

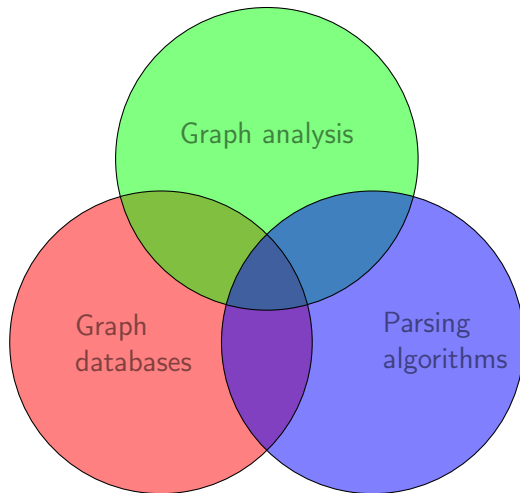


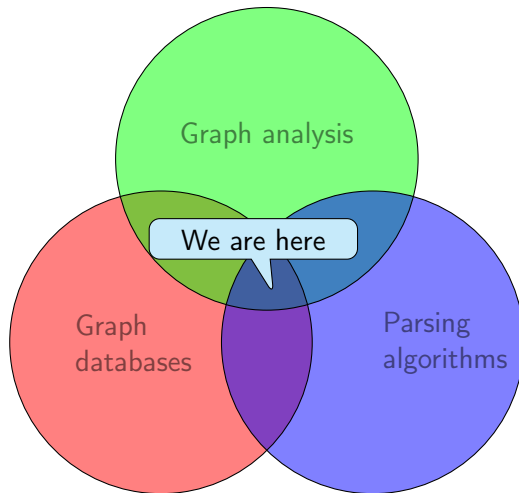
Formal Language Driven Data Analysis Research Group Report

Semyon Grigorev

Saint Petersburg State University

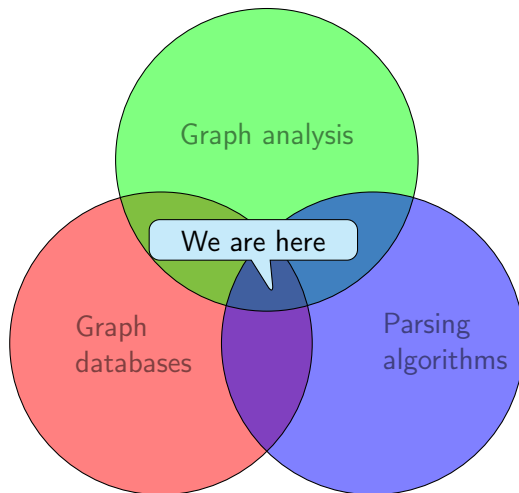
September 14, 2022





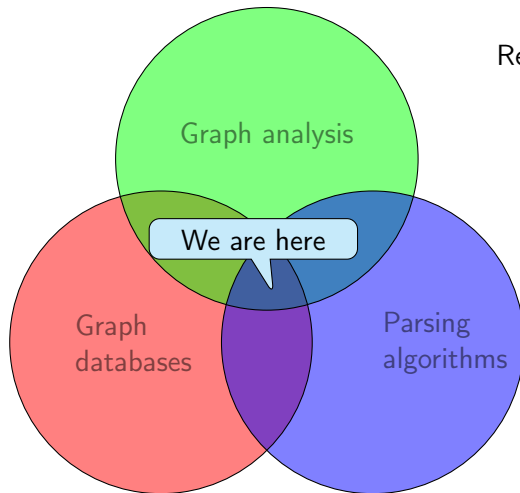
Applications

- Code analysis
- Code querying
- Code parsing



Applications

- Code analysis
- Code querying
- Code parsing



Research directions

- Graph algorithms
 - ▶ Dynamic graphs
 - ▶ Linear algebra
 - ▶ Path querying
- Formal languages
 - ▶ Languages classes and properties
 - ▶ Parsing algorithms
 - ▶ Formal language constrained path querying

Huge software projects

- Millions LOC
- Complex structure
- Dynamic (IDE-level analysis)

Code Analysis and Querying

Huge software projects

- Millions LOC
- Complex structure
- Dynamic (IDE-level analysis)



Huge graphs for analysis

- Millions of vertices
- Complex structure
- Dynamic

Code Analysis and Querying

Huge software projects

- Millions LOC
- Complex structure
- Dynamic (IDE-level analysis)



Huge graphs for analysis

- Millions of vertices
- Complex structure
- Dynamic



Graph storage

- Graph representation
- Query languages
- Query evaluation engines

Code Analysis and Querying

Huge software projects

- Millions LOC
- Complex structure
- Dynamic (IDE-level analysis)



