

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

Применение
предикативной
абстракции
программы для
решения
поточковых
задач

Применение предикативной абстракции программы для решения потоковых задач

Дипломная работа студента 544 группы
Буре Дмитрия Владимировича

Научный руководитель: Друнин А. В.
Рецензент: Булычев Д. Ю.

Введение

Постановка

Алгоритм

Пример

Особенности

Результаты

- ▶ Статический анализ программ
 - ▶ Классические задачи, имеющие известные алгоритмы решения
 - ▶ Смежные проблемы, возникающие на практике
- ▶ Задача определения значения переменных в точке программы
 - ▶ Протягивание констант
 - ▶ Анализ валидности указателей
 - ▶ Проверка входных параметров
 - ▶ Анализ достигающих определений
 - ▶ ...

Введение

Постановка

Алгоритм

Пример

Особенности

Результаты

Постановка задачи

- ▶ Обобщение некоторых алгоритмов статического анализа
- ▶ Разработка механизма решения смежных потоковых задач
- ▶ Анализ возможности применения методов, распространенных в узких областях
 - ▶ Проверки на модели (model checking)
 - ▶ Абстрактная интерпретация
 - ▶ Диаграммы двоичных решений (BDD)

- ▶ Предикативная абстракция исходной программы
 - ▶ используются только логические переменные (предикаты)
 - ▶ например: $x := y \Rightarrow \{x = y\} := \text{true};$
 - ▶ абстракция имеет эквивалентный поток управления
- ▶ Анализ достигающих определений
 - ▶ множества состояний представляется набором бит-векторов
 - ▶ множества представляются диаграммами двоичных решений
- ▶ Логическая формула в каждой точке
 - ▶ несовместность означает недостижимость точки
 - ▶ происходит вывод результирующих значений
 - ▶ нумерация SSA формы определяет порядок подстановки

Пример

<pre>init(x); init(y); if(y==6) { x = 1; exit(); } else x = 2; use x;</pre>	<pre>init(x₀); init(y₀); if(y₀==6) { x₄ = 1; exit(); } else x₃ = 2; x₂ = φ(x₃, x₄); use x₂;</pre>	<pre>if(*) { assume({y₀==6}); {x₄ == 1} = true; exit(); {x₂==x₄}={x₄==1}; } else { assume(!{y₀==6}); {x₃ == 2} = true; {x₂==x₃}={x₃==2}; } use x₂;</pre>
---	---	--

$$!\{y_0 == 6\} \wedge \{x_3 == 2\} \wedge \{x_2 == x_3\}$$

Введение

Постановка

Алгоритм

Пример

Особенности

Результаты

Особенности алгоритма

- ▶ Выбор предикатов позволяет задавать интересующие свойства
- ▶ Путезависимость
- ▶ Итеративная часть алгоритма
 - ▶ не использует систему времени выполнения
 - ▶ языконезависима
 - ▶ основана на диаграммах двоичных решений
- ▶ Вывод результатов
 - ▶ вычисление лишь необходимой информации
 - ▶ кэширование
 - ▶ оптимизации связанные с ленивыми вычислениями

Результаты работы

- ▶ Обнаружены сходства некоторых рассмотренных алгоритмов
- ▶ Предложен способ промежуточного представления программы
- ▶ Разработан механизм решения смежных потоковых задач
- ▶ Реализован алгоритм решения задачи протягивания констант
- ▶ Дальнейшее развитие
 - ▶ Поддержка полного набора операторов и типов языка
 - ▶ Применение абстрактного решателя потоковых задач
 - ▶ Использование итеративного выбора предикатов
 - ▶ Уточнение результатов в случае анализа циклов

Применение
предикативной
абстракции
программы для
решения
потоковых
задач

Введение

Постановка

Алгоритм

Пример

Особенности

Результаты