
 - несколько параметризованных многопоточных моделей Java программ, описанных в терминах однопоточных моделей;
 - ряд свойств распределенных вычислительных систем, описанных в терминах сетей Петри.



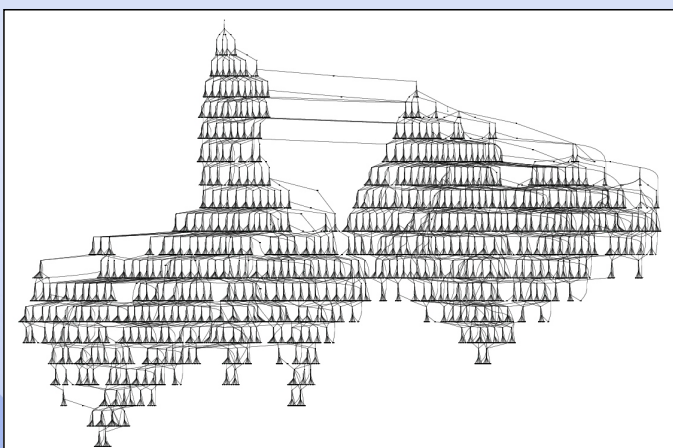
- при интерактивном использовании суперкомпилятора SCP4 найдены ошибки и построены тесты, указывающие на эти ошибки, в:
 - некорректной спецификации Xerox PARC Dragon протокола когерентности кэша;
 - некорректной спецификации абстрактной модели протокола Inc/Dec (Atomic Increments and decrements on a Point object) взаимодействия нитей в многопоточных Java программах;
 - спецификации SPS2 протокола когерентности кэша.



SPS2 snooping cache coherence protocol for CMP

Функциональные и логические языки

- Реализованы: Микро-Пролог, FLAC (диалект РЕФАЛа, ориентированный на компьютерную алгебру), РЕФАЛ Плюс.
- Поддерживаются: РЕФАЛ-5, РЕФАЛ Плюс.



Граф автоматического доказательства корректности одной из моделей взаимодействия нитей в многопоточных Java программах

