Mikhail Razakov

Compiler Infrastructure / Language Tooling Engineer

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

I build language infrastructure that separates semantic decisions from execution. I currently work on symbolic computation in SharpChecker at ISP RAS; UniversalToolchain fixes typed backend routes in a checked plan, while PlanFuzz probes configuration failures.

2022: Compilers / VMs · 2023: CIL + native x86-64 · 2024–2025: AST / IR / backends → toolchain · 2026 →: LLVM + program analysis

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

UniversalToolchain / Wist2

Resolve dependencies, conflicts, and backend routes into an immutable LanguagePlan. An independent verifier checks the plan; runtime does not repeat semantic selection.

Current manifest: 1,324/1,324 tests; dedicated checks cover interpreter/CIL parity and configuration integrity before execution.

PlanFuzz

Experimental testing of programs and configuration plans: isolated replay, control/fault cases, and counterexample reduction that preserves failure fingerprints.

Frozen fault/control corpora and replayable artifacts make planning defects inspectable without claiming formal proof.

LLVM Interprocedural Global IV Optimization

An LLVM 22 pass localizes supported induction-like globals after interprocedural effect analysis, affine evolution, and a separate applicability check; APInt models fixed-width wraparound.

29 positive/negative regression cases cover LLVM Verifier checks, structural expectations, idempotence, and observable before/after behavior.

SKILLS

Compiler Infrastructure
dependency/conflict resolution, component/provider selection, deterministic pass ordering, typed artifact routing, pre-execution checks

Compiler / IR
C++23, LLVM 22/IR, CFG/SSA, loop and interprocedural analysis, transformation applicability

Backends / Runtime
.NET, CIL/MSIL, native x86-64, SysV ABI, stack frames/calling conventions, register allocation, interpreter/runtime composition

Verification
regression suites, LLVM Verifier, interpreter/backend parity, differential execution

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.

Earlier distinctions
MEPhI Junior: Information Technologies section winner in 2025 and first-degree diploma in 2026; HSE Vysshaya Proba Industrial Programming diploma holder; Baltic Science and Engineering Competition: first-degree diploma and the "Perfection as Hope" prize.

Moscow / remote