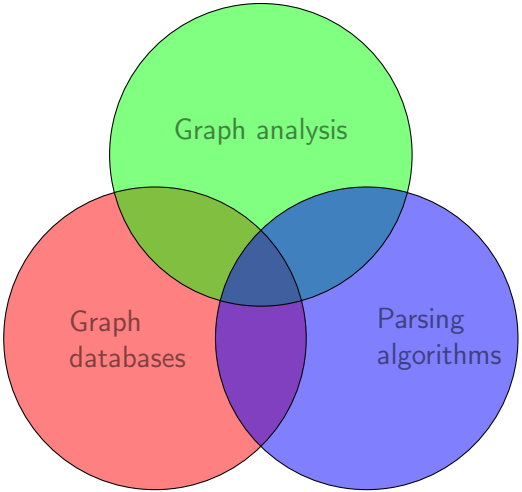
Formal Language Driven Data Analysis Research Group Report

Semyon Grigorev

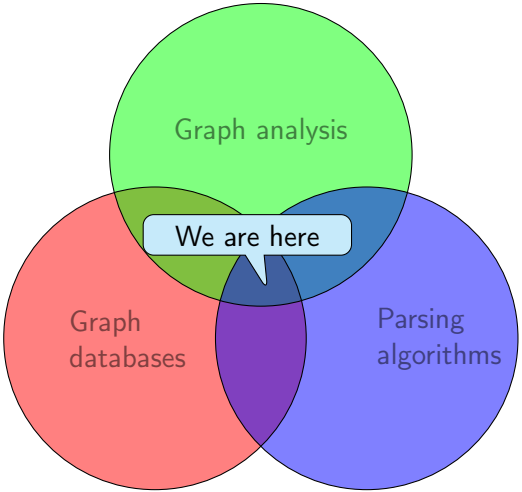
Saint Petersburg State University

September 14, 2022

Research Area



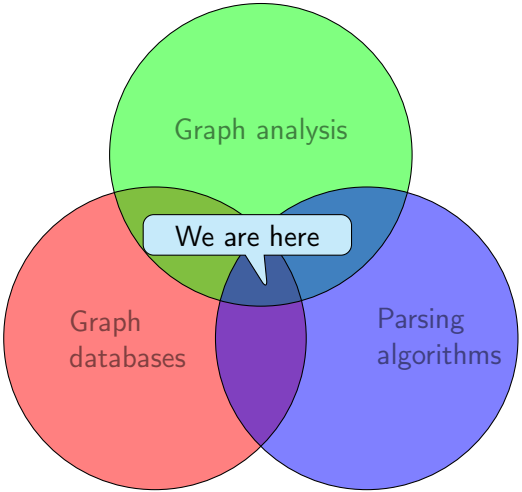
Research Area



Research Area

Applications

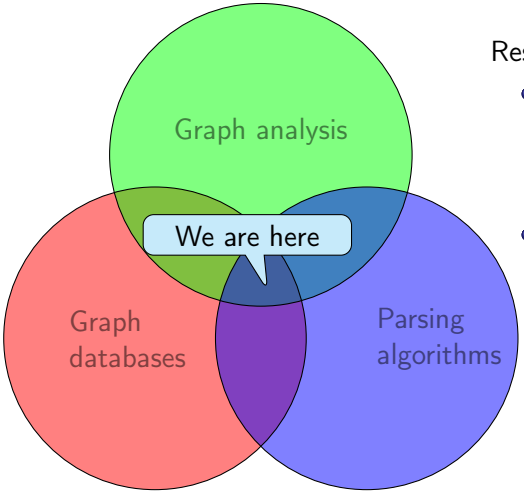
- Code analysis
- Code querying
- Code parsing



Research Area

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

Code Analysis and 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

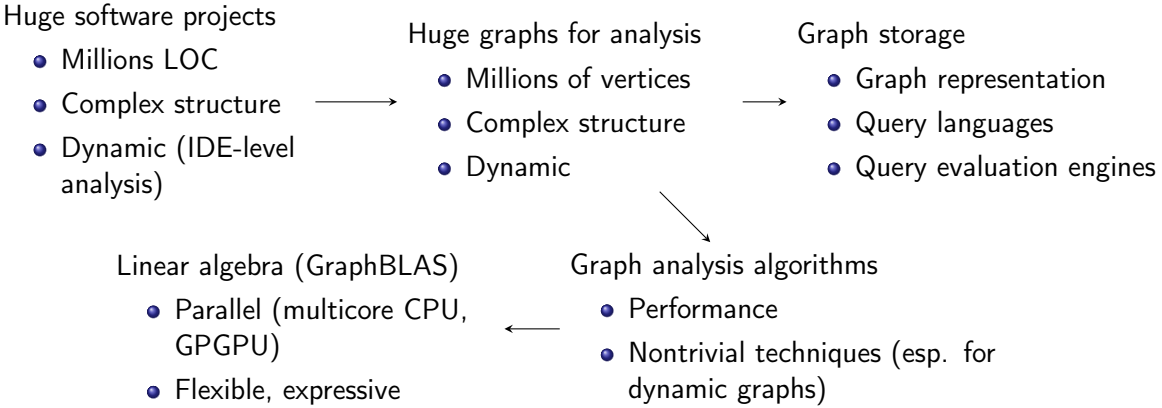
- Graph representation
- Query languages
- Query evaluation engines

↘

Graph analysis algorithms

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

Code Analysis and Querying



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

Parsing technique

- Error recovery
- Reparsing
- Performance
- Flexibility

Language description

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

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

Advanced parsing algorithms

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

Language description

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

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

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

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)

- Egor Orachyov
- Alexandra Istomina
- Kirill Garbar
- Denis Porsev
- 2-3 new students