

Optimizing lazy functional languages: utilizing reference counting for memory reuse

Martin Fredin

June 2023

1 Introduction

All values in purely functional languages are immutable. Immutability is important because it limits shared state, and makes it easier to reason about the program. In practice, this means that a function such as `map` returns a new list instead of modifying the input.

```
map : (A → B) → List A → List B
map f [] = []
map f (x :: xs) = f x :: map f xs
```

This is great if the input list (`List A`) is used later in the program. Otherwise, it is better to reuse the allocation for `List A` by updating the list in place. This would avoid both the deallocation of `List A` and the allocation of `List B`.

Precise reference counting has been proposed to identify when it is possible to update objects in place. Ullrich and de Moura (2021) uses reference counting with borrowed references in the Lean theorem prover. Reinking* et al. (2020) Lorenzen and Leijen (2022) and Lorenzen et al. (2023) also uses reference counting in the Koka language.

Both Lean and Koka are, however, eagerly evaluated. Lazy languages...

2 Graph Reduction Intermediate Notation

3 Extending GRIN with reference counting

4 Result

5 Relevant Work

6 Conclusion and Future Work

References

- Lorenzen, A., & Leijen, D. (2022). Reference counting with frame limited reuse. *Proc. ACM Program. Lang.*, 6(ICFP). <https://doi.org/10.1145/3547634>
- Lorenzen, A., Leijen, D., & Swierstra, W. (2023, May). *Fp²: Fully in-place functional programming* (tech. rep. No. MSR-TR-2023-19). Microsoft. <https://www.microsoft.com/en-us/research/publication/fp2-fully-in-place-functional-programming/>
- Reinking*, A., Xie*, N., de Moura, L., & Leijen, D. (2020, November). *Perceus: Garbage free reference counting with reuse (extended version)* (tech. rep. No. MSR-TR-2020-42) ((*)) The first two authors contributed equally to this work. v4, 2021-06-07. Extended version of the PLDI'21 paper.). Microsoft. <https://www.microsoft.com/en-us/research/publication/perceus-garbage-free-reference-counting-with-reuse/>
- Ullrich, S., & de Moura, L. (2021). Counting immutable beans: Reference counting optimized for purely functional programming. *Proceedings of the 31st Symposium on Implementation and Application of Functional Languages*. <https://doi.org/10.1145/3412932.3412935>