



Type-Safe Modular Hash-Consing Library in Rust and Haskell

BY KARAN AHLUWALIA
ADVISED BY PROF.
MATTHEW FLUET

Type-Safe Modular Hash-Consing

Proposed Work

Technique to save memory and speed up certain operations by sharing instances of immutable values.

Goal of hash-consing is to optimize memory usage.

How Hash-consing works:

- *Hashing*
 - *Equality Checking*
 - *Sharing*
-
- *Develop a robust and efficient Type-Safe Modular Hash-Consing library in Haskell and Rust using unique features of the languages.*
 - *Demonstrate how Rust and Haskell handle hash-consing in different ways.*
 - *Collect extensive performance and memory usage data for benchmarking and comparison.*

Milestone 1 Goals

Research and Design

- *In-depth research on Type-Safe Modular Hash-Consing principles.*
 - *Study existing implementations and related academic work.*
 - *Formulate a detailed design plan for libraries implementation in Rust and Haskell.*
- *Expected Outcome: A well-defined and documented implementation plan.*

Progress

- *Researched on Type-Safe Modular Hash-Consing principles.*
- *Analyzed existing implementations and academic works related to the topic.*
- *Developed a design plan for the library in Haskell.*
- *Starting working on the Haskell library implementation while ensuring it is both pure and efficient.*

Challenges

- *Determining the appropriate data structures for the library.*
- *Designing the library architecture to align with Haskell's strengths.*
- *Initial design wasn't "Haskelly" enough, prompting a design change after consultation with my advisor.*

Next Step

- *Complete the Haskell library implementation with accompanying tests.*
- *Initiate and finish the Rust library implementation.*
- *Consult with my advisor for review and refinement of the Rust implementation.*

Vision for final completion

- *Comprehensive documentation for both libraries.*
- *Development and execution of a thorough testing strategy.*
- *Performance benchmarking against existing implementations.*
- *Analysis of performance metrics to ensure efficiency and effectiveness.*

Background

- Jean-Christophe Filliâtre and Sylvain Conchon. 2006. Type-safe modular hash-consing. In *Proceedings of the 2006 workshop on ML (ML '06)*. Association for Computing Machinery, New York, NY, USA, 12–19. <https://doi.org/10.1145/1159876.1159880>
- ZHOU, N., & HAVE, C. (2012). Efficient tabling of structured data with enhanced hash-consing. *Theory and Practice of Logic Programming*, 12(4-5), 547-563. [doi:10.1017/S1471068412000178](https://doi.org/10.1017/S1471068412000178)
- Braibant, T., Jourdan, JH., Monniaux, D. (2013). Implementing Hash-Consed Structures in Coq. In: Blazy, S., Paulin-Mohring, C., Pichardie, D. (eds) *Interactive Theorem Proving. ITP 2013*. Lecture Notes in Computer Science, vol 7998. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39634-2_36

Thank you