



Санкт-Петербургский государственный университет
Кафедра системного программирования

Трассирующая нормализация

Березун Даниил Андреевич

Специальность: [05.13.11](#) “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей”

Научный руководитель: д.т.н., проф. Кознов Д.В. (СПбГУ)

Официальные оппоненты: д.ф.-м.н., проф. Непейвода Н.Н. (ИПС им.Айламазяна РАН)

Ведущая организация: к.ф.-м.н. Миронов А.М. (МГУ)
ИПМ им. М.В. Келдыша РАН

Санкт-Петербург

2018

Лямбда-исчисление (λ)

- › A. Church [1930-e]
- › S. Kleene и J. B. Rosser: противоречивость [1935]
- › A. Church: ULC [1936]
STLC [1940]
- › Одна из основных моделей вычислений

Стратегии вычислений

- › Строгие (сильные) порядки редукций
 - Нормальный порядок
 - Аппликативный порядок
- › Слабые порядки
 - Вызов по имени (ALGOL 60, делегаты в .Net)
 - Вызов по значению (ALGOL 60, Common Lisp, Java, Scheme, OCaml, C#)
 - Вызов по необходимости (Haskell, R, Clean)
- › Гибридные порядки, ...

Предметная область — специализация программ

Проекция Футамуры-Ершова-Турчина

- 1 $target = \llbracket spec \rrbracket(int, source)$ $spec$ компилирует λ -терм
- 2 $comp = \llbracket spec \rrbracket(spec, int)$ компилятор
- 3 $cogen = \llbracket spec \rrbracket(spec, spec)$ генератор компиляторов

> Неизменяемость программ

Существующие работы

Трассирующая нормализация (TBN)

- Нестандартный подход к вычислениям
 - Нормализация путём обхода входного термина
 - Трассы (история обхода) вместо замыканий, окружений, и т.п.
 - Множество трасс определяет нормальную форму

Существующие решения

- Для безопасного λ -исчисления (SLC) [Blum, Ong, 09]
- Для STLC (SLC \subset STLC) [Ong, 15]
- ✗ Нет решения для алгоритмически полного исчисления

Связь с синтаксической редукцией λ -термов

Головная линейная редукция (HLR) [Danos, Regnier, 04]

Задачи

- 1 Разработать алгоритм трассирующей нормализации термов нетипизированного лямбда-исчисления
- 2 Формально доказать корректность предложенного алгоритма
- 3 Исследовать возможность адаптации трассирующей нормализации для различных стратегий вычислений
- 4 Исследовать выразительную мощность предложенного подхода с точки зрения компиляции основных конструкций функциональных языков программирования

Результаты

- 1 Разработан алгоритм трассирующей нормализации для нетипизированного лямбда-исчисления, соответствующий нормальному порядку редукций
- 2 Представлена модель CHLR, формализованная в виде системы переходов, доказана её корректность
- 3 Доказана корректность представленного алгоритма TBN относительно предложенной модели CHLR
- 4 Предложенный алгоритм TBN адаптирован для аппликативного порядка редукций и вызова по необходимости
- 5 Предложен новый метод компиляции функциональных языков в низкоуровневое представление путём специализации представленного алгоритма TBN на входной терм

Алгоритм трассирующей нормализации (для ULC)

$$\mathcal{R}[[e]] = eval [\langle e F [] [] \rangle]$$

eval $h =$ **let** $it : _ = h$ **in case** it **of**

$$\langle (FV x) \alpha bh ch \rangle \Rightarrow apk (FV x) ch h$$

$$\langle (BV x i) \alpha bh ch \rangle \Rightarrow lookup i \alpha bh ch h$$

$$\langle \lambda x.e T bh ch \rangle \Rightarrow apk \lambda x.e ch h$$

$$\langle \lambda x.e F bh ch \rangle \Rightarrow eval \langle e F bh ch \rangle : h$$

$$\langle e_1 @ e_2 \alpha bh ch \rangle \Rightarrow eval \langle e_1 T bh h \rangle : h$$

lookup $0 \alpha (\langle _ T _ ch' \rangle : _) ch h =$ **case** ch' **of**

$$\langle e _ bh _ \rangle : _ \Rightarrow evoperand \langle e \alpha bh ch \rangle h$$

$$_ \Rightarrow \mathbf{case} \mathit{ch} \mathbf{of}$$

$$[] \Rightarrow h$$

$$\langle ap _ bh'' ch'' \rangle : _ \Rightarrow evoperand \langle ap F bh'' ch'' \rangle h$$

lookup $0 _ (\langle _ F _ ch' \rangle : h') ch h = apk (BV _ 0) ch h$

lookup $i \alpha (\langle _ _ bh' _ \rangle : _) ch h = lookup (i - 1) \alpha bh' ch h$

apk $_ [] h = h$

apk $\lambda x.e (\langle _ \alpha _ ch \rangle : _) h = eval \langle e \alpha h ch \rangle : h$

apk $_ (\langle e \alpha bh ch \rangle : _) h = evoperand \langle e F bh ch \rangle h$

evoperand $\langle e_1 @ e_2 \alpha bh ch \rangle h = eval \langle e_2 \alpha bh ch \rangle : h$

