

Применение метода суперкомпиляции для специализации реляционных программ

Мария Куклина, M4236

Университет ИТМО

Научный руководитель: Блинец Иван Александрович

Научный консультант: Вербицкая Екатерина Андреевна

2020

Реляционное программирование

Определение

Вид декларативного программирования, в котором программы представляются как набор отношений между аргументами.

Пример

Пример *запросов* для отношения умножения $\text{mul}^o \subseteq \text{Int} \times \text{Int} \times \text{Int}$:

- $\text{mul}^o(2, 2, 4)$ — проверка корректности отношения.
- $\text{mul}^o(2, 2, C)$ — поиск всех C , таких что $2 * 2 = C$.
- $\text{mul}^o(A, 1, 4)$ — поиск всех A , таких что $A * 1 = 4$.
- $\text{mul}^o(A, B, C)$ — поиск всех троек A, B, C , таких что $A * B = C$.

Встраиваемый предметно-ориентированный язык реляционного программирования¹.

Применение

- Легковесная логическая подсистема проекта.
- Поиск лечения редких генетических заболеваний в точной медицине².
- Генерация программ по спецификации входов и выходов на основе *реляционного интерпретатора*.
- Порождение решения задач поиска по решению задачи распознавания³.

¹“*Relational Programming in miniKanren: Techniques, Applications, and Implementations*”, Byrd, 2009

²“*The Algorithm for Precision Medicine (Invited Talk)*”, Might, 2019

³“*Relational Interpreters for Search Problems*”, Lozov, Verbitskaia и Boulytchev, 2019

Постановка проблемы

- Сложные алгоритмы в miniKanren работают медленно.
- Поиск входов по выходам отношения работает значительно медленнее, чем “прямой” запуск.

Примеры

- Программы, порождённые инструментом по трансляции из функционального языка в miniKanren⁴.
- Порождение решения задач поиска.

⁴“*Typed Relational Conversion*”, Lozov P., 2018

Специализация

Определение

Автоматизированная техника оптимизации программ, при которой из программы удаляются избыточные вычисления, зависимые от частично известного входа⁵.

- Частичная дедукция — класс методов специализации для логических языков, в частности, для Prolog.⁶
- Специализация miniKanren на основе *конъюнктивной частичной дедукции (CPD)*⁷.
 - Сложна в поддержке, даёт нестабильные результаты.
 - Однако предоставляет библиотеку для построения специализаторов.

⁵ *Partial evaluation and automatic program generation*, Jones, Gomard и Sestoft, 1993

⁶ *Advanced Techniques for Logic Program Specialisation*, Leuschel, 1997

⁷ *“Relational Interpreters for Search Problems”*, Lozov, Verbitskaia и Boulytchev, 2019

Суперкомпиляция

Определение

Техника автоматической трансформации и анализа программ, при которой программа символично исполняется с сохранением истории вычислений, на основе которой принимаются решения о трансформации и оптимизации.

- Суперкомпиляторы применяются во основном для функциональных языков⁸.
- Суперкомпиляция позволяет достичь не более чем линейного ускорения⁹
- Полуавтоматическая суперкомпиляция для Prolog¹⁰.
- Теоретические доводы для автоматической суперкомпиляции для Prolog⁹.

⁸ "Introduction to Supercompilation", Sørensen и Glück, 1998

⁹ *Turchin's Supercompiler Revisited - An operational theory of positive information propagation*, Sørensen, 1996

¹⁰ *A Prolog Positive Supercompiler*, Diehl, 1997

Цели и задачи

Цель

Улучшение результатов специализации реляционных программ путём применения метода суперкомпиляции.

Задачи

- Реализовать базовый суперкомпилятор для miniKanren.
- Рассмотреть возможные методы улучшения получившегося суперкомпилятора.
- Протестировать результаты и сравнить их с результатами CPD и с оригинальными программами.

Суперкомпиляция для miniKanren

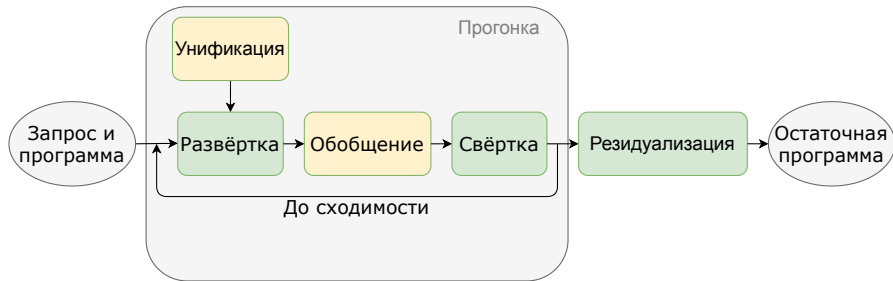


Рис.: Схема алгоритма суперкомпиляции



— библиотека по специализации miniKanren с дополнениями



— собственная разработка

Особенности шага развёртки для miniKanren

Развёртка определяет шаг символьного вычисления в суперкомпиляторе, на котором порождается множество возможных состояний программы.

