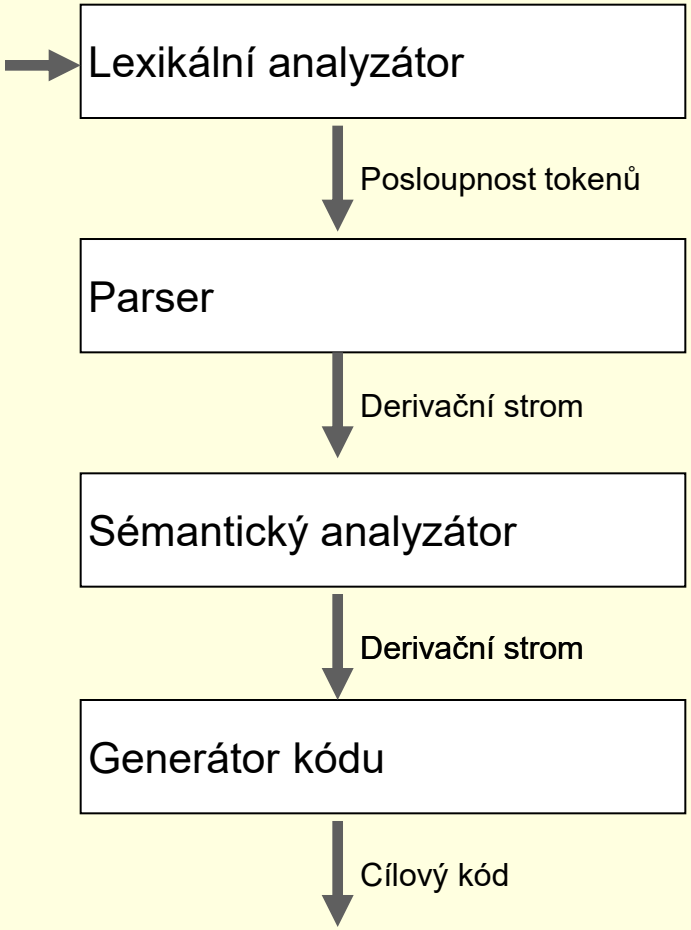
Architektura překladače

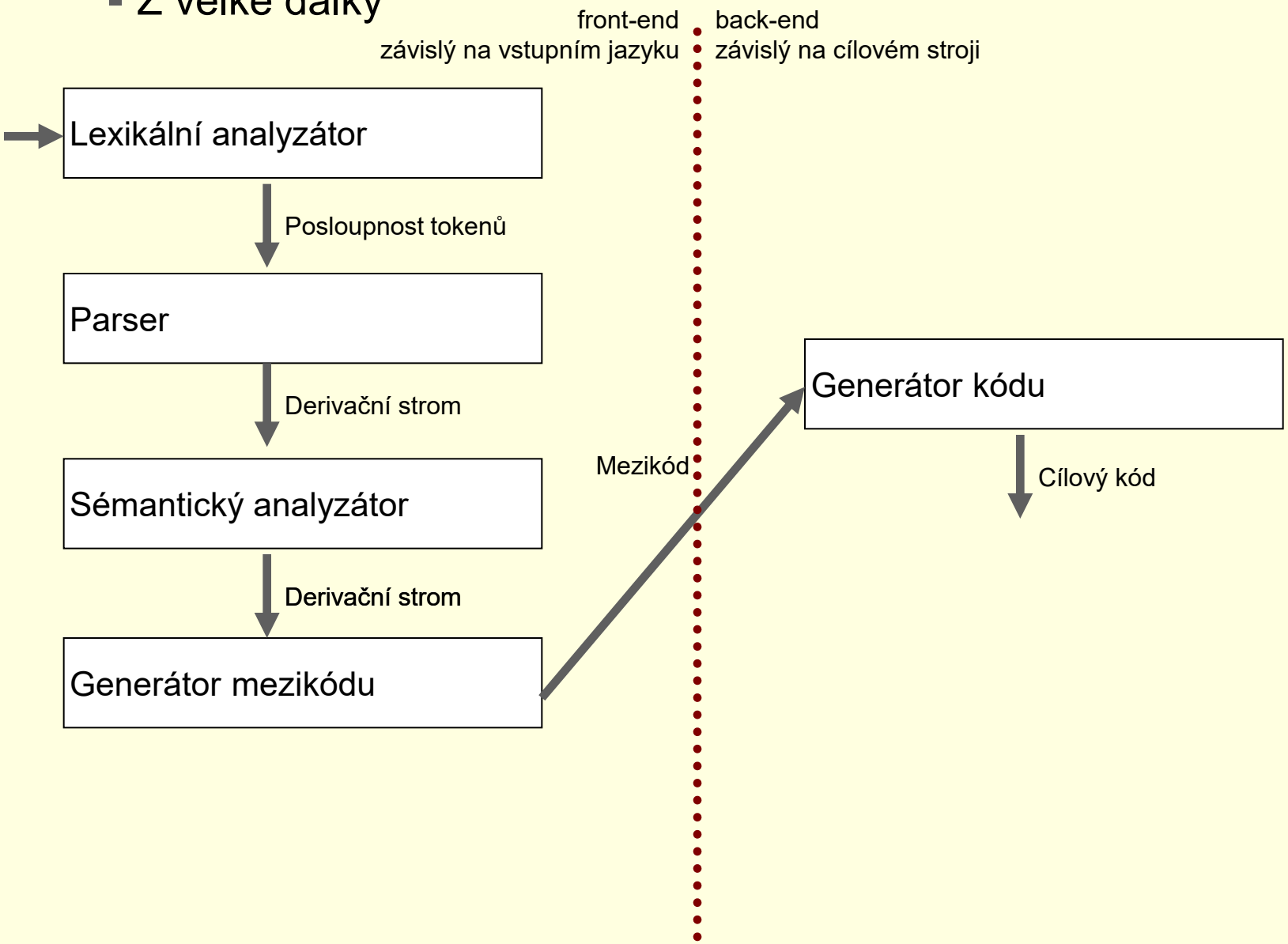
Architektura překladače

- Amatérský pohled



