David Zhao

d-z@outlook.com | Sydney, Australia

Compiler engineer with years of experience in working with large systems, along with programming languages and databases research. Sound understanding of high-level compiler systems and logic programming techniques from experiences working on Soufflé and Rel.

Education

University of Sydney (PhD, 2023)

- PhD in Computer Science. **Supervisor:** Prof. Bernhard Scholz
- **Thesis Title:** "Provenance, Incremental Evaluation, and Debugging in Datalog"

University of Sydney (2018)

- Bachelor of Science (Advanced Mathematics) (Honours)
- Honours Class I and University Medal
- **Thesis Title:** "Large-Scale Provenance for Soufflé", **Supervisor:** Prof. Bernhard Scholz

Academic Achievements and Awards

- **Allan Bromley Prize (2018).** For achieving the highest mark across Honours Theses in Computer Science
- **G.S. Caird Scholarship (2015).** For achieving the highest mark in second year Computer Science

Work Experience

RelationalAI May 2022 - Current
Compiler Engineer

- Demonstrates strong teamwork and communication within the language team to design and implement new language features, fix bugs, and improve robustness of the codebase.
- Communicates and works across teams to understand requirements for various other parts of the system, incorporating these into the designs and implementations.
- Communicates with users of Rel to understand important feature requirements for clients, working to plan and design such features.

RelationalAI March 2021 - July 2021
Software Engineer/Research Intern (Language team)

- Integrated a worst-case optimal join in the Datalog-based type inference and cleaned up legacy type inference code, leading to performance improvements of up to 80×.
- Implemented additions to the Rel standard library, including an algebraic constraints library and various statistical summary functions.
- Improved error reporting in the compiler and other bug fixes.

Oracle Labs December 2017 - April 2018
Research Intern

- Research project to detect malware embedded in PDF documents using abstract interpretation techniques.
- Worked on PDF parsing tools to extract JavaScript embedded in PDF documents.
- Implemented abstractions to enable abstract interpretation for JavaScript features in PDFs, including form handling libraries.
- Results demonstrated improvements in detection accuracy with minimal false negatives, and this was published in the PLAS 2019 workshop.

Optiver November 2016 - March 2017
Software Development Intern

- Created a tool to visualise and analyse performance bottlenecks in the trading pipeline.
- Worked on a tool to detect certain trading patterns which could be exploited to maximise trading effectiveness.
- Analyses led to improvements in Optiver's trading patterns, improving the success rate of trades.

University of Sydney July 2016 - December 2020
Academic Tutor

- Taught undergraduate courses, including INFO1103 (Introduction to Programming), COMP2022 (Formal Languages and Logic), COMP3109 (Programming Languages and Paradigms), COMP3308 (Introduction to Artificial Intelligence), and COMP5703 (IT Capstone Project)

Research Interests

Provenance and Debugging for Datalog. A new approach for debugging Datalog programs by utilizing *provenance* information. Developed a novel hybrid scheme to efficiently encode and query provenance information for Datalog programs, interactively producing proof trees for debugging use cases. Performance is shown to have minimal overhead for Datalog evaluation, while being more generalisable compared to previous approaches. Our provenance method was integrated into the Soufflé Datalog engine.

Furthermore, provenance has been combined with *incremental evaluation* to provide efficient delta debugging to localize and repair input faults for Datalog programs where an unexpected output is introduced during an incremental update.

Incremental Evaluation for Datalog. A new approach for the efficient incremental evaluation of Datalog programs, allowing small changes to input data to be processed effectively without requiring a full re-evaluation of the Datalog program. Proposed a scheme to efficiently encode auxiliary information necessary for the incremental processing of Datalog programs. A working prototype of incremental evaluation was implemented in the Soufflé Datalog engine.

Synthesis of Datalog Programs. A SAT-solving based approach for automatically synthesizing a Datalog program given an input set and an output set. Utilizes provenance-based debugging information to generate constraints which minimize the search space for finding Datalog rules. Demonstrated improved performance compared to existing approaches.

