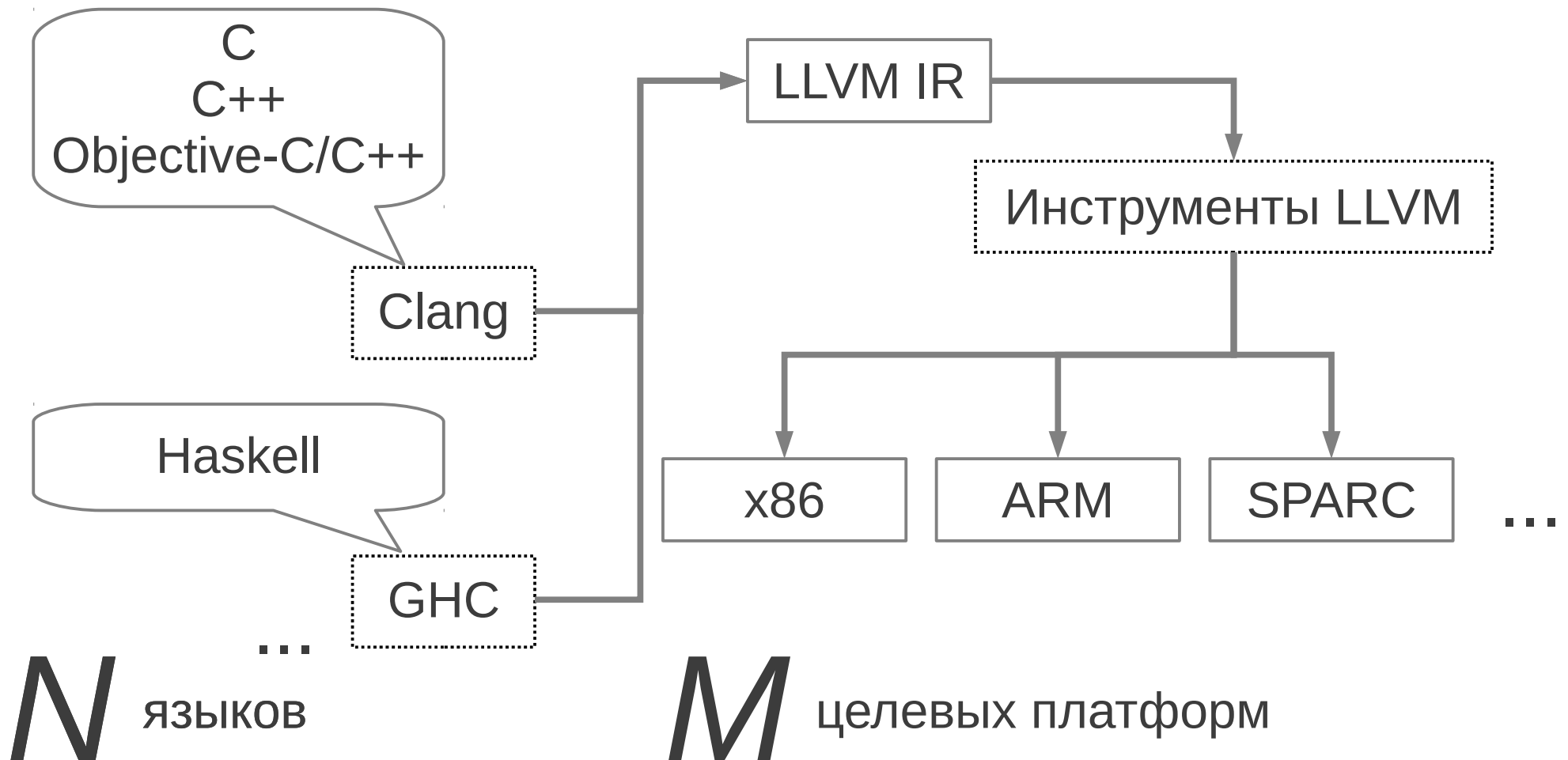


# Перенос компилятора OCaml в инфраструктуру LLVM

Студент: Терешин Роман, 344 группа  
Научный руководитель: к.ф.-м.н. Булычев Д. Ю.