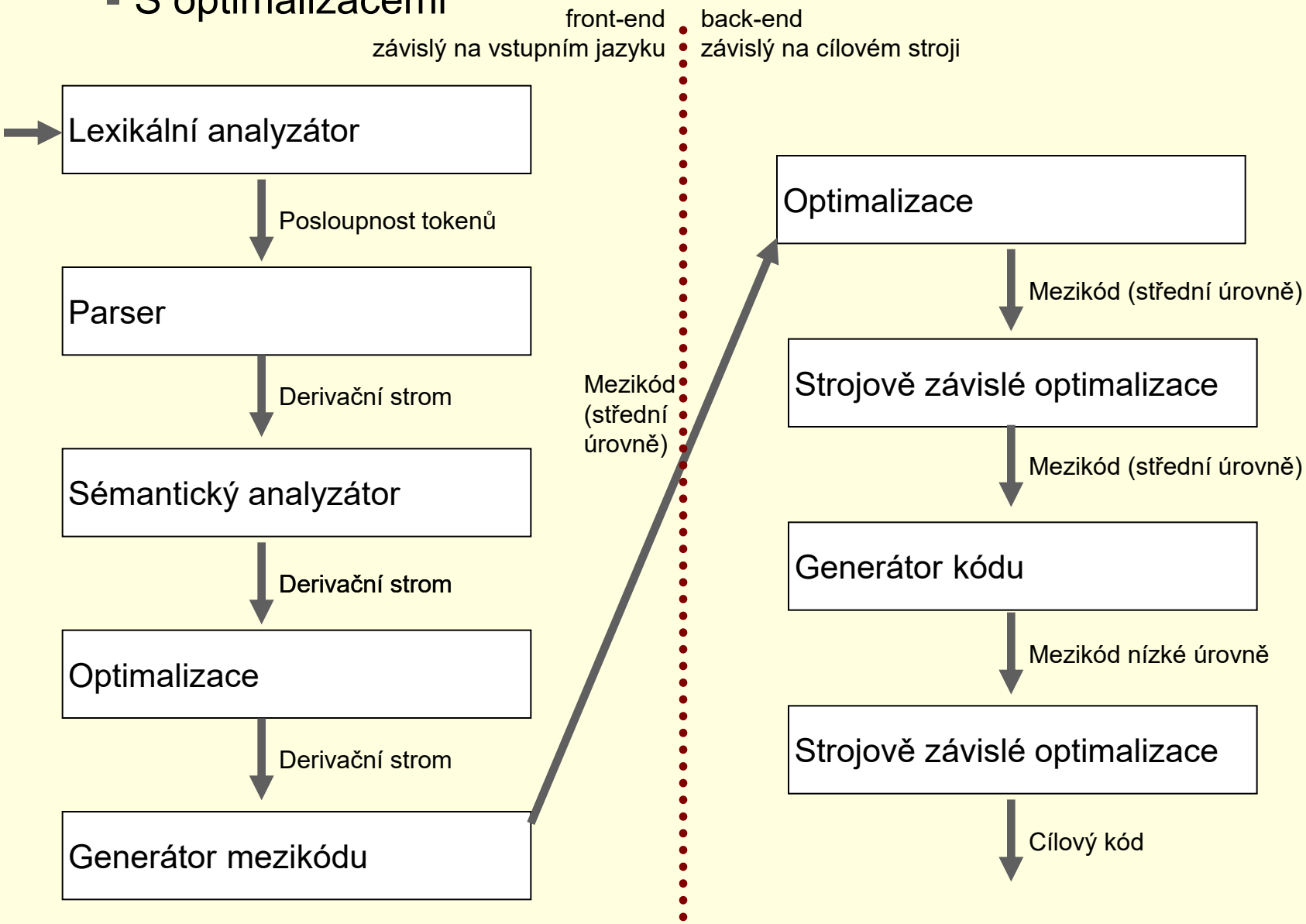
Architektura překladače

- Z velké dálky



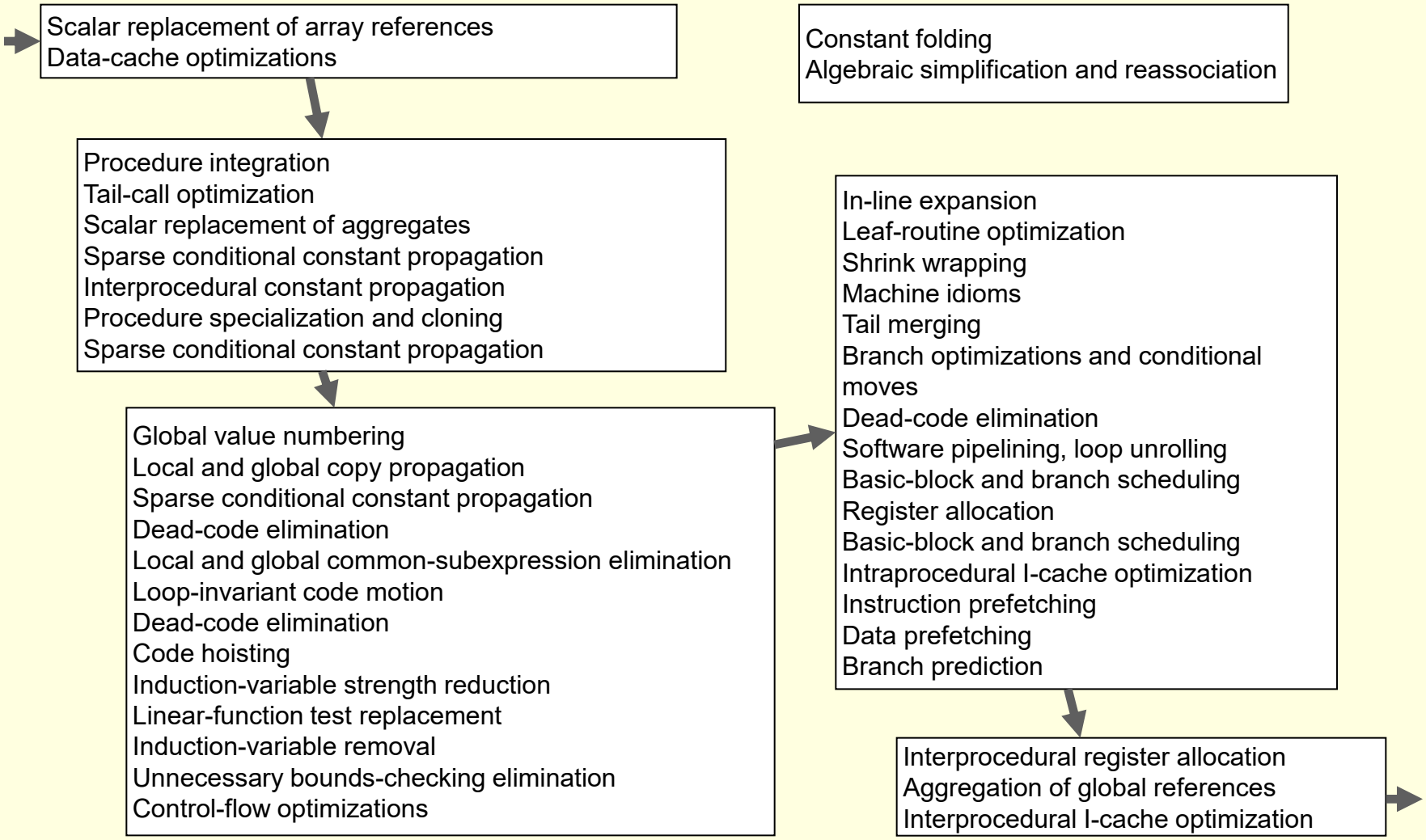
Architektura překladače

- S optimalizacemi