Other Research Experience

Genetic Analysis of Rust Fungus. Bioinformatics project building a pipeline for gene assembly and analysis for *oat stem rust*, a newly sequenced species of rust fungus. Analysed and compared the new species with results for previously analysed species, providing the groundwork for future research.

Publications

- Zhao, D., Subotić, P., Raghothaman, M., and Scholz, B. (2023). Automatic Rollback Suggestions for Incremental Datalog Evaluation. In *Practical Aspects of Declarative Languages (PADL)*. Springer, Cham. doi:10.1007/978-3-031-24841-2_19
- Arch, S., Hu, X., Zhao, D., Subotić, P, Scholz, B. (2022, September). Building a Join Optimizer for Soufflé. In *Logic-Based Program Synthesis and Transformation (LOPSTR)*. Springer, Cham. doi:10.1007/978-3-031-16767-6_5
- Hu, X., Karp, J., Zhao, D., Zreika, A., Wu, X., and Scholz, B. (2021, October). The Choice Construct in the Soufflé Language. In *Asian Symposium on Programming Languages and Systems (APLAS) (pp. 163-181)*. Springer, Cham. doi:10.1007/978-3-030-89051-3_10
- Zhao, D., Subotić, P., Raghothaman, M., and Scholz, B. (2021, September). Towards Elastic Incrementalization for Datalog. In *23rd International Symposium on Principles and Practice of Declarative Programming (PPDP)* (pp. 1-16). doi:10.1145/3479394.3479415
- Hu, X., Zhao, D., Jordan, H. and Scholz, B. (2021, June). An efficient interpreter for Datalog by de-specializing relations. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)* (pp. 681-695). doi:10.1145/3410297
- Zhao, D., Subotić, P. and Scholz, B. (2020). Debugging Large-scale Datalog: A Scalable Provenance Evaluation Strategy. In *ACM Transactions on Programming Languages and Systems (TOPLAS), 42(2)*, (pp. 1-35). doi:10.1145/3379446
- Jordan, H., Subotić, P., Zhao, D. and Scholz, B. (2020). Specializing parallel data structures for Datalog. In *Concurrency and Computation: Practice and Experience*, (p.e5643). doi:10.1002/cpe.5643
- Raghothaman, M., Mendelson, J., Zhao, D., Naik, M. and Scholz, B. (2019). Provenance-guided synthesis of Datalog programs. In *Proceedings of the ACM on Programming Languages, 4 (POPL)*, (pp. 1-27). doi:10.1145/3371130
- Jordan, A., Gauthier, F., Hassanshahi, B. and Zhao, D. (2019, November). Unacceptable Behavior: Robust PDF Malware Detection Using Abstract Interpretation. In *Proceedings of the 14th ACM SIGSAC Workshop on Programming Languages and Analysis for Security* (pp. 19-30). doi:10.1145/3338504.3357341
- Nappa, P., Zhao, D., Subotić, P. and Scholz, B. (2019, September). Fast Parallel Equivalence Relations in a Datalog Compiler. In *2019 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)* (pp. 82-96). IEEE. doi:10.1109/PACT.2019.00015
- Jordan, H., Subotić, P., Zhao, D. and Scholz, B. (2019, February). A specialized B-tree for concurrent datalog evaluation. In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'19)*. ACM, New York, NY, USA. doi:10.1145/3293883.3295719
- Jordan, H., Subotić, P., Zhao, D. and Scholz, B. (2019, February). Brie: A Specialized Trie for Concurrent Datalog. In *Proceedings of the 10th International Workshop on Programming Models and Applications for Multicores and Manycores* (pp. 31-40). ACM. doi:10.1145/3303084.3309490

Presentations

- Zhao, D., 2019. Incremental Datalog Prototype in Soufflé. Presented at *The Second Workshop on Incremental Computing (IC)*.