

Оптимизация исполнения реляционных программ

Дима Розплохас

Руководитель: Дмитрий Юрьевич Булычев

Санкт-Петербургский Академический университет

Осенний семестр 2016

MiniKanren

$T ::= x$
| $C(T, \dots, T)$

$R ::= T \equiv T$
| $R \wedge R$
| $R \vee R$
| $\exists x, R$
| $R_name(T, \dots, T)$

Зачем? Чтобы писать обратимые функции

Проблема

$R \vee R$ выполняется параллельно.

$R \wedge R$ выполняется последовательно.

Порядок может влиять на завершаемость, причем оптимальный порядок может зависеть от запроса.

Будем менять порядок в процессе исполнения программы.

Задачи

- ▶ Создать платформу для оптимизации
- ▶ Придумать алгоритм
- ▶ Реализовать его и проверить работоспособность
- ▶ Формализовать результат

Shallow embedding vs Deep embedding

Shallow: функции исполняют программу.

- ▶ Быстрее
- ▶ Легче расширять язык

Deep: функции создают промежуточное представление программы, которое потом интерпретируется.

- ▶ Проще реализуется
- ▶ Есть доступ к программе в процессе исполнения

Shallow $\xrightarrow{\text{нужно изменить синтаксис}}$ Deep

Идея алгоритма

1. Периодически выполняем проверку программы на незавершаемость
2. Идём вверх по коду и пытаемся вытащить вперед какой-нибудь конъюнкт
3. Запускаем вычисления с самого начала

Реализация

- ▶ Большинство простых программ “чинятся” оптимизацией (работа со списками, сортировка, арифметика Пеано)
- ▶ Есть неподдающиеся примеры
- ▶ Бонус: простая двоичная арифметика

Формальные результаты (в идеале)

Hypothesis

Если программа завершается, она завершится исполнении с оптимизацией.

Hypothesis

Если существует порядок при котором программа завершается и при любом порядке расходимость отлавливается проверкой, то программа завершится при исполнении с оптимизацией.

Результаты

- ▶ Deep embedding
- ▶ Динамический алгоритм
- ▶ Реализация
- ▶ Идеи, что ещё можно попробовать или попытаться доказать

Ограничения:

- ▶ Измененный синтаксис
- ▶ Поиск конечного числа решений

Ура! Это конец

Репозиторий: <https://github.com/rozplokhas/OCanren/tree/deep>