Михаил Разаков

C++ / Systems Engineer | LLVM · x86-64 · Algorithms

razakov.mikhail@outlook.com
t.me/Micodiy
github.com/Misha1302

Разрабатываю C++/LLVM и системные компоненты: безопасные преобразования IR, генерацию x86-64 и анализ зависимостей памяти. Для аффинных случаев использую точный подсчёт, для остальных — консервативные проверки и независимые тестовые оракулы.

Профессиональная и проектная база
ИСП РАН: SharpChecker, символьные вычисления · МЦСТ: LLVM 22/C++23.

x86-64 pipeline
SSA/CFG validation, liveness/interference, linear scan, seeded simulated annealing, SysV codegen и differential execution.

Conservative optimization
LLVM Global-IV: APInt affine evolution, transitive effects, explicit legality и 29 verifier/idempotence/execution regressions.

ОПЫТ

октябрь 2026 — настоящее время

ИСП РАН — разработка SharpChecker

- Работаю над символьными вычислениями в SharpChecker — инструменте статического анализа C#/.NET.

1 июля — 31 августа 2026

МЦСТ — стажёр по разработке компиляторов

- Реализовал LLVM LICM-pass на C++: допустимость преобразования определяется по структуре цикла, обращениям к памяти, side effects и speculative safety.
- Построил Python-harness для сравнения исполнения до/после оптимизации и явной проверки transformation / no-transformation cases; также реализовал RPO, Dijkstra, Dinic и Tarjan SCC на C++23.

ПРОЕКТЫ

LLVM Interprocedural Global IV Optimization

Консервативный LLVM 22 prototype/MVP с реальным IR transform; 29 positive/negative regression cases проверяют verifier validity, idempotence и before/after execution.

x86-64 Compiler Backend & Register Allocation

Verifier назначения регистров/stack slots и differential execution: native-код сравнивается с эталонным интерпретатором; тесты покрывают ветвления, циклы и spills.

PS-form Memory Dependence Analyzer

Точный подсчёт пар итераций в аффинных случаях и модульные/НОД-доказательства независимости; yes/no проверяются независимым оракулом и метаморфными тестами, неопределённость — maybe.

КОМПЕТЕНЦИИ

C++ / toolchain
C++23, C17, LLVM 22, CMake, Linux, clang-tidy, ASan/UBSan

Algorithms / analysis
graph algorithms, CFG, dominance, SSA, SCC, dataflow, memory dependence

Machine-level systems
x86-64 SysV, code generation, liveness/interference, register allocation, generated-code analysis

Verification
exact/reference oracles, differential/metamorphic testing, isolated execution, reproducible counterexamples

ОБРАЗОВАНИЕ

НИУ ВШЭ — Программная инженерия

ДОСТИЖЕНИЯ

Отбор МЦСТ — LLVM / C++
По письму организатора: единственная оценка 5,0/5,0 и единственное решение, прошедшее 104/104 теста, среди 49 присланных работ.

LangDev'26 — Build the Language, Then Make the Abstractions Disappear
9 октября 2026 выступил на LangDev с докладом об архитектуре UniversalToolchain.

Москва / удалённо