Результаты

- 1 Разработан алгоритм TBN для Λ , соответствующий нормальному порядку редукций
- 2 Представлена модель *полной головной линейной редукции*, формализованная в виде системы переходов, доказана её корректность
- 3 Доказана корректность представленного алгоритма трассирующей нормализации относительно предложенной модели полной головной линейной редукции
- 4 Предложенный алгоритм TBN адаптирован для аппликативного порядка редукций и вызова по необходимости
- 5 Предложен новый метод компиляции функциональных языков в низкоуровневое представление путём специализации представленного алгоритма TBN на входной терм

Корректность

Теорема 1 (головной линейной редукции)

- 1 HLR завершается \Leftrightarrow HR завершается
- 2 HLR завершается в qhn

Теорема 2 (полной головной линейной редукции)

- 1 CHLR завершается \Leftrightarrow HR завершается
- 2 CHLR завершается в нормальной форме

Теорема 3 (предложенного алгоритма TBN)

Системы переходов для CHLR и UNP связаны отношением бисимуляции

Результаты

- 1 Разработан алгоритм TBN для Λ , соответствующий нормальному порядку редукций
- 2 Представлена модель CHLR, формализованная в виде системы переходов, доказана её корректность
- 3 Доказана корректность представленного алгоритма TBN относительно предложенной модели CHLR
- 4 Предложенный алгоритм трассирующей нормализации адаптирован для аппликативного порядка редукций и вызова по необходимости
- 5 Предложен новый метод компиляции функциональных языков в низкоуровневое представление путём специализации представленного алгоритма TBN на входной терм

Адаптация UNP для других стратегий вычислений

UNP адаптирован для

- Вызова по значению
- Аппликативного порядка
- Вызова по необходимости

Ограничения

- Возможность сохранения результата вычисления в обходе
- Частичная утрата полуконпозициональности

- Пилотная реализация на Haskell



Результаты

- 1 Разработан алгоритм TBN для Λ , соответствующий нормальному порядку редукций
- 2 Представлена модель CHLR, формализованная в виде системы переходов, доказана её корректность
- 3 Доказана корректность представленного алгоритма TBN относительно предложенной модели CHLR
- 4 Предложенный алгоритм TBN адаптирован для аппликативного порядка редукций и вызова по необходимости
- 5 Предложен новый метод компиляции функциональных языков в низкоуровневое представление путём специализации представленного алгоритма трассирующей нормализации на входной терм



Компиляция путём специализации UNP

Если $NP \in \begin{array}{|c|} \hline \Lambda \\ \hline L \\ \hline \end{array}$, то $\llbracket spec \rrbracket(spec, NP) \in \begin{array}{|c|} \hline \Lambda \quad \rightarrow \quad LLL \\ \hline L \\ \hline \end{array}$

Полукомпозициональность

Любой рекурсивный вызов в UNP применяется к подтерму исходного терма

Пилотная реализация

Реализовано на Haskell и Racket с помощью генерирующих расширений

Новизна результатов

- Введено и формализовано понятие полной головной линейной редукции, доказана её корректность
- Представлен алгоритм трассирующей нормализации для нетипизированного лямбда-исчисления, доказана его корректность
- Предложен новый подход к компиляции функциональных языков программирования путём специализации процедуры трассирующей нормализации на входной терм, и исследованы её свойства

ВАК

- 1 Д.Березун. Полная головная линейная редукция // НТВ-ИТУ/2017 #3.
- 2 Д.Березун. Трассирующая нормализация нетипизированного лямбда-исчисления // Известия вузов. Северо-Кавказский регион. Технические науки. 2017.

Web of Science и SCOPUS

- 3 D. Berezun, Neil D. Jones. Compiling Untyped Lambda Calculus to Lower-Level Code by GS and PE. PEPM 2017.
- 4 D. Berezun, D. Boulytchev. Precise Garbage Collection for C++ with a Non-Cooperative Compiler. SECR-2014.

РИНЦ

- 5 D. Berezun, N. D. Jones. 2016. Working Notes: Compiling ULC to Lower-Level Code by GS and PE. META 2016.

Результаты

- 1 Разработан алгоритм *трассирующей нормализации* для термов нетипизированного лямбда-исчисления
- 2 Представлена модель *полной головной линейной редукции*, формализованная в виде системы переходов, доказана её корректность
- 3 Доказана корректность представленного алгоритма трассирующей нормализации относительно предложенной модели полной головной линейной редукции
- 4 Предложенный алгоритм трассирующей нормализации адаптирован для вызова по значению, аппликативного порядка редукций и вызова по необходимости
- 5 Предложен новый метод компиляции функциональных языков в низкоуровневое представление путём специализации представленного алгоритма трассирующей нормализации на входной терм