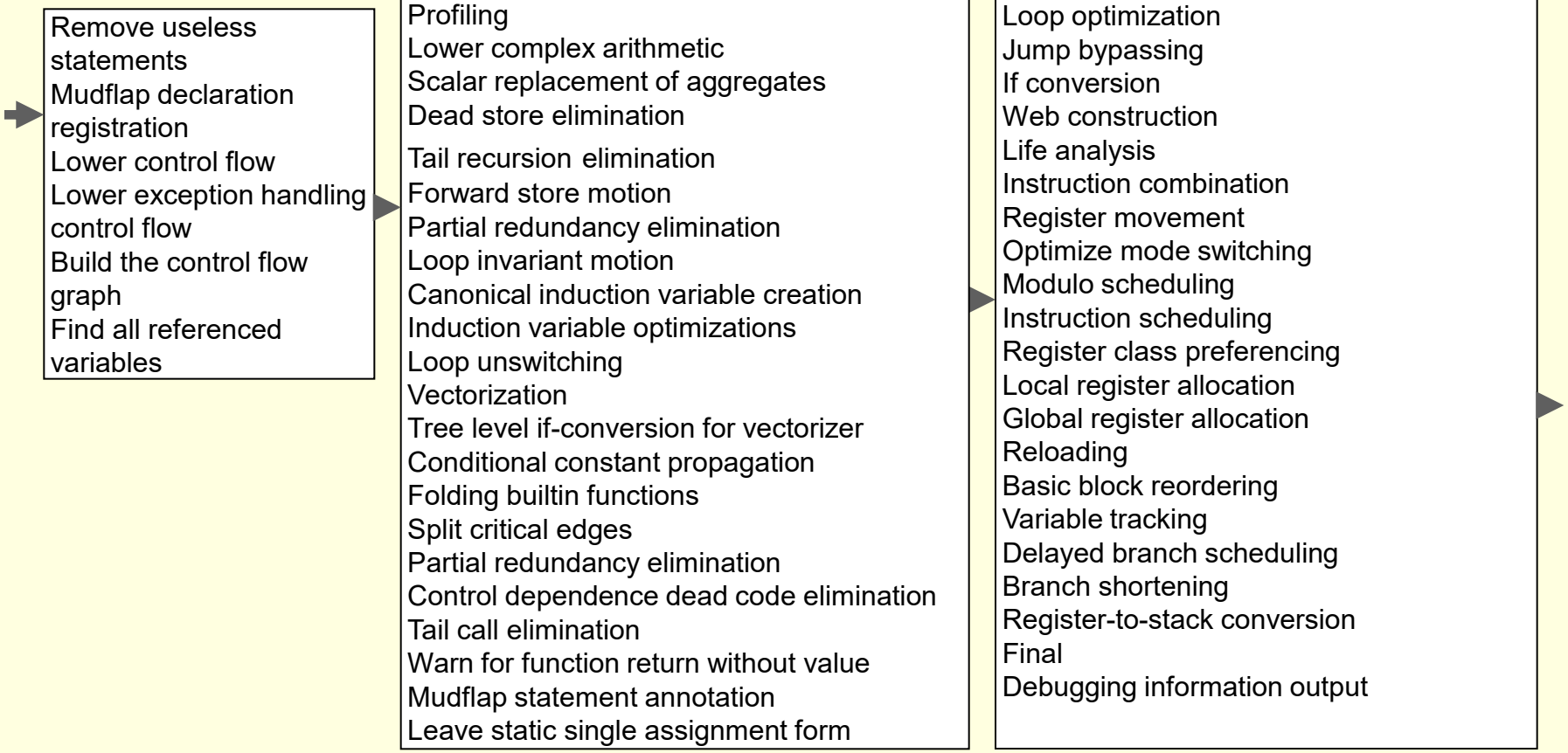
Architektura překladače

- Detailní pohled akademika (pouze optimalizace)
  - Muchnick: Advanced Compiler Design and Implementation



