Mikhail Razakov

Compiler & Program Analysis Engineer

razakov.mikhail@outlook.com
github.com/Misha1302
LinkedIn

Compiler and program-analysis engineer working on symbolic computation in SharpChecker at ISP RAS, with prior LLVM 22/C++ experience at MCST. Public projects span interprocedural LLVM analysis, Roslyn CFG data flow, and x86-64 code generation.

PROFESSIONAL EXPERIENCE

October 2026 — present

ISP RAS — SharpChecker development

- Work on symbolic computation in SharpChecker, a C#/.NET static-analysis tool.

July — August 2026

MCST — Compiler Engineering Intern

- Implemented a LICM pass for LLVM 22 in C++; hoisting conditions check loop structure, memory access, side effects, and speculative safety.
- Built a Python harness comparing program behavior before/after optimization and checking transform/no-transform cases.

PROJECTS

LLVM Interprocedural Global IV Optimization

An LLVM 22 pass localizes supported induction-like globals after interprocedural effects/affine analysis and a separate legality decision; APInt models fixed-width wraparound.

29 positive/negative regression cases: global-iv,verify, structural expectations, idempotence, and observable before/after execution comparison.

UniversalToolchain / Wist2

Designed immutable semantic plans binding dependencies, contributions and typed artifact routes before runtime materialization.

Independent plan verification and exact package-manifest identity; 1,324-test manifest at the inspected October 2026 revision.

x86-64 Compiler Backend & Register Allocation

A compiler backend with SSA/CFG validation, liveness/interference analysis, linear-scan and seeded simulated-annealing allocators, spills, phi lowering, and a SysV x86-64 emitter.

An independent assignment verifier plus native-vs-interpreter differential execution; tests cover branches, loops, spills, and parallel phi moves; a separate public native-codegen lineage starts with Visk (2023) and Wist4 (2024).

SKILLS

Compiler / IR

C++23, LLVM 22/IR, APInt, interprocedural analysis, CFG/SSA, dominators, loops, LICM, legality

Program Analysis

C#/.NET, Roslyn ControlFlowGraph, fixed-point data flow, conditional edges, loops/back edges, joins

Backend / Codegen

x86-64, SysV ABI, NASM IA-32, CDECL/x87, liveness/interference, register allocation, spills, phi lowering

Correctness / Tooling

LLVM Verifier, conservative rejection, regression checks, differential execution, Linux, CMake

EDUCATION

HSE University — Software Engineering

RECOGNITION

MCST LLVM/C++ selection

Organizer correspondence: only 5.0/5.0 score and only solution passing all 104/104 tests among 49 submissions.

LangDev'26 — Build the Language, Then Make the Abstractions Disappear

Delivered the UniversalToolchain architecture talk at LangDev on 9 October 2026.

HSE Faculty of Computer Science — open-source projects competition

Selected as one of the winners with UniversalToolchain / Wist2.

Moscow / remote