Huge graphs for analysis

- Millions of vertices
- Complex structure
- Dynamic



Graph storage

- Graph representation
- Query languages
- Query evaluation engines



Graph analysis algorithms

- Performance
- Nontrivial techniques (esp. for dynamic graphs)

Code Analysis and Querying

Huge software projects

- Millions LOC
- Complex structure
- Dynamic (IDE-level analysis)

Huge graphs for analysis

- Millions of vertices
- Complex structure
- Dynamic

Graph storage

- Graph representation
- Query languages
- Query evaluation engines

Linear algebra (GraphBLAS)

- Parallel (multicore CPU, GPGPU)
- Flexible, expressive

Graph analysis algorithms

- Performance
- Nontrivial techniques (esp. for dynamic graphs)

Code parsing (for IDE)

Parsing for IDE

- Frequent code updates
- Partially correct code
- Multiple languages support
- Performance-critical

Code parsing (for IDE)

Parsing for IDE

- Frequent code updates
- Partially correct code
- Multiple languages support
- Performance-critical



Parsing technique

- Error recovery
- Reparsing
- Performance
- Flexibility

Code parsing (for IDE)

Parsing for IDE

- Frequent code updates
- Partially correct code
- Multiple languages support
- Performance-critical



Language description

- Modern syntax support (ambiguity, formatting-sensitivity)
- Human-friendly

Parsing technique

- Error recovery
- Reparsing
- Performance
- Flexibility



Code parsing (for IDE)

Parsing for IDE

- Frequent code updates
- Partially correct code
- Multiple languages support
- Performance-critical



Language description

- Modern syntax support (ambiguity, formatting-sensitivity)
- Human-friendly



Parsing technique

- Error recovery
- Reparsing
- Performance
- Flexibility



Advanced parsing algorithms

- New formal classes of languages
- Error recovery
- Incrementalization
- Performance



Results

Graph analysis for symbolic execution engine

Research prototype

- Graph extraction and update mechanism
 - Constrained shortest paths for dynamic graph
-

Results

Graph analysis for symbolic execution engine

Research prototype

- Graph extraction and update mechanism
- Constrained shortest paths for dynamic graph

Graph querying algorithms

Research prototype

- New algorithms
 - Complexity analysis
 - Performance analysis
-

Results

Graph analysis for symbolic execution engine

Research prototype

- Graph extraction and update mechanism
- Constrained shortest paths for dynamic graph

Graph querying algorithms

Research prototype

- New algorithms
- Complexity analysis
- Performance analysis

Sparse linear algebra library on GPGPU

Research prototype

- Operations implementation
- Optimizations
- Performance analysis

- ✓ Linear algebra based graph analysis algorithms research and development
 - ▶ Regular path querying
 - ▶ Context-free path querying
 - ▶ Multiple context-free path querying
- ✓ Graph analysis algorithms evaluation
 - ▶ Formal language constrained path querying
 - ▶ Static code analysis cases
 - ▶ Linear algebra based algorithms

✓ Linear algebra based graph analysis algorithms research and development

- ▶ Regular path querying
- ▶ Context-free path querying
- ▶ Multiple context-free path querying

✓ Graph analysis algorithms evaluation

- ▶ Formal language constrained path querying
- ▶ Static code analysis cases
- ▶ Linear algebra based algorithms

⌚ Code analysis specific dataset for graph analysis algorithms evaluation and comparison

- ▶ New graphs
- ▶ New scenarios
- ▶ New query types

⌚ Graph analysis algorithms evaluation and comparison

- ▶ New cases
- ▶ New graphs
- ▶ New algorithms

- ✓ Linear algebra based graph analysis algorithms research and development
 - ▶ Complexity analysis
 - ▶ Specific cases of graphs and language classes
- ✓ Graph analysis algorithms evaluation
 - ▶ Formal language constrained path querying
 - ▶ Static code analysis cases
 - ▶ GLL-based algorithms

- ✓ Linear algebra based graph analysis algorithms research and development
 - ▶ Complexity analysis
 - ▶ Specific cases of graphs and language classes
- ✓ Graph analysis algorithms evaluation
 - ▶ Formal language constrained path querying
 - ▶ Static code analysis cases
 - ▶ GLL-based algorithms

- ⌚ Dynamic graph analysis
 - ▶ Specific algorithms for symbolic execution engine and code analysis
 - ▶ Theoretical analysis
 - ▶ Performance analysis
- ⌚ Parsing algorithms development and evaluation
 - ▶ Dynamic reparsing
 - ▶ Error recovery

- ✓ Graph analysis algorithms development and evaluation
 - ▶ Linear algebra based algorithm
 - ▶ Static code analysis cases
- ✓ Graph querying algorithm for Neo4j development and evaluation
 - ▶ Static code analysis cases
 - ▶ GLL-based algorithms