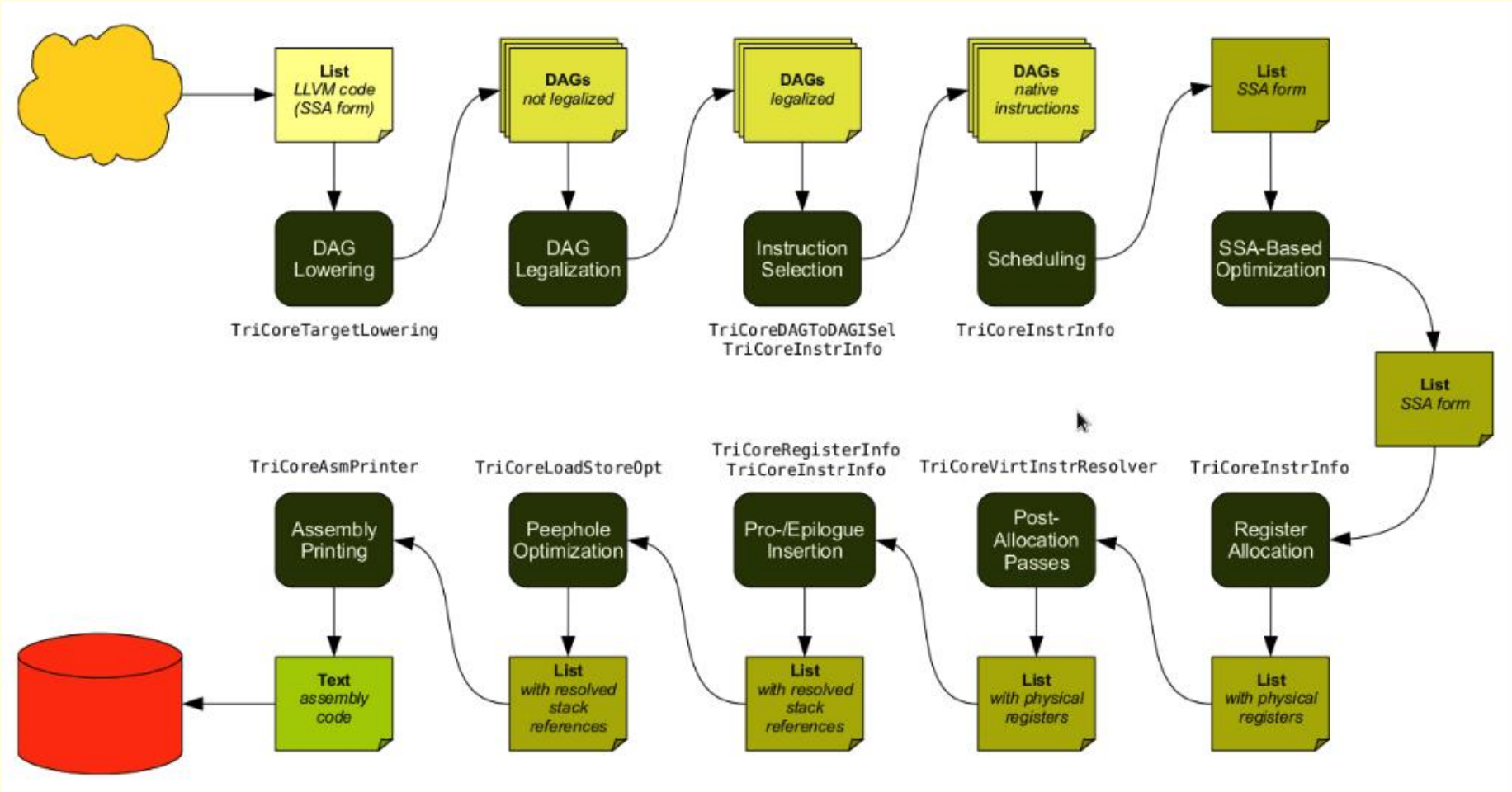
Architektura překladače

- Realita
  - GNU Compiler Collection Internals



Architektura překladače

- Realita
  - LLVM back-end



LLVM 6.0 code generator (AMD64, -O3 -mavx2)

- Instruction Selection
- Expand ISel Pseudo-instructions
- X86 Domain Reassignment Pass
- Tail Duplication
- Optimize machine instruction PHIs
- Merge disjoint stack slots
- Local Stack Slot Allocation
- Remove dead machine instructions
- Early If-Conversion
- Machine InstCombiner
- X86 cmov Conversion
- Machine Loop Invariant Code Motion
- Machine Common Subexpression Elimination
- Machine code sinking
- Peephole Optimizations
- Remove dead machine instructions
- Live Range Shrink
- X86 Fixup SetCC
- X86 LEA Optimize
- X86 Optimize Call Frame
- X86 WinAlloca Expander
- Detect Dead Lanes
- Process Implicit Definitions
- Live Variable Analysis
- Machine Natural Loop Construction
- Eliminate PHI nodes for register allocation
- Two-Address instruction pass
- Simple Register Coalescing
- Rename Disconnected Subregister Components

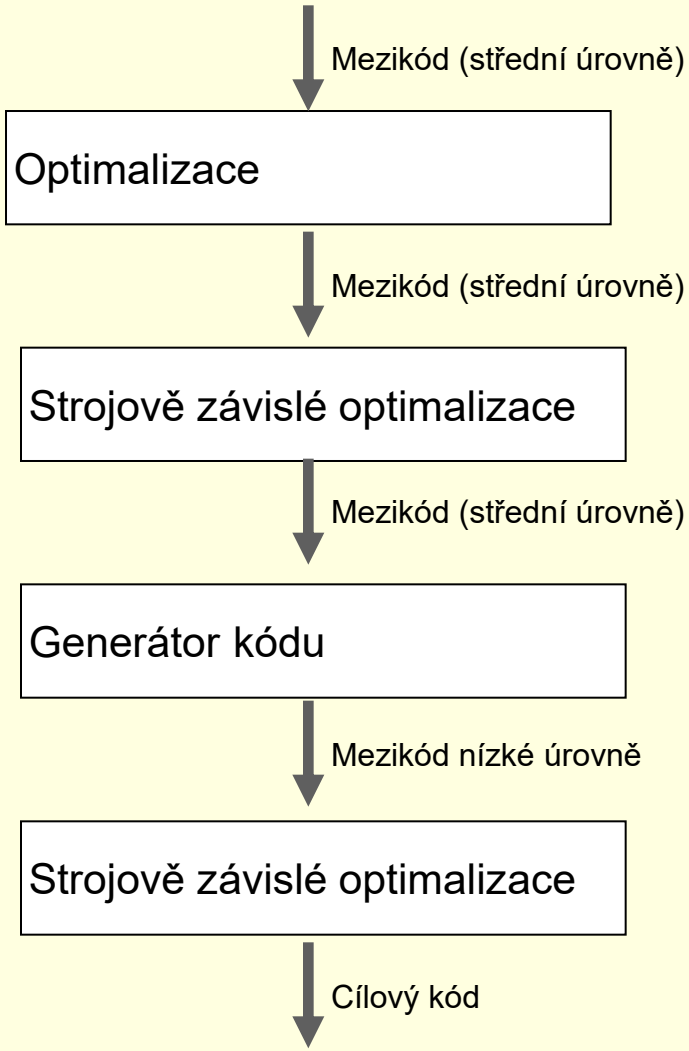
- Machine Instruction Scheduler
- Greedy Register Allocator
- Virtual Register Rewriter
- Stack Slot Coloring
- Machine Loop Invariant Code Motion
- X86 FP Stackifier
- Shrink Wrapping analysis
- Prologue/Epilogue Insertion & Frame Finalization
- Control Flow Optimizer
- Tail Duplication
- Machine Copy Propagation Pass
- Post-RA pseudo instruction expansion pass
- X86 pseudo instruction expansion pass
- Post RA top-down list latency scheduler
- Analyze Machine Code For Garbage Collection
- Branch Probability Basic Block Placement
- X86 Execution Dependency Fix
- X86 vzeroupper inserter
- X86 Byte/Word Instruction Fixup
- X86 Atom pad short functions
- X86 LEA Fixup
- Compressing EVEX instrs to VEX encoding when possible
- Contiguously Lay Out Funclets
- StackMap Liveness Analysis
- Live DEBUG_VALUE analysis
- Insert fentry calls
- Insert XRay ops
- Implement the 'patchable-function' attribute
- X86 Retpoline Thunks

LLVM 18.0 code generator (AMD64, -O3 -mno-sse)

