

Графическое отображение некоторых понятий суперкомпиляции (трассировщик суперкомпилятора SCP4) (Тезисы доклада)

Андрей П. Немытых

Институт программных систем РАН
nemytykh@math.botik.ru

Суперкомпиляция есть метод специализации программ. Суперкомпилятор SCP4 был разработан и реализован в конце 1990-х – начале 2000-х гг. для функционального языка программирования рефал-5. Для данной программы p и её частично определенной входной точки $f(x_0, y)$ суперкомпилятор разворачивает потенциально бесконечное дерево всех возможных вычислений p , начинающихся в $f(x_0, y)$, анализирует и оптимизирует это дерево и сворачивает его в граф, который представляет результат суперкомпиляции.

В рамках поддержки реализации и развития суперкомпилятора SCP4, совершенствования технологии суперкомпиляции автором разработаны принципы построения графического трассировщика суперкомпилятора SCP4 и реализована его первая версия. В данной статье описываются принципы реализации этого трассировщика и демонстрируется часть его инструментария.

Суперкомпилятор SCP4 реализован на функциональном аппликативном языке программирования рефал-5 и является оптимизатором программ, написанных на этом же языке, основанным на технологии суперкомпиляции, основополагающие идеи которой были заложены В. Ф. Турчиным. Суперкомпиляция выполняет метаинтерпретацию исходных программ на их частично заданных входных данных. Стартовая версия экспериментального суперкомпилятора SCP4 была разработана в конце 1990 годов автором данной статьи – под научным руководством В. Ф. Турчина. Руководство пользователя суперкомпилятора SCP4 было написано в 1999 г. А. В. Корлюковым. Первые публикации по SCP4 появились в начале 2000-х. В 2007 году А. П. Немытых опубликовал монографию «Суперкомпилятор SCP4: Общая структура». Название – SCP4, отражающее историю суперкомпиляции, было предложено В. Ф. Турчиным. Одна из старых версий суперкомпилятора SCP4 может быть вызвана в режиме on-line с электронной страницы <http://www.botik.ru/pub/local/scp/refal5/>. Позднее посредством SCP4 был произведен ряд удачных экспериментов по верификации некоторого класса недетерминированных параметризованных коммуникационных протоколов с условиями глобальной корректности. На данный момент суперкомпилятор SCP4 является единственным экспериментальным суперкомпилятором для реального языка программирования рефал-5 (без каких-либо ограничений на язык); в то же время существуют несколько простых суперкомпилято-

ров для модельных языков программирования или простейших фрагментов реальных языков программирования.

Процесс суперкомпиляции автоматически развивает и анализирует потенциально бесконечное дерево всех возможных вычислений исходной программы P в контексте задачи Z на специализацию программы P , поставленной суперкомпилятору SCP4. При этом, как исходная задача Z , так и промежуточные задачи, возникающие перед суперкомпилятором, могут автоматически разлагаться на композицию подзадач. То есть, суперкомпилятор работает со сложной структурой – лесом деревьев, помеченных информацией о параметризованных состояниях в точках ветвления этих деревьев и условиями выбора ветвей при конкретном вычислении исследуемой программы P . Задача суперкомпилятора эффективно свернуть этот лес потенциально бесконечных деревьев в конечный граф, который и будет представлять результат суперкомпиляции. Линейно-текстовое представление такой сложной структуры данных крайне трудоемко для анализа процесса суперкомпиляции и, как правило, сводится к ручному рисованию этой структуры. Разработка графического отображения трассы вычисления суперкомпилятора является принципиальным шагом как для развития и совершенствования самой технологии суперкомпиляции, как таковой, так и для расширения областей применения суперкомпиляции. Анализ графического представления трассы суперкомпилятора SCP4 позволяет повысить эффективность: процесса отладки суперкомпилятора SCP4 при его дальнейшем развитии; улучшения существующих алгоритмов суперкомпиляции; генерации новых идей, которые должны позволить существенно расширить границы применения суперкомпилятора SCP4; процесса обучения студентов основам технологии суперкомпиляции.

Качество остаточных программ¹ существенно зависит от стратегий, выбранных для конкретного процесса суперкомпиляции. Выбор наиболее сильных с точки зрения преобразований стратегий может привести: к большому времени исполнения этого процесса для некоторых исходных (входных) программ; более того, к незавершаемости процесса суперкомпиляции. В суперкомпиляторе SCP4 реализованы несколько уровней подробности трассировки, предоставляющие возможности изучения алгоритмов суперкомпиляции с целью их улучшения или другими целями (например, для извлечения формального доказательства в случае удачной попытки верификации или для изучения причин: почему попытка верификации оказалось неуспешной).

Ранее графическая поддержка отображения трассы суперкомпиляции была реализована И.Г. Ключниковым в простейшем модельном суперкомпиляторе SCPS для простейшего подмножества языка Scala.

1 Принципы реализации графического трассировщика

Говоря точнее, описываемый нами трассировщик является графически-текстовым. Введено понятие шага графического трассировщика, которое со-

¹ Программ поученных в результате суперкомпиляции.

