

Sandy Maguire’s “Brag Sheet”

1. Wrote and self-published *Thinking with Types*, a book on type-level programming in Haskell. It’s sold 1,859 copies, and has a 4.56/5.0 rating on Goodreads (<https://www.goodreads.com/book/show/42736297-thinking-with-types-type-level-programming-in-haskell>)
2. Wrote and self-published *Algebra-Driven Design*, a book on designing software by viewing it as the solution to a set of mathematical equations. It’s sold 1,123 copies, and has a 4.26/5.0 rating on Goodreads (<https://www.goodreads.com/book/show/53001511-algebra-driven-design>)
3. Conceived of and implemented *Wingman for Haskell*, a code synthesizer for the Haskell Language Server, which hooks directly into the GHC API. Performs case splitting, lambda introduction, and has support for automatically synthesizing solutions involving recursion that are guaranteed to terminate. Also, comes with a metaprogramming interface to write tactic scripts for synthesizing more complicated functions. (<https://github.com/haskell/haskell-language-server/tree/master/plugins/hls-tactics-plugin>)
4. Conceived of and implemented *Polysemy*, an effect system based on freer monads and Wu et al.’s paper “Effect Handlers in Scope.” My contributions were to popularize the technique, and realizing most of the boilerplate could be automated away by giving a co-Yoneda encoding of the “weaving” functor. Additionally, required writing a GHC plugin that performs ad-hoc functional dependency resolution to automatically solve ambiguous uses of effects without restricting ourselves to Haskell’s heavy-handed restrictions around fundeps. (<https://github.com/polysemy-research/polysemy>)
5. As part of improving the performance of polysemy, I submitted a patch to GHC’s simplifier, allowing it to specialize dictionaries that exist under lambdas. https://gitlab.haskell.org/ghc/ghc/-/merge_requests/668
6. I’ve submitted a patch to the Agda compiler, implementing much-desired support for running automatic proof search in the context of a co-pattern match. This feature has been missing for at least nine years. (<https://github.com/agda/agda/pull/5827>)
7. I am the author and maintainer of *Cornelis*, the first full-feature implementation of `agda-mode` for vim, allowing would-be users of Agda to not need to learn a new text editor in order to get started. (<https://github.com/isovector/cornelis>)
8. I spent a few months contributing heavily to the Haskell 3d-modeling library *ImplicitCAD*. My contributions involve adding a test suite, removing all sources of partiality, as well as giving an entirely new type model to simplify dimensional-generic shapes. (<https://github.com/Haskell-Things/ImplicitCAD/pulls?q=is%3Apr+author%3Aisovector+is%3Aclosed>)
9. Using *ImplicitCAD*, I designed and built a custom Roomba. It didn’t work very well, but I think it’s more important to fail things sometimes than it is to never try anything hard.

- (<https://reasonablypolymorphic.com/blog/roomba/index.html>)
10. I wrote a scraper and taught myself basic data science in order to write an exploration of Canadian legal case law. (<https://reasonablypolymorphic.com/blog/exploring-case-law/index.html>)
 11. I've been invited to speak at Haskell eXchange and Zurihac, and given a guest lecture at Imperial College London.
 12. I've been reviewing classic CS literature on my website for several years. The latest entries are all formalized in Agda. (<https://reasonablypolymorphic.com/>, search for any article prefixed with "Review:".)
 13. With Li-yao Xia, I'm the author of *lascutter*, a Haskell library that provides a general tree-scraping interface with guaranteed single-pass traversals. Behind the scenes, it uses a novel application of Day convolution to freely reassociate sub-scrapers. Li-yao is responsible for the idea, and me for the implementation. (<https://github.com/isovector/lascutter>)
 14. I'm the author of a little confluent-rule checker. You give it some type signatures and some desired properties about the design surface, and it determines if any of your properties conflict with one another. (<https://github.com/isovector/algebra-checkers>)
 15. I'm the author of a GHC typechecker plugin and library to get type-level sets with $O(\log n)$ lookups (rather than the $O(n)$ we get by pretending lists are sets.) (<https://github.com/isovector/type-sets>)
 16. After being frustrated with Google reliably giving me awful search results, I designed and implemented a new "spatial" search engine for the web, with a custom index and ranking algorithm. (<https://marlo.sandymaguire.me/>)