- ✓ Graph analysis algorithms development and evaluation
 - ▶ Linear algebra based algorithm
 - ▶ Static code analysis cases
- ✓ Graph querying algorithm for Neo4j development and evaluation
 - ▶ Static code analysis cases
 - ▶ GLL-based algorithms
- **Scholarship will be paused**

- ✓ New linear algebra based algorithm for multiple source regular path querying
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation

✓ New linear algebra based algorithm for multiple source regular path querying

- ▶ Development
- ▶ Implementation
- ▶ Evaluation

⌚ The algorithm improvements

- ▶ Performance analysis and improvements
- ▶ Evaluation and comparison
- ▶ Flexibility improvements

- ✓ Neo4j-based graph analysis algorithm evaluation
 - ▶ General cases
 - ▶ Static code analysis cases

- ✓ Neo4j-based graph analysis algorithm evaluation
 - ▶ General cases
 - ▶ Static code analysis cases
- **Scholarship will be paused**

- ✓ Multiple Context-Free Language
constrained path querying algorithm in
terms of linear algebra
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation

- ✓ Multiple Context-Free Language constrained path querying algorithm in terms of linear algebra
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation
- **Collaboration stopped**

- ✓ GPGPU-based sparse linear algebra library design and implementation
 - ▶ High-level design and architecture of portable extensible library
 - ▶ Project infrastructure
 - ▶ High-level concepts implementation

- ✓ GPGPU-based sparse linear algebra library design and implementation
 - ▶ High-level design and architecture of portable extensible library
 - ▶ Project infrastructure
 - ▶ High-level concepts implementation

- ⌚ Basic low-level primitives implementation
 - ▶ Data structures
 - ▶ Operations
- ⌚ Basic graph algorithms implementation and evaluation
 - ▶ BFS
 - ▶ TC
- ⌚ Out-of-GPGPU-memory graphs handling

- ✓ Kernel fusion optimization for element-wise matrix-matrix operations
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation

- ✓ Kernel fusion optimization for element-wise matrix-matrix operations
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation
- **Scholarship will be paused**

- ✓ Well-typed element-wise matrix-matrix operations: generic generalized kernels for GPPGU
 - ▶ Development
 - ▶ Implementation
 - ▶ Evaluation

- ✓ Well-typed element-wise matrix-matrix operations: generic generalized kernels for GPPGU

- ▶ Development
- ▶ Implementation
- ▶ Evaluation

- ⌚ Applied linear algebra based graph analysis algorithms

- ▶ BFS
- ▶ TC
- ▶ Supplementary matrix-vector operations
- ▶ Implementation and evaluation

- ✓ GPGPU-based operations on vectors and matrices
 - ▶ Collections sort
 - ▶ Matrix transpose

- ✓ GPGPU-based operations on vectors and matrices
 - ▶ Collections sort
 - ▶ Matrix transpose
- ⌚ Generic well-typed matrix-matrix multiplication
 - ▶ Sparse matrices
 - ▶ Evaluation and comparison

Alexandra Istomina (Graph analysis and querying)

- Collaboration research in previous academic year but was paused for summer
- Research topic: graph analysis algorithms for special types of graphs and languages

- Collaboration research in previous academic year but was paused for summer
 - Research topic: graph analysis algorithms for special types of graphs and languages
 - ▶ Code analysis specific cases
- ⌚ Graph analysis algorithms for special types of graphs and languages

Code querying for declarative code analysis

- Code querying and graph querying languages
 - ▶ CodeQL
 - ▶ Datalog
 - ▶ GQL
 - ▶ ...
- Query evaluation engines
 - ▶ Performance
 - ▶ Flexibility
- Graph analysis algorithms
 - ▶ Performance
 - ▶ Scalability
 - ▶ Incrementalization

The Plan

Code querying for declarative code analysis

- Code querying and graph querying languages
 - ▶ CodeQL
 - ▶ Datalog
 - ▶ GQL
 - ▶ ...
- Query evaluation engines
 - ▶ Performance
 - ▶ Flexibility
- Graph analysis algorithms
 - ▶ Performance
 - ▶ Scalability
 - ▶ Incrementalization

Parsing techniques and algorithms

- Language specification formalisms
- Error recovery techniques
- Reparsing techniques

Scholarships request (2022–2023 academic year, 9 months)

Student	Per month	Total (9 months: September – May)
Egor Orachyov	40 000	360 000
Alexandra Istomina	40 000	360 000
Kirill Garbar	20 000	180 000
Denis Porsev	20 000	180 000
Total:	120 000	1 080 000