Declarative Code Analysis

Existing Solutions, Challenges and Research Directions

Semyon Grigorev

September 19, 2022

Declarative Code Analysis

What is the goal of analysis?

- **?** Analytics
- **?** Vulnerability detection
- **?** Code smells detection
- **?** ...

Declarative Code Analysis

What is the goal of analysis?

- ? Analytics
- ? Vulnerability detection
- ? Code smells detection
- ? ...

Where the place of developed tool in software development process?

- ? Part of CI
- ? IDE-level analysis
- ? Standalone server-side analysis
- ? ...

Declarative Code Analysis

What is the goal of analysis?

- ? Analytics
- ? Vulnerability detection
- ? Code smells detection
- ? ...

Where the place of developed tool in software development process?

- ? Part of CI
- ? IDE-level analysis
- ? Standalone server-side analysis
- ? ...

Who is a user?

- ? Software architect/analyst
- ? Regular developer
- ? Advanced developer
- ? ...

Declarative Code Analysis

What is the goal of analysis?

- ? Analytics
- ? Vulnerability detection
- ? Code smells detection
- ? ...

Where the place of developed tool in software development process?

- ? Part of CI
- ? IDE-level analysis
- ? Standalone server-side analysis
- ? ...

Who is a user?

- ? Software architect/analyst
- ? Regular developer
- ? Advanced developer
- ? ...

How it should be done?

- ? Information storage
- ? Analysis specification language
- ? Advanced topics
- ? ...

Declarative Code Analysis: How

How it should be done?

- ? Information storage
- ? Analysis specification language
- ? Advanced topics
- ? ...

Declarative Code Analysis: How

How it should be done?

- ? Information storage
- ? Analysis specification language
- ? Advanced topics
- ? ...

Information storage

- Relational database
- Graph database
- Custom problem-specific storage

Declarative Code Analysis: How

How it should be done?

- ? Information storage
- ? Analysis specification language
- ? Advanced topics
- ? ...

Analysis specification language

- Cypher/GQL-like language
- Datalog-like language
- Custom domain-specific language

Information storage

- Relational database
- Graph database
- Custom problem-specific storage

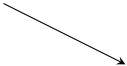
Declarative Code Analysis: How

How it should be done?

- ? Information storage
- ? Analysis specification language
- ? Advanced topics
- ? ...

Information storage

- Relational database
- Graph database
- Custom problem-specific storage



Analysis specification language

- Cypher/GQL-like language
- Datalog-like language
- Custom domain-specific language

Advanced topics

- Dynamic data analysis (incremental analysis)
- Results analysis
- Query debugging
- ...

Infer (Facebook)

- `https://fbinfer.com/`
- General-purpose static code analysis
- Separation logic + abstract interpretation
  - Modular engine
  - Program API (OCaml)
  - Predefined analysis

CodeQL (GitHub/Microsoft)

- https://codeql.github.com/

NG SAST (ShiftLeft)

- `https://www.shiftleft.io/`
- Static application security testing (vulnerability detection)
- Ocular (Joern) as a graph storage and query engine
  - Custom graph database
  - Custom graph query language

Soufflé (Oracle Labs/The University of Sydney)

- `https://souffle-lang.github.io/index.html`
- General-purpose static code analysis
- Logic programming language inspired by Datalog
  - Translation to C++
  - Can use external storages for relations

Soufflé (Oracle Labs/The University of Sydney)

- `https://souffle-lang.github.io/index.html`
- General-purpose static code analysis
- Logic programming language inspired by Datalog
  - Translation to C++
  - Can use external storages for relations
- Query debugging and results analysis (provenance)
- Incrementalization
- Cloud infrastructure

IncA (Johannes Gutenberg University Mainz)

- `https://github.com/szabta89/IncA`
- Incremental static code analysis framework
- Datalog-like DSL
- Aimed to provide IDE-level incremental analysis

ProgQuery

- `https://github.com/OscarRodriguezPrieto/ProgQuery`
- An Efficient and Scalable Platform for Java Source Code Analysis Using Overlaid Graph Representations (2020)
- Neo4j-based
  - Cypher query language
  - Gremlin API
  - Java native API
- Evaluation shows (see paper above)
  - Can be more expressive than CodeQL and other tools
  - Can demonstrates better performance than CodeQL and other tools

Conclusion

- Cypher can be expressive enough against custom and Datalog-like DSLs

Conclusion

- Cypher can be expressive enough against custom and Datalog-like DSLs
- Graph database can be an appropriate storage (even Neo4j)

Conclusion

- Cypher can be expressive enough against custom and Datalog-like DSLs
- Graph database can be an appropriate storage (even Neo4j)
- There is no production ready solutions for IDE-level declarative code analysis

Conclusion

- Cypher can be expressive enough against custom and Datalog-like DSLs
- Graph database can be an appropriate storage (even Neo4j)
- There is no production ready solutions for IDE-level declarative code analysis
- Incremental analysis is a nontrivial challenge

Conclusion

- Cypher can be expressive enough against custom and Datalog-like DSLs
- Graph database can be an appropriate storage (even Neo4j)
- There is no production ready solutions for IDE-level declarative code analysis
- Incremental analysis is a nontrivial challenge
- Query debugging and results analysis is a nontrivial challenge

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison
- Query languages evaluation

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison
- Query languages evaluation
  - Whether advanced DSL needed?

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison
- Query languages evaluation
  - Whether advanced DSL needed?
  - Can GQL be an appropriate language?
  - GQL is SQL for graphs: **ISO standard** for graph query language
  - Cypher-like
  - Friendly to non-advanced users, widely used

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison
- Query languages evaluation
  - Whether advanced DSL needed?
  - Can GQL be an appropriate language?
  - GQL is SQL for graphs: **ISO standard** for graph query language
  - Cypher-like
  - Friendly to non-advanced users, widely used
- Dynamic data analysis
  - Incremental view maintenance
  - Incremental static code analysis
  - Persistent queries
  - ...

Challenges/Research Directions

- Graph databases evaluation
  - Code analysis related scenarios
  - Graph representations comparison
  - Low-level API comparison
- Query languages evaluation
  - Whether advanced DSL needed?
  - Can GQL be an appropriate language?
  - GQL is SQL for graphs: **ISO standard** for graph query language
  - Cypher-like
  - Friendly to non-advanced users, widely used
- Dynamic data analysis
  - Incremental view maintenance
  - Incremental static code analysis
  - Persistent queries
  - ...
- Query debugging and results analysis
  - Appropriate data structures
  - Quick fixes
  - ...