ответствует логическому шагу изменения ориентированного графа, анализируемого и преобразуемого суперкомпилятором. На каждом шагу трассировщика генерируется графическое описание состояния суперкомпилятора в языке описания графов Dot, разработанном компанией AT&T. Язык Dot является абстрактным языком, независимым от конкретного графического формата. Преобразователь состояний суперкомпилятора SCP4 в язык Dot реализован, как и сам SCP4, на языке рефал-5. Далее используется свободно распространяемая система Graphviz, которая позволяет получить из данного Dot-описания конкретную картинку, поддерживает все широко-распространенные графические форматы. Система Graphviz вызывается непосредственно из суперкомпилятора. Масштабирование и навигация по графам реализована с использованием свободно распространяемой системой ZGRViewer. Графические состояния текущего и предыдущего шага трассировщика отображаются на экране посредством браузера² в двух соседних окнах, что позволяет пользователю проанализировать результат исполнения текущего шага суперкомпилятора. Для отображения свойств узлов и ребер графа состояний суперкомпилятора используются как текстовая информация, так и цветовая. Управление трассировщиком, в частности, осуществляется посредством текстовой консоли, которая также, при необходимости, дает возможность более детального изучения состояния суперкомпилятора SCP4 в терминах внутренних структур (кодировок) суперкомпилятора. Графическое Dot-описание состояния текущего шага трассировщика строится на основании уже имеющегося Dot-описания состояния предыдущего шага трассировщика, – заменой описания изменившихся частей рисунка: генерация заново всего описания графического состояния не производится.

В полной версии данной статьи показано отображение ряда понятий суперкомпиляции средствами графического трассировщика суперкомпилятора SCP4. В данных тезисах ограничимся графическим отображением результата прогонки и дадим краткие пояснения к нему.

2 Пример (результат прогонки)

Пусть дан некоторый язык программирования L . Любой интерпретатор `Int` языка L обычно имеет следующую структуру.

```
Int(p,d) {
  инициализация;
  while (вычисления не завершены)
    { СТЕП; }
  return (результатирующее состояние);
}
```

Здесь СТЕП суть минимальный логически замкнутый шаг, включающий в себя все механизмы необходимые для интерпретации любой L -программы p

² Браузер может быть выбран пользователем.

на любых входных данных d . Скорее невозможно, чем трудно, реализовать какой-нибудь интерпретатор другим способом³. STEP изменяет глобальное состояние интерпретатора; спецификация этого состояния зависит от L . Если физическое время вычисления STEP равномерно ограничено по размеру его входных данных, тогда число шагов, необходимое для вычисления p на d , можно рассматривать как логическое время вычислений, которое *адекватно* отражает соответствующее физическое время вычислений.

Вернёмся к языку рефал. Одним из базовых механизмов суперкомпиляции является прогонка данной параметризованной конфигурации (параметризованного состояния стека вызовов функций специализируемой рефал программы). Алгоритм прогонки является метарасширением интерпретации одного шага рефал машины, преобразующего данное конкретное состояние стека вызовов (в терминах рефала – конкретного поля зрения) в следующее конкретное состояние стека вызовов. Результатом прогонки является конечное дерево возможных вычислений данного на входе прогонки параметризованного (т.е. частично неизвестного) стека вызовов функций. На рисунке 1 показан пример снимка экрана с результатом первого вызова алгоритма прогонки. Серые кластеры показывают структуры параметризованных стеков в листьях грозди прогонки. Зеленым цветом отмечены полностью вычисленные параметризованные части стеков (т.е. в них нет вызовов функций), соответствующих пути вычисления из корня грозди в данный лист грозди. Вершины грозди прогонки, обозначенные точками, не учитываются суперкомпилятором SCP4 при анализе «истории вычислений» – развертки метадрева возможных вычислений поставленной задачи на суперкомпиляцию. По этой причине параметризованное состояние вычислительной системы не показано в вершинах-точках. Остальные вершины называются опорными и могут в процессе анализа стать входными точками функций (подпрограмм) результата суперкомпиляции. Стартовая параметризованная конфигурация, задача на суперкомпиляцию, являющаяся корнем метадрева возможных вычислений, выделена указанием на неё стрелки-ребра, в исходящей вершине которого нет никакой конфигурации. Ребра грозди прогонки помечены предикатами выбора/расщепления конкретных значений параметров (выбора конкретного рефал шага в момент исполнения преобразуемой программы). Отметим, что, в отличие от интерпретации конкретного (не параметризованного) стека вызова функций, разложение синтаксической композиции вызовов функций в стек, линейную последовательность их исполнения, может быть нетривиальным и зависит от стратегии суперкомпиляции. Примерами таких стратегий могут быть «ленивое» (call by need) и аппликативное (call by value) вычисление в процессе суперкомпиляции. Интерпретатор рефала-5 является консольным приложением. В нижнем правом углу экрана находится окно консоли интерпретатора рефала-5, исполняющего суперкомпилятор SCP4, трасса которого (результат первого вызова прогонки) отображена в правом верхнем окне – это результат первого шага

³ Это следует из того, что операционная семантика языка L описывается текстом конечной длины.

