

Лаборатория анализа программ и верификации

Марат Ахин Михаил Беляев

2016

- Лаборатория анализа и верификации программ
- Базируется на кафедре КСПТ, ИКНТ, СПбПУ (ака Политех)
- Работает с JetBrains с начала 2015 года

- Статический и динамический анализ
- Верификация
- Программная инженерия

- Примерные темы для исследований
- Детали более чем обсуждаемы
- Собственные темы приветствуются¹

¹если они устраивают обе стороны процесса

- В LLVM есть различные примитивные реализации
- Средствам статического анализа хочется иметь анализы точнее существующих
- Реализовать один из известных алгоритмов анализа указателей в рамках LLVM

- Множество средств статического анализа используют контракты
- Ошибки использования определенной библиотеки можно искать при помощи предметно-ориентированных контрактов
- Реализовать поиск ошибок использования какой-либо библиотеки в программах на языке C²

²на основе существующих средств анализа

- `rustc` использует LLVM в качестве бекэнда
- Существует множество средств анализа LLVM IR
- Написать фронтэнд, осуществляющий сопряжение Rust и средств анализа LLVM

- Языки семейства C поддерживают ассемблерные вставки
- LLVM IR является переносимым ассемблером
- Реализовать механизм ассемблерных вставок на LLVM IR

- Стандартный механизм интероперабельности — FFI
- С другой стороны, множество языков транслируются в LLVM IR
- Добавить возможность вызова функций языка A из языка B³

³ например, Rust \Rightarrow C++

- Тестировать код — нужно
- Тестировать код автоматически — удобно
- Реализовать поддержку QuickCheck для Kotlin

- Конфликты при merge — это плохо
- Поздние конфликты при merge — это совсем плохо
- Спекулятивно предсказывать будущие конфликты и уведомлять об этом пользователя

- Большинство проектов имеют определенную политику создания коммит-сообщений
- Иногда программисты не хотят придумывать хорошие коммит-сообщения
- Генерировать шаблон коммит-сообщения по содержанию коммита

- Исправление конфликтов при слияниях веток — это неприятный процесс
- Иногда исправления выполняются по какому-либо шаблону
- Предложить интеллектуальный вариант слияния конфликтов

- Большие коммиты усложняют понимание внесенных изменений
- Часто большой коммит представляет собой нескольких слабо связанных изменений
- Автоматически разбивать коммит на несколько частей

- `akhin@kspt.icc.spbstu.ru`
- `belyaev@kspt.icc.spbstu.ru`

- `vlad@icc.spbstu.ru`