- Target Transform Information
- Target Library Information
- Assumption Cache Tracker
- Target Pass Configuration
- Machine Module Information
- Type-Based Alias Analysis
- Scoped NoAlias Alias Analysis
- Profile summary info
- Create Garbage Collector Module Metadata
- Machine Branch Probability Analysis
- Default Regalloc Eviction Advisor
- Default Regalloc Priority Advisor
  - ModulePass Manager
    - FunctionPass Manager
      - Dominator Tree Construction
      - Basic Alias Analysis (stateless AA impl)
      - Function Alias Analysis Results
      - ObjC ARC contraction
    - Pre-ISel Intrinsic Lowering
    - FunctionPass Manager
      - Expand large div/rem
      - Expand large fp convert
      - Expand Atomic instructions
      - Lower AMX intrinsics
      - Lower AMX type for load/store
      - Dominator Tree Construction
      - Basic Alias Analysis (stateless AA impl)
      - Natural Loop Information
      - Canonicalize natural loops
      - Scalar Evolution Analysis
      - Loop Pass Manager
        - Canonicalize Freeze Instructions in Loops
        - Induction Variable Users
        - Loop Strength Reduction
      - Basic Alias Analysis (stateless AA impl)
      - Function Alias Analysis Results
      - Merge contiguous icmps into a memcmp
      - Natural Loop Information
      - Lazy Branch Probability Analysis
      - Lazy Block Frequency Analysis
      - Expand memcmp() to load/stores
      - Lower Garbage Collection Instructions
      - Shadow Stack GC Lowering
      - Lower constant intrinsics
      - Remove unreachable blocks from the CFG
      - Natural Loop Information
      - Post-Dominator Tree Construction
      - Branch Probability Analysis
      - Block Frequency Analysis
      - Constant Hoisting
      - Replace intrinsics with calls to vector library
      - Partially inline calls to library functions
      - Expand vector predication intrinsics
      - Scalarize Masked Memory Intrinsics
      - Expand reduction intrinsics
      - Natural Loop Information
      - TLS Variable Hoist
      - Interleaved Access Pass
      - X86 Partial Reduction
      - Expand indirectbr instructions
      - Natural Loop Information
      - CodeGen Prepare
      - Dominator Tree Construction
      - Exception handling preparation
      - Prepare callbr
      - Safe Stack instrumentation pass
      - Insert stack protectors
      - Basic Alias Analysis (stateless AA impl)
      - Function Alias Analysis Results
      - Natural Loop Information
      - Post-Dominator Tree Construction
      - Branch Probability Analysis
      - Assignment Tracking Analysis
      - Lazy Branch Probability Analysis
      - Lazy Block Frequency Analysis
      - X86 DAG->DAG Instruction Selection
      - MachineDominator Tree Construction
      - Local Dynamic TLS Access Clean-up
      - X86 PIC Global Base Reg Initialization
      - Argument Stack Rebase
      - Finalize ISel and expand pseudo-instructions
      - X86 Domain Reassignment Pass
      - Lazy Machine Block Frequency Analysis
      - Early Tail Duplication
      - Optimize machine instruction PHIs
      - Slot index numbering
      - Merge disjoint stack slots
      - Local Stack Slot Allocation
      - Remove dead machine instructions
      - MachineDominator Tree Construction
      - Machine Natural Loop Construction
      - Machine Trace Metrics
      - Early If-Conversion
      - Lazy Machine Block Frequency Analysis
      - Machine InstCombiner
      - X86 cmov Conversion
      - MachineDominator Tree Construction
      - Machine Natural Loop Construction
      - Machine Block Frequency Analysis
      - Early Machine Loop Invariant Code Motion
      - MachineDominator Tree Construction
      - Machine Block Frequency Analysis
      - Machine Common Subexpression Elimination
      - MachinePostDominator Tree Construction
      - Machine Cycle Info Analysis
      - Machine code sinking
      - Peephole Optimizations
      - Remove dead machine instructions
      - Live Range Shrink
      - X86 Fixup SetCC
      - Lazy Machine Block Frequency Analysis
      - X86 LEA Optimize
      - X86 Optimize Call Frame
      - X86 Avoid Store Forwarding Blocks
      - X86 speculative load hardening
      - MachineDominator Tree Construction
      - X86 EFLAGS copy lowering
      - X86 DynAlloca Expander
      - MachineDominator Tree Construction
      - Machine Natural Loop Construction
      - Tile Register Pre-configure
      - Detect Dead Lanes
      - Process Implicit Definitions
      - Remove unreachable machine basic blocks
      - Live Variable Analysis
      - Eliminate PHI nodes for register allocation
      - Two-Address instruction pass
      - Slot index numbering
      - Live Interval Analysis
      - Register Coalescer
      - Rename Disconnected Subregister Components
      - Machine Instruction Scheduler
      - Machine Block Frequency Analysis
      - Debug Variable Analysis
      - Live Stack Slot Analysis
      - Virtual Register Map
      - Live Register Matrix
      - Bundle Machine CFG Edges
      - Spill Code Placement Analysis
      - Lazy Machine Block Frequency Analysis
      - Machine Optimization Remark Emitter
      - Greedy Register Allocator
      - Tile Register Configure
      - Greedy Register Allocator
      - Virtual Register Rewriter
      - Register Allocation Pass Scoring
      - Stack Slot Coloring
      - Machine Copy Propagation Pass
      - Machine Loop Invariant Code Motion
      - X86 Lower Tile Copy
      - Bundle Machine CFG Edges
      - X86 FP Stackifier
      - MachineDominator Tree Construction
      - Machine Dominance Frontier Construction
      - X86 Load Value Injection (LVI) Load Hardening
      - Remove Redundant DEBUG_VALUE analysis
      - Fixup Statepoint Caller Saved
      - PostRA Machine Sink
      - Machine Block Frequency Analysis
      - MachinePostDominator Tree Construction
      - Lazy Machine Block Frequency Analysis
      - Machine Optimization Remark Emitter
      - Shrink Wrapping analysis
      - Prologue/Epilogue Insertion & Frame Finalization
      - Machine Late Instructions Cleanup Pass
      - Control Flow Optimizer
      - Lazy Machine Block Frequency Analysis
      - Tail Duplication
      - Machine Copy Propagation Pass
      - Post-RA pseudo instruction expansion pass
      - X86 pseudo instruction expansion pass
      - Insert KCFI indirect call checks
      - MachineDominator Tree Construction
      - Machine Natural Loop Construction
      - Post RA top-down list latency scheduler
      - Analyze Machine Code For Garbage Collection
      - Machine Block Frequency Analysis
      - MachinePostDominator Tree Construction
      - Branch Probability Basic Block Placement
      - Insert fentry calls
      - Insert XRay ops
      - Implement the 'patchable-function' attribute
      - ReachingDefAnalysis
      - X86 Execution Dependency Fix
      - BreakFalseDeps
      - X86 Indirect Branch Tracking
      - X86 vzeroupper inserter
      - MachineDominator Tree Construction
      - Machine Natural Loop Construction
      - Lazy Machine Block Frequency Analysis
      - X86 Byte/Word Instruction Fixup
      - Lazy Machine Block Frequency Analysis
      - X86 Atom pad short functions
      - X86 LEA Fixup
      - X86 Fixup Inst Tuning
      - X86 Fixup Vector Constants
      - Compressing EVEX instrs when possible
      - X86 Discriminate Memory Operands
      - X86 Insert Cache Prefetches
      - X86 insert wait instruction
      - Contiguously Lay Out Funclets
      - StackMap Liveness Analysis
      - Live DEBUG_VALUE analysis
      - Machine Sanitizer Binary Metadata
      - Lazy Machine Block Frequency Analysis
      - Machine Optimization Remark Emitter
      - Stack Frame Layout Analysis
      - X86 Speculative Execution Side Effect Suppression
      - X86 Indirect Thunks
      - X86 Return Thunks
      - Check CFA info and insert CFI instructions if needed
      - X86 Load Value Injection (LVI) Ret-Hardening
      - Pseudo Probe Inserter
      - Unpack machine instruction bundles
      - Lazy Machine Block Frequency Analysis
      - Machine Optimization Remark Emitter
      - X86 Assembly Printer
      - Free MachineFunction