графического трассировщика суперкомпилятора SCP4. Для получения следующей пары картинок необходимо отдать команду в командной строке на консоли рефал-5 интерпретатора.

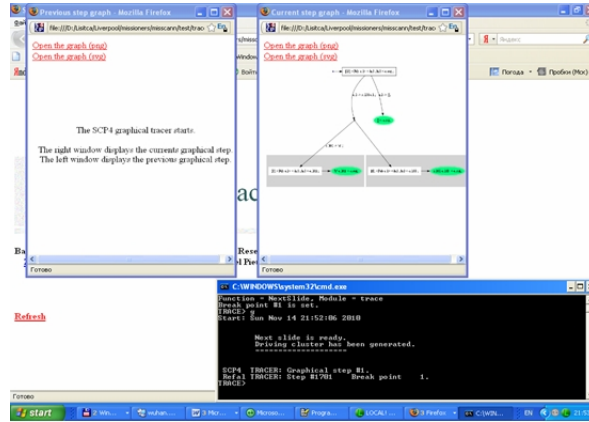


Рис. 1. Результат первого вызова алгоритма прогонки

Рисунок 2 показывает результаты первого и второго шагов трассировщика – левый и правый экраны соответственно. Результатом второго графического шага суперкомпилятора SCP4 является развитие дерева возможных вычислений задачи на суперкомпиляцию (пары – преобразуемой программы и ее параметризованной входной точки): результат второго вызова алгоритма прогонки был подклеен вместо листа левой ветви грозди прогонки, построенной предыдущим вызовом прогонки.

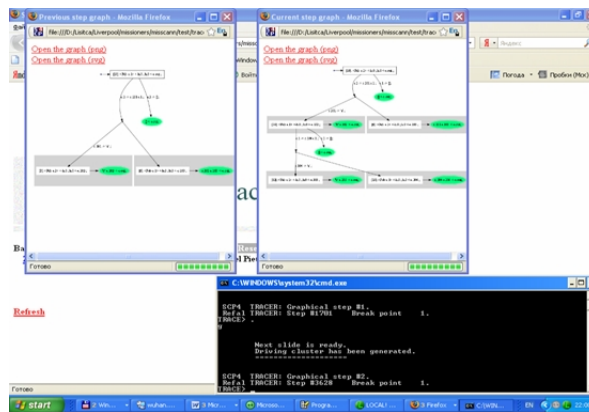


Рис. 2. Результаты первого и второго шагов трассировщика

Эти картинки дают хороший повод для пояснения – почему алгоритмы обобщения не были вызваны суперкомпилятором SCP4 между этими двумя вызовами прогонки. Конфигурация, которой помечен самый левый лист грозди прогонки, построенной первым вызовом прогонки, может быть транзитной и это свойство данной конфигурации есть предмет анализа второго вызова алгоритма прогонки. Другими словами, второй узел прогонки, возможно, построит единственную ветвь, исходящую из рассматриваемого листа. В этом случае представляется шанс для оптимизации исходной программы: два шага рефал машины могут быть соединены в один шаг. С другой стороны, если из узла грозди прогонки исходят несколько ветвей, тогда некоторые из них могут быть обрезаны последующими вызовами алгоритма прогонки. Таким образом, в данной точке алгоритма суперкомпиляции может работать стратегия, которая должна решать: сколько последовательных вызовов прогонки должны быть выполнены между двумя последовательными вызовами алгоритма свёртки.

В левых верхних углах окон шагов графического трассировщика красным шрифтом обозначены ссылки, позволяющие увеличить представленный граф (в png или svg формате). На рисунке 3 показан результат такого увеличения.

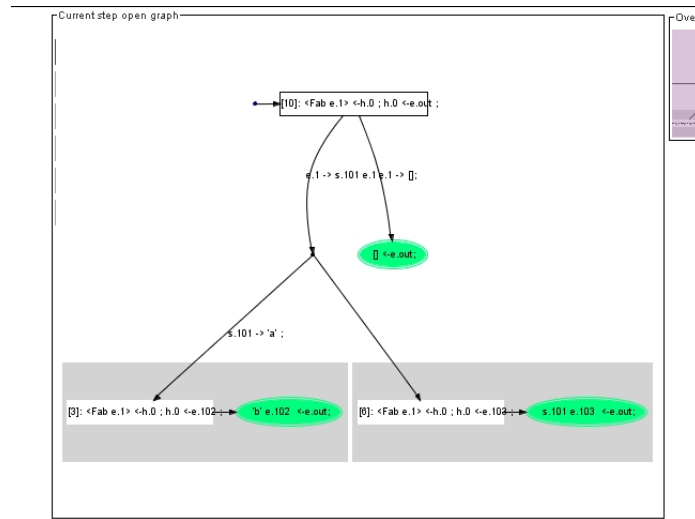


Рис. 3. Результат первого вызова прогонки (крупный масштаб)