# Инфраструктура LLVM

Chris A. Lattner, Иллинойский университет, 2002



# OCaml

Xavier Leroy, INRIA, 1990/2011

- Функциональный
- Императивный
- Объектно-ориентированный
  
- Полиморфизм
- Сборка мусора
- Развитая система модулей
- Автоматический вывод типов



# Цели работы

- Апробация LLVM
- Улучшение компилятора OCaml
- Создание среды для тестирования сборки мусора

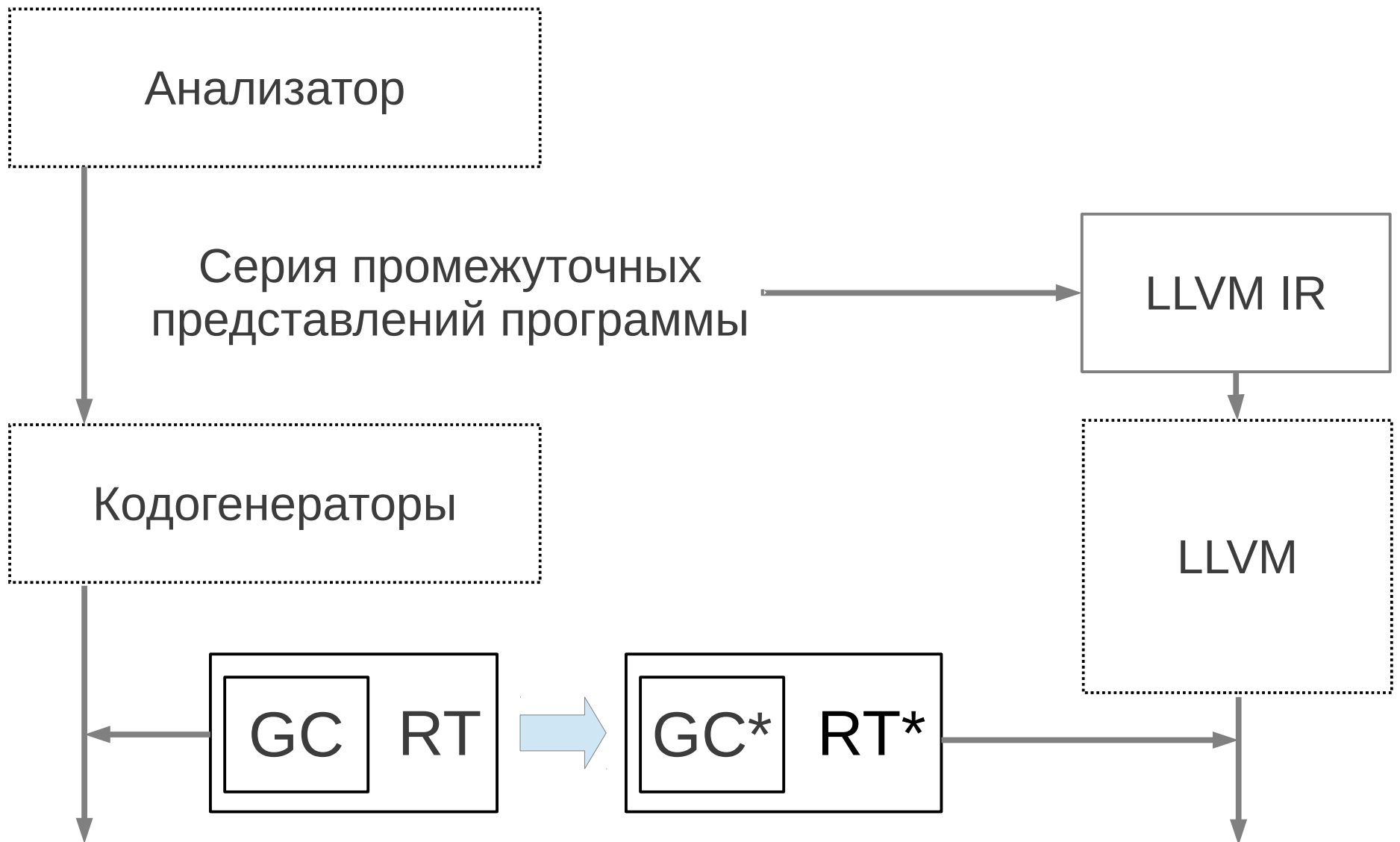
# Связанные работы

- LLVM-бэкенд для GHC (Haskell), 2010
- LLVM-бэкенд для HiPE (Erlang OTP), 2012
- Проект osamlvm (OCaml), 2012