Значимые отличия

- Несколько возможных шагов вычисления.
- Допускается переупорядочивание элементов выражения.

Результаты задачи

- Реализован базовый алгоритм суперкомпиляции.
 - Развёртка рассматривает все возможные состояния.
 - Используемый алгоритм обобщения основан на алгоритме для конъюнктивной частичной дедукции, для которого доказана терминируемость.
- Разработан и реализован алгоритм построения оптимизированной программы по графу суперкомпиляции.

Улучшение суперкомпиляции для miniKanren

Проблемы

- Повторение символьных вычислений из-за стратегии свёртки.
- Классическое использование обобщения может приводить к избыточным вычислениям.
 - Существует техника обобщения, описанная в статьях¹¹.
 - Придумана специфичная для miniKanren техника обобщения.
- Тривиальная стратегия вычисления порождает слишком много ветвей исполнения.
- В используемой реализации miniKanren нет способа эффективно сообщить, что можно прервать вычисление.

¹¹ "Introduction to Supercompilation", Sørensen и Glück, 1998

Результаты задачи

- Применены подходы по улучшению алгоритма суперкомпиляции.
 - Добавлено кэширование.
 - Реализованы модификации обобщения.
 - Проанализированы и реализованы допустимые стратегии вычисления.
- Расширение библиотеки для специализации неравенствами.
- Расширение суперкомпилятора, при котором учитывается “негативная” информация.

Тестирование

Реализация `miniKanren`: проект `OCanren`¹²

Реализация CPD для `miniKanren`: проект `uKanren_transformations`¹³

Реализация CPD для `Prolog`: проект `ECCE`¹⁴

Платформа: Intel Core i5-6200U CPU, 2.30GHz, DDR4, 12GiB.

Сценарий тестирования:

- 1 Суперкомпиляция тестовой программы.
- 2 Трансляция остаточной программы в `OCanren`.
- 3 Замер времени исполнения.
- 4 Сравнение времени исполнения с оригинальной программой и реализациями CPD.

¹²<https://github.com/JetBrains-Research/OCanren>

¹³https://github.com/kajigor/uKanren_transformations/

¹⁴<https://github.com/leuschel/ecce>

Тестирование

Программы для тестирования

sort Алгоритм реляционной сортировки.

Запрос : сортировка случайного списка длины 50

isPath Проверка принадлежности пути графу.

Запрос: поиск произвольного пути длины 10, принадлежащих графу с 21 вершиной и 50 рёбрами.

logint Реляционный интерпретатор формул логики высказываний.

Запрос: поиск 1000 истинных формул в данной подстановке.

lam Реляционный интерпретатор лямбда-выражений.

Запрос: поиск n термов, сводящихся к указанной форме.

Результаты тестирования

Базовый суперкомпилятор

Параметр	Оригинал	ECCE	CPD	Суперкомп.
sort	случайный список длины 50			
	8.42	12.28	13.2	0.28
isPath	произвольный путь длины 10			
граф 1	> 300	9	10	0.25
граф 2		2.7	2.8	0.1
logint	размер подстановки			
0	> 300	0.17	2.7	0.09*
1		0.09	1.7	0.07*
lam	редуцируются			
10 термов к себе	0.17	0.001	0.008	0.002
50 термов к себе	> 300	2.98	4.32	1.89
1000 термов к const	1.01	0.126	0.263	0.274

Рис.: Тестовые результаты, секунды

Результаты тестирования

Улучшения

Программа: генерация все пар термов и типов, соответствующие заданной спецификации.

Оригинал: 4.10s

Ессе: 0.28s

Стратегия	Базовый	M.1	M.2	M.3
Full	-	0.72	1.44	1.38
Seq	0.11	0.14	0.08	0.08
Non-rec	0.07	0.08	0.07	0.07
Rec	0.15	0.13	0.11	0.11
Min	0.20	0.12	0.12	0.11
Max	0.09	0.14	0.10	0.12

Рис.: Сравнение модификаций, секунды

Результаты работы

- Реализован и протестирован суперкомпилятор для задачи специализации.
- Применены подходы по улучшению качества суперкомпиляции для задачи специализации.
- Добавлены ограничения неравенства в библиотеку по специализации.
- Исправление багов библиотеки для специализации.

Спасибо за внимание!

- Работа будет представлена во второй половине мая на воркшопе по трендам логического программирования TEASE-LP.
- Ссылка на репозиторий:
<https://github.com/RehMaar/uKanren-spec>