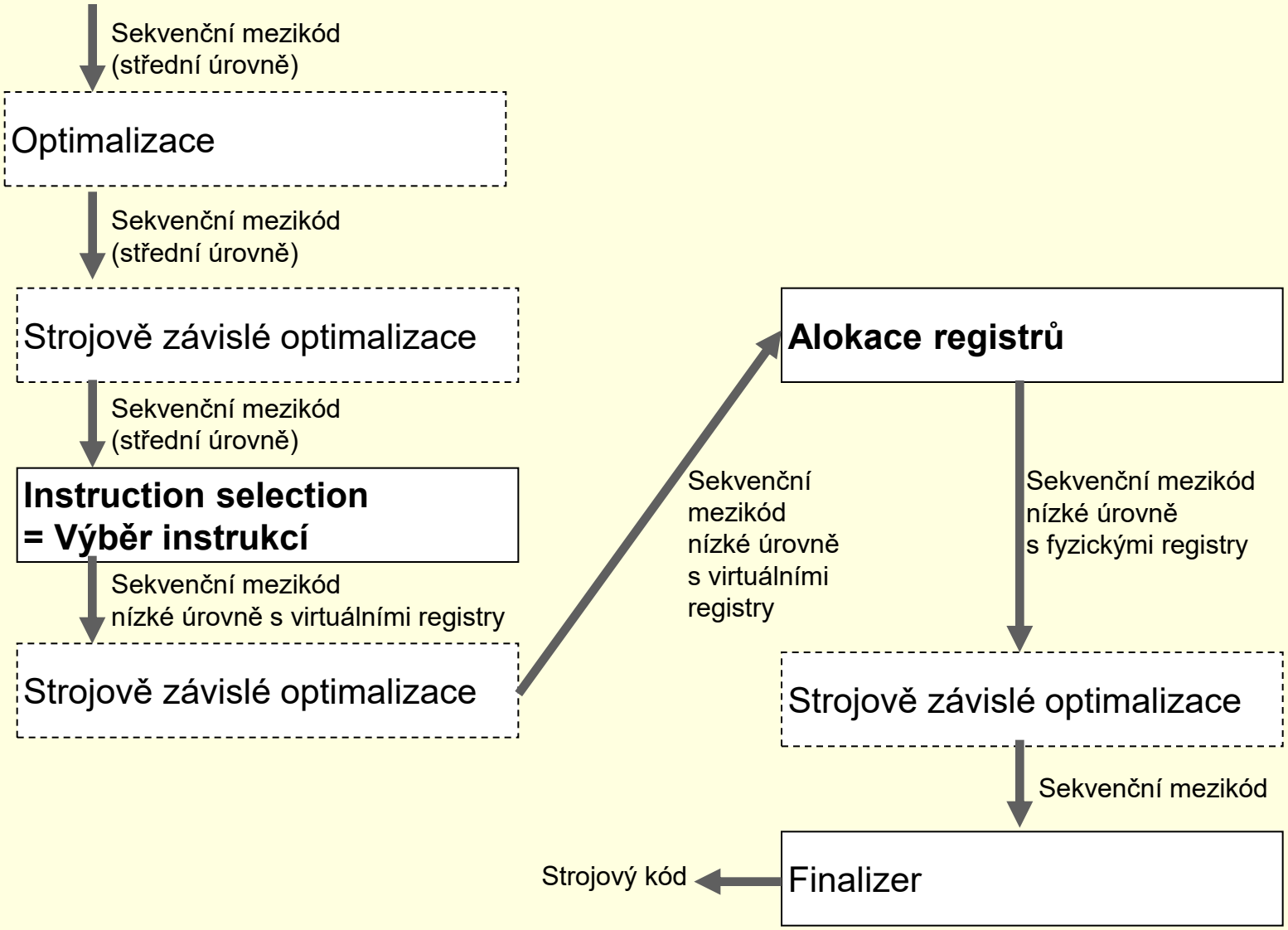
Architektura back-endu

- **Různé vnitřní reprezentace**
  - Mezikód střední úrovně
    - Nezávislá sada operací
      - ADD_I32 a,b,c
    - Forma
      - Nesekvenční
      - Částečně sekvenční
  - Mezikód nízké úrovně
    - Ekvivalenty strojových instrukcí
      - add r1,r2
    - Forma
      - Nesekvenční
      - Částečně sekvenční
      - Sekvenční



