

Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра системного программирования

Анализ зависимостей в среде PRANLIB

Дипломная работа студента 544 группы
Сыча Геннадия Александровича

Научный руководитель: к.ф.-м.н. Булычев Д. Ю.
Рецензент: Вигдорчик Е. Н.

Введение

Введение

Постановка
задачи

Реализация

Пример
построения

Результаты

- ▶ Такое представление программы как граф зависимостей по данным имеет широкое применение:
 - ▶ распараллеливание программ: необходимо сохранять семантику;
 - ▶ генерация кода: наличие графа зависимостей для элементарных блоков требуется для построения расписаний;
 - ▶ оптимизации кода: оптимизации использования кэша данных.

Введение

Введение

Постановка
задачи

Реализация

Пример
построения

Результаты

- ▶ Такое представления программы как граф зависимостей по данным имеет широкое применение:
 - ▶ распараллеливание программ: необходимо сохранять семантику;
 - ▶ генерация кода: наличие графа зависимостей для элементарных блоков требуется для построения расписаний;
 - ▶ оптимизации кода: оптимизации использования кэша данных.
- ▶ PRANLIB - библиотека алгоритмов анализа потока управления

Введение

Введение

Постановка
задачи

Реализация

Пример
построения

Результаты

- ▶ Такое представление программы как граф зависимостей по данным имеет широкое применение:
 - ▶ распараллеливание программ: необходимо сохранять семантику;
 - ▶ генерация кода: наличие графа зависимостей для элементарных блоков требуется для построения расписаний;
 - ▶ оптимизации кода: оптимизации использования кэша данных.
- ▶ PRANLIB - библиотека алгоритмов анализа потока управления
- ▶ Необходимо расширить библиотеку, дополнив ее функциональностью для решения задач анализа потока данных и анализа зависимостей

Постановка задачи

Рассматривают 3 типа зависимостей:

1. истинная зависимость:

$a \leftarrow$

$\leftarrow a$

2. анти-зависимость:

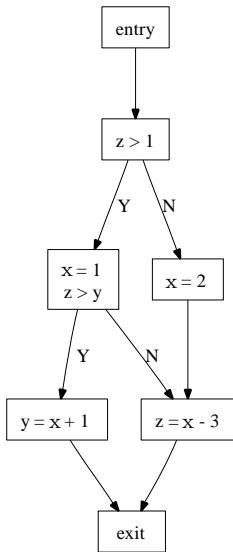
$\leftarrow a$

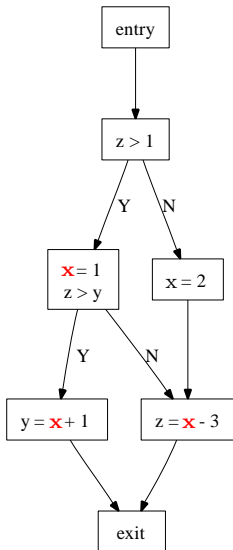
$a \leftarrow$

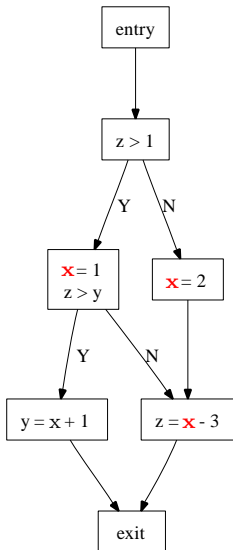
3. выходная зависимость:

$a \leftarrow$

$a \leftarrow$







Постановка задачи

1. Разработать представление для задач анализа потоков данных и реализовать его в виде модулей библиотеки PRANLIB

Постановка задачи

1. Разработать представление для задач анализа потоков данных и реализовать его в виде модулей библиотеки PRANLIB
2. На основе разработанного представления реализовать задачи о достигающих определениях и восходящем анализе использования переменных (reaching definitions, upward exposed uses)

