

Реализация оптимизирующих трансформаций предикатных программ

КАБЛУКОВ ИВАН ВЛАДИМИРОВИЧ

Институт систем информатики имени А.П. Ершова СО РАН (Новосибирск), Россия
e-mail: vanyok05@mail.ru

Аннотация

Эффективность программ на языке предикатного программирования [1] достигается применением оптимизирующих трансформаций с последующей конвертацией на один из императивных языков: C, C++, ФОРТРАН и др. Применяются трансформации: замена хвостовой рекурсии циклом, открытая подстановка процедур, склеивание переменных [2], кодирование рекурсивных структур с помощью массивов и указателей. Итоговая программа по эффективности не уступает написанной вручную и, как правило, короче. Отметим, что для функциональных языков не удалось достичь приемлемой эффективности даже с применением изоощренных методов оптимизации.

Задачей работы является реализация нескольких трансформаций: склеивания переменных, замены хвостовой рекурсии циклом, открытой подстановки программ на места их вызовов. Реализация трансформаций требует предварительного потокового анализа предикатной программы. Трансформации сопровождаются проведением упрощений программы.

Для проведения трансформаций подстановки предиката на место вызова и склеивания переменных требуется построение графа вызов программы. Для трансформации склеивания переменных необходимо определение аргументов, результатов и пост-аргументов операторов в телах предикатов. Пост-аргументами оператора яв-

ляются переменные предиката, используемые после завершения исполнения текущего оператора. В отличие от замены хвостовой рекурсии циклом и открытой подстановки программ на места их вызовов, реализованных во многих языках и отличающихся только спецификой конкретных языков, инновационной является трансформация склеивание переменных.

Склеивание переменных – это замена (сохраняющая эквивалентность) в тексте программы всех вхождений одной переменной на другую. Значительный эффект достигается при склеивании структурных переменных, таких как массивы и списки, поскольку склеивание обычно позволяет избежать копирования структур.

В отличие от задачи экономии памяти [3], склеиванию в программе подлежат результаты с аргументами, аргументы с локалами и локалы с результатами. Набор склеиваний может быть частично задан пользователем. Необходимо проверить его корректность и дополнить. В языке P правилами языка запрещено присваивание вида: $x := op(x, y)$. Поэтому в языке P существуют только присваивания $x := op(x1, y)$. При трансляции $x1$ склеивается с x . Например, при склеивании переменных s и d оператор $s := d + 1$ будет преобразован в оператор присваивания $s := s + 1$, а оператор $a := b$ при склеивании a и b превратится в оператор $a := a$, удаляемый из программы. Типы склеиваемых переменных должны совпадать.

Регион склеивания для оператора G есть набор аргументов и результатов одного ти-

па $\langle x: y \rangle$, где x — список аргументов и y — список результатов оператора. **Пример.** Пусть имеется оператор F с аргументами a, b, c, d, e и результатами f, g, h . Пусть переменные a, b, d и g, h имеют натуральный тип, а переменные c, e и f — массивы. Тогда для оператора F регионы склеивания таковы: $\langle a, b, d: g, h \rangle$ и $\langle c, e: f \rangle$.

Алгоритм склеивания реализуется уточнением регионов. Уточнение регионов — это процесс, в результате которого исходный регион преобразуется в набор регионов, состоящих из одного аргумента и одного результата. Регионы оператора уточняются на основе регионов подоператоров для операторов суперпозиции, условного и параллельного операторов.

Список литературы

1. Шелехов В.И. Введение в предикатное программирование. - Новосибирск, 2002. - 82с. - (Препр. / ИСИ СО РАН; N 100).
2. Каблуков И. В., Шелехов В.И. Реализация склеивания переменных в предикатной программе. - Новосибирск, 2012. - 6с. - (Препр. / ИСИ СО РАН; N 167).
3. Ершов А.П. Введение в теоретическое программирование - М.: Наука, 1977. - 288с.

Работа выполнена при поддержке РФФИ № 12-01-000686.