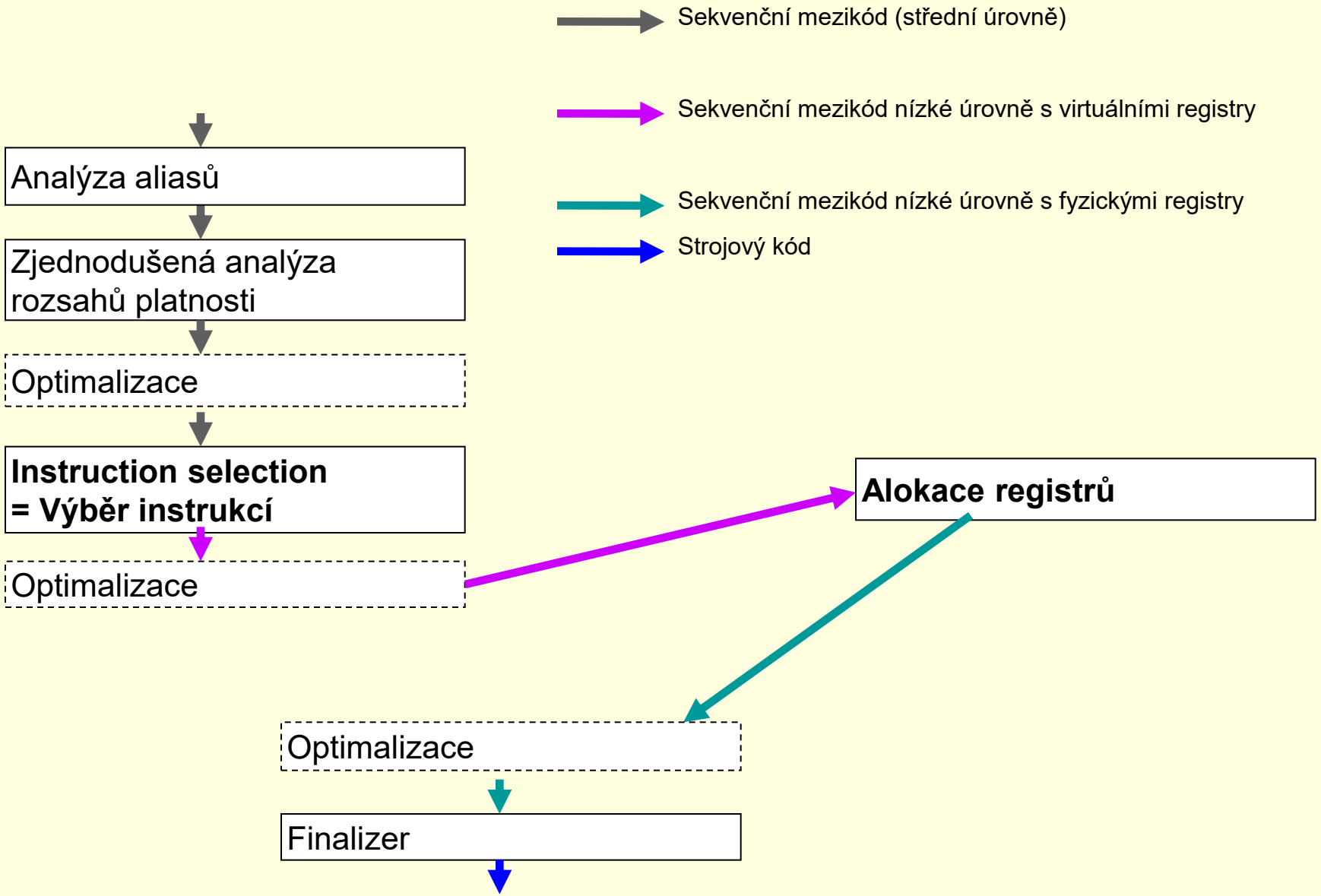
Architektura back-endu

- Sekvenční mezikód



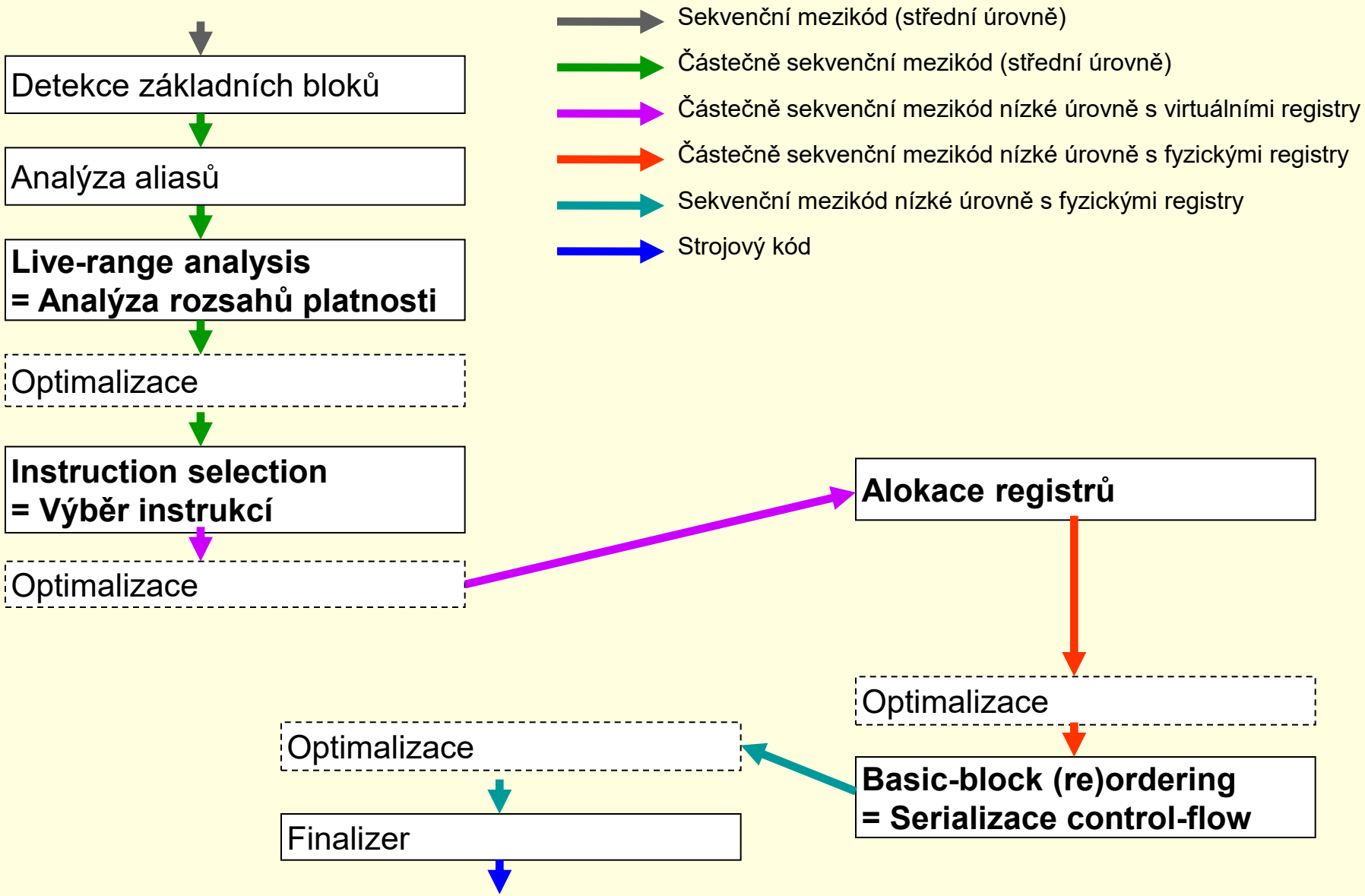
Architektura back-endu

- Sekvenční mezikód



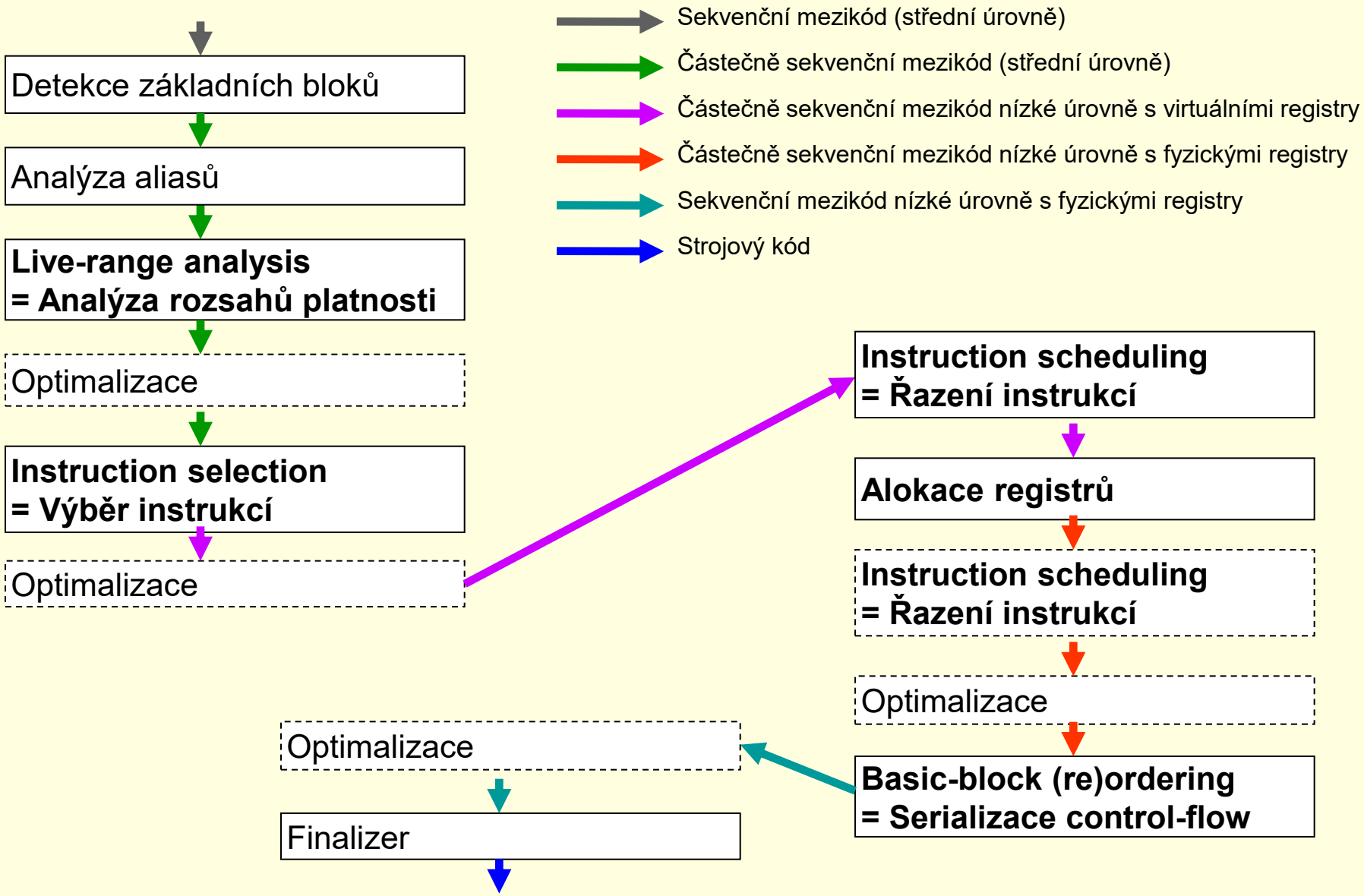
Architektura back-endu

- Částečně sekvenční mezikód bez schedulingu



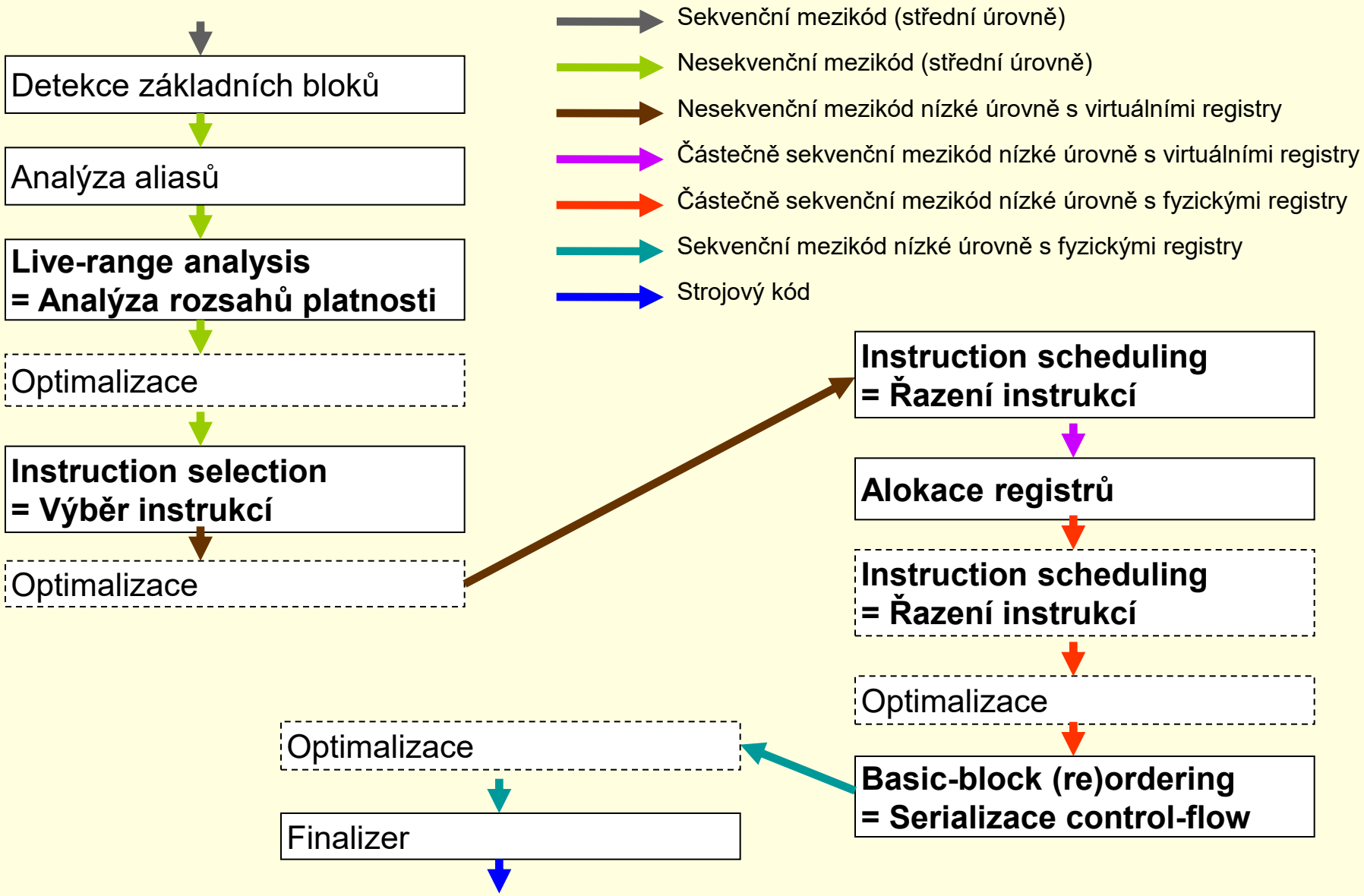
Architektura back-endu

- Částečně sekvenční mezikód se schedulingem



Architektura back-endu

- Nesekvenční mezikód



Architektura back-endu

- Instruction selection
  - Výběr strojových instrukcí
    - 1:n – přímočaré řešení
    - m:n – stromové/grafové gramatiky apod.
  - Vliv na kvalitu kódu poklesl
    - RISC, load-store kód apod.
- Instruction scheduling
  - Řazení instrukcí pro lepší využití ILP (instruction-level parallelism)
    - NP-úplná úloha
  - Lokální v BB
    - Speciální řešení smyček (software pipelining)
    - Částečně globální varianty (trace scheduling)
  - Zrychluje kód o 30-150%
- Register allocation
  - Přidělování fyzických registrů
    - NP-úplná úloha
    - Standardní řešení: Barvení grafu