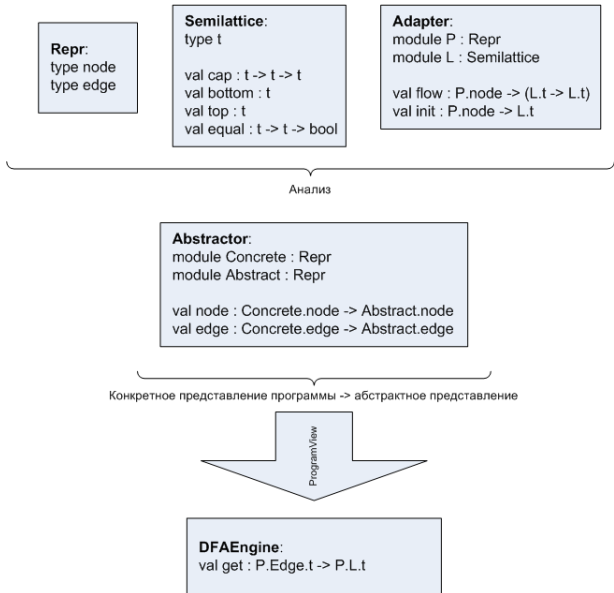
Постановка задачи

1. Разработать представление для задач анализа потоков данных и реализовать его в виде модулей библиотеки PRANLIB
2. На основе разработанного представления реализовать задачи о достигающих определениях и восходящем анализе использования переменных (reaching definitions, upward exposed uses)
3. Используя две задачи анализа потока данных, реализовать алгоритм построения графа зависимостей

Реализация

- ▶ Реализация проводилась на языке Objective Caml
- ▶ Представление задач анализа потока данных:
 - ▶ абстрактное представление программы;
 - ▶ полурешетка — элементы, операции пересечения (объединения);
 - ▶ потоковые функции;
 - ▶ отображение конкретного представления на абстрактное;
 - ▶ алгоритмы итерирования.

Реализация



Реализация

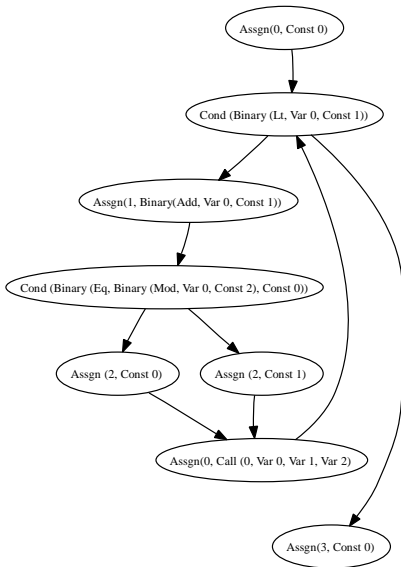
Граф зависимостей строится на основе решения двух задач анализа потока данных.

- ▶ достигающие определения — потоковые и выходные зависимости;
- ▶ задача о восходящем анализе использования переменных — анти-зависимости.

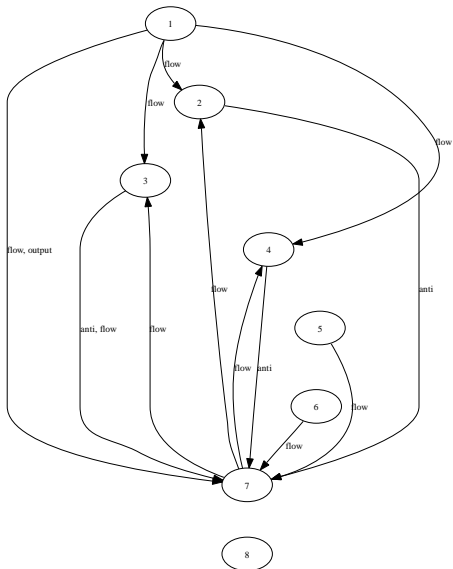
Программа

```
1.     i = 0;
2.     while(i < 10)
      {
3.         j = i + 1;
4.         if(i % 2 == 0)
          {
5.             k = 0;
              }
          else
              {
6.             k = 1;
              }
7.         i = f(i, j, k);
      }
8.     z = 0;
```

Граф потока управления



Граф зависимостей



Результаты

- ▶ Разработаны модули для реализации задач анализа потока данных

Результаты

- ▶ Разработаны модули для реализации задач анализа потока данных
- ▶ Модули организованы таким образом, что достигается значительное переиспользование кода на разных уровнях:
 - ▶ благодаря наличию абстракции программы можно использовать реализованный анализ на различных конкретных представлениях программ;
 - ▶ одно и то же абстрактное представление программы можно использовать для разных задач;
 - ▶ переиспользование полурешеток;
 - ▶ набор независимых алгоритмов итерирования.

Результаты

- ▶ Разработаны модули для реализации задач анализа потока данных
- ▶ Модули организованы таким образом, что достигается значительное переиспользование кода на разных уровнях:
 - ▶ благодаря наличию абстракции программы можно использовать реализованный анализ на различных конкретных представлениях программ;
 - ▶ одно и то же абстрактное представление программы можно использовать для разных задач;
 - ▶ переиспользование полурешеток;
 - ▶ набор независимых алгоритмов итерирования.
- ▶ На основе разработанных модулей реализованы две задачи анализа потока данных: достигающие определения и восходящий анализ использования переменных.

Результаты

- ▶ Реализован алгоритм построения графа зависимостей
- ▶ Разработанное представление задач анализа потока данных использовано при реализации анализа указателей в проекте Creen.