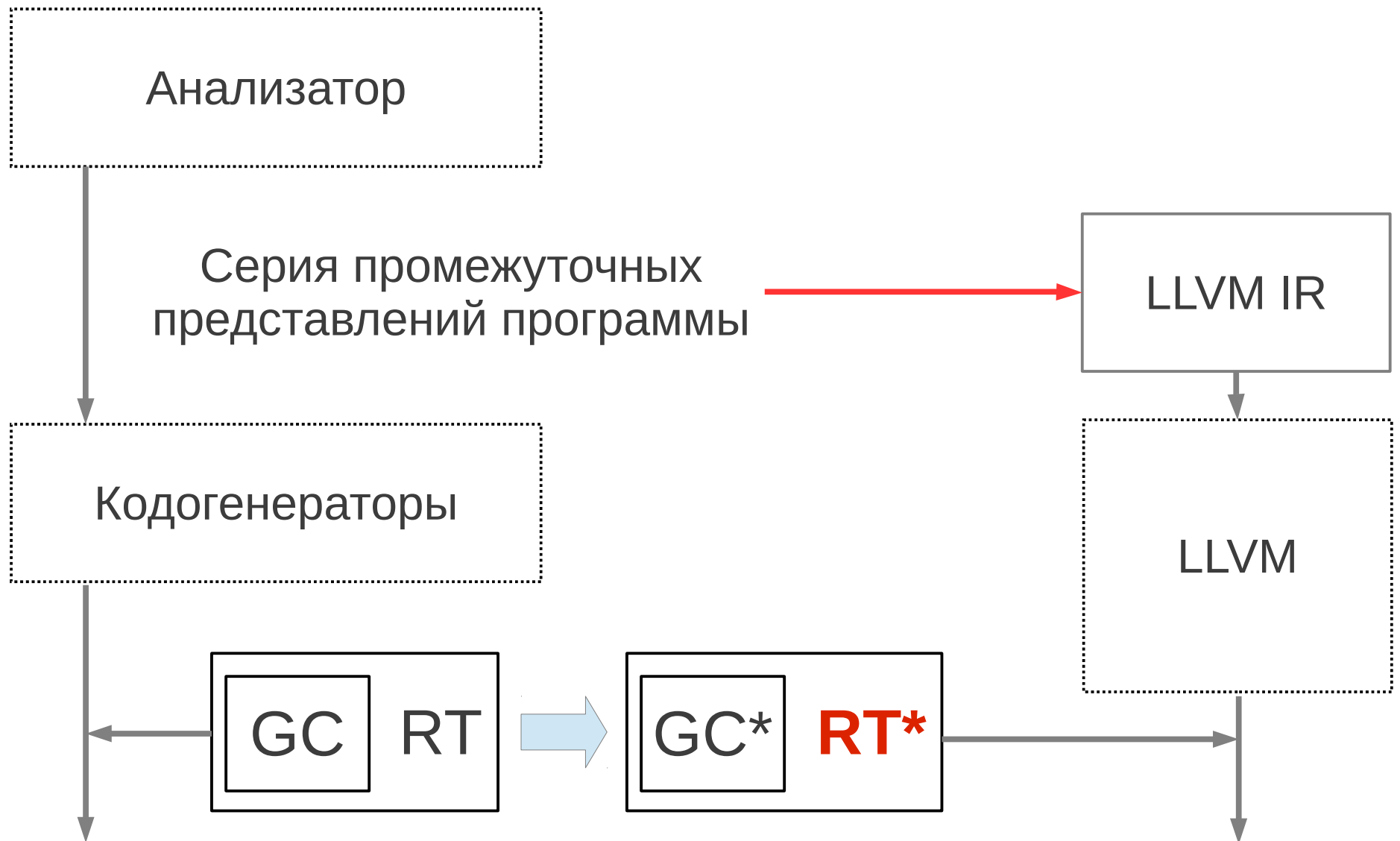
# Компиляторы ocamlopt и ocamlnc



# Компиляторы ocamlopt и ocamlnc



# Компиляторы ocamlopt и ocamlnc



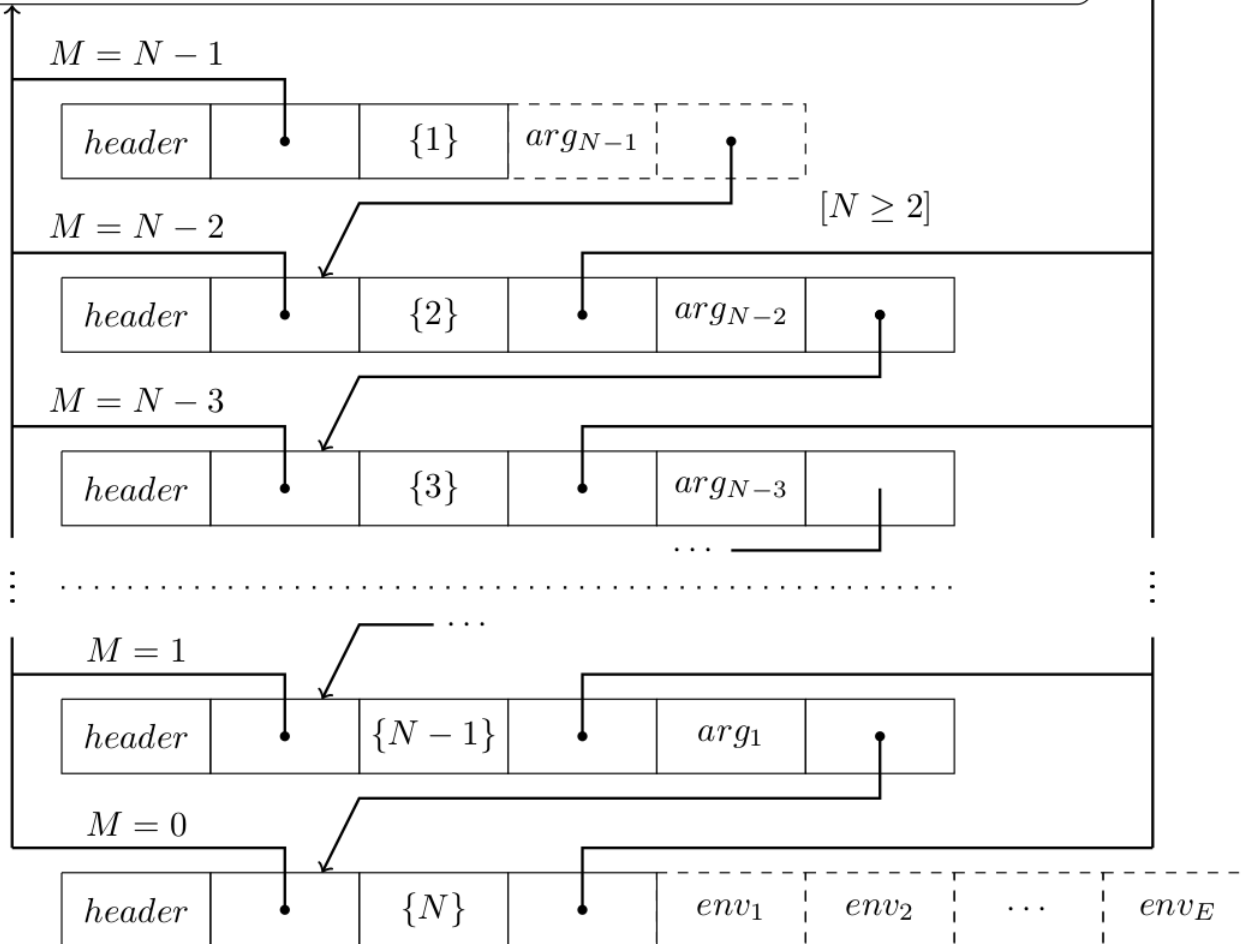


# Транслятор Cmm → LLVM IR

- Cmm – C-подобное промежуточное представление
- LLVM IR – типизированная RISC-подобная SSA-форма

# Частично применённые функции

$$\begin{cases} i64 @caml\_curryN\_M\_app(i64 \%arg_{M+1}, i64 \%arg_{M+2}, \dots, i64 \%arg_N, i64 \%closure), & M \neq 0 \\ i64 @f(i64 \%arg_1, i64 \%arg_2, \dots, i64 \%arg_N[, i64 \%closure]_{[E \neq 0]}), & M = 0 \end{cases}$$

$$\begin{cases} i64 @caml\_curryN[_M]_{[M \neq 0]}(i64 \%arg_{M+1}, i64 \%closure), & N \neq 1 \\ i64 @f(i64 \%arg_1[, i64 \%closure]_{[E \neq 0]}), & N = 1 \end{cases}$$


# Исключительные ситуации

- Нелокальные переходы (setjmp / longjmp)
- Табличный подход (Intel Itanium C++ ABI)

# Результаты

- Апробирована и изучена инфраструктура LLVM
- Разработан действующий LLVM-бэкенд для некоторого подмножества языка OCaml
- Показана перспективность дальнейших работ
- Создана среда для разработки и тестирования сборщика мусора для LLVM

<https://github.com/ramntry/ocamlinc>  
<http://goo.gl/pNbF6f>