

A Language-Agnostic Framework for Dependency Graph Construction

Francesco Refolli



Outline

- 1 Assessing Software Quality
- 2 The "Tower of Babel" Problem
- 3 A Bit of History
- 4 A New Solution
- 5 Evaluation and Comparison
- 6 Conclusions



Assessing Software Quality

Architectural Smells

God Component

Deep Hierarchy

Wide Hierarchy

Cyclic Hierarchy

Unstable Dependency

Cyclic Dependency

Hub-Like Dependency

The Dependency Graph

The Architectural Smell Graph

The "Tower of Babel" Problem

Language Segmentation

A matter of scale

A matter of perspective

A Bit of History

Manual/Automatic Transpilation

AST/CST Translation

Meta Model Abstraction

Advanced Data Structures

A New Solution

Very different ...

... Or very similar?

Tree Sitter

Why Tree Sitter?

The Approach

Why it works

Shenanigans

An Example

Evaluation and Comparison

The Dependency Detection Benchmark

The Dependency Detection Benchmark (JAVA)

The Dependency Detection Benchmark (RUST)

Comparison with Arcan / Accuracy

Comparison with Arcan / Similarity

Comparison with Arcan / Efficiency

Conclusions

Open Problems

Future Works

Bibliography I

- [1] F. Refolli, D. Sas, and F. A. Fontana, “Lessons learned from implementing a language-agnostic dependency graph parser,” in *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE*, INSTICC, SciTePress, 2025, pp. 484–491, ISBN: 978-989-758-742-9. DOI: 10.5220/0013277600003928.
- [2] K. Weiss and C. Banse, “A language-independent analysis platform for source code,” *CoRR*, 2022. DOI: 10.48550/ARXIV.2203.08424. arXiv: 2203.08424. [Online]. Available: <https://doi.org/10.48550/arXiv.2203.08424>.
- [3] V. J. Marin and C. R. Rivero, “Towards a framework for generating program dependence graphs from source code,” in *Proceedings of the 4th ACM SIGSOFT International Workshop on Software Analytics*, ser. SWAN 2018, Lake Buena Vista, FL, USA: Association for Computing Machinery, 2018, 30–36, ISBN: 9781450360562. DOI: 10.1145/3278142.3278144.

- [4] M. L. Collard, M. J. Decker, and J. I. Maletic, “Srcml: An infrastructure for the exploration, analysis, and manipulation of source code: A tool demonstration,” in *2013 IEEE International conference on software maintenance*, IEEE, 2013, pp. 516–519.
- [5] S. Ducasse, N. Anquetil, M. U. Bhatti, A. Cavalcante Hora, J. Laval, and T. Girba, “MSE and FAMIX 3.0: an Interexchange Format and Source Code Model Family,” Research Report, Nov. 2011. [Online]. Available: <https://inria.hal.science/hal-00646884>.
- [6] G. Antoniol, M. Di Penta, G. Masone, and U. Villano, “Xogastan: Xml-oriented gcc ast analysis and transformations,” in *Proceedings Third IEEE International Workshop on Source Code Analysis and Manipulation*, 2003, pp. 173–182. DOI: 10.1109/SCAM.2003.1238043.
- [7] S. Tichelaar, S. Ducasse, and S. Demeyer, “Famix and xmi,” in *Proceedings Seventh Working Conference on Reverse Engineering*, IEEE, 2000, pp. 296–298. DOI: 10.1109/WCRE.2000.891485.