Alternative Methods for Implementing Explicit and Finding Implicit Sharing in embedded DSLs

Curtis D'Alves, Christopher Anand, Lucas Dutton, and Steven Gonder

McMaster University, 1280 Main St W Hamilton, Canada

Abstract. TODO The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

TODO describe sharing problem (mention observable sharing [4] and implicit/-explicit sharing [5] papers)

TODO describe finally tagless [1]

We present methods for implementing embedded DSLs with sharing that are both safe and maintain all the benefits of being embedded in the Haskell ecosystem. This means DSL functions are type-safe, do not require the use of unsafe referencing (i.e., via unsafePerformIO) and can return Haskell's container types (i.e., tuples, lists, etc) without breaking sharing.

2 Detecting Sharing

A naive DSL implementation of an expression in Haskell can be done via standard Haskell data types, for example:

```
data Exp
  = Add Exp Exp
  | Variable String
  | Constant Int

-- Example
v0 = Variable ''v0''
exp0 = Add v0 (Constant 0)
exp1 = Add exp0 exp0
```

Note the DSL generates a tree, or to be more specific an Abstract Syntax Tree (AST).

2.1 Detecting Sharing In Finally Tagless DSLs

2.2 Implicit Sharing Via Hash-Consing

TODO cite Ershov's original description of hash-consing [2] cite Type safe consing implementation (with performance benchmarks) [3]

2.3 Limitations of Hash-Consing

TODO add-chains example

2.4 Explicit Sharing and Limitations

TODO

3 Implicit Sharing Via ByteString ASTs

TODO

3.1 Memory Limitations

TODO

4 Explicit Sharing Of ByteString ASTs

TODO

Acknowledgements Please place your acknowledgments at the end of the paper, preceded by an unnumbered run-in heading (i.e. 3rd-level heading).

References

1. Carette, J., Kiselyov, O., Shan, C.c.: Finally tagless, partially evaluated. In: Asian Symposium on Programming Languages and Systems. pp. 222–238. Springer (2007)
2. Ershov, A.P.: On programming of arithmetic operations. Communications of the ACM **1**(8), 3–6 (1958)
3. Filliâtre, J.C., Conchon, S.: Type-safe modular hash-consing. In: Proceedings of the 2006 Workshop on ML. pp. 12–19 (2006)
4. Gill, A.: Type-safe observable sharing in haskell. In: Proceedings of the 2nd ACM SIGPLAN symposium on Haskell. pp. 117–128 (2009)
5. Kiselyov, O.: Implementing explicit and finding implicit sharing in embedded dsls. arXiv preprint arXiv:1109.0784 (2011)