YOCTO-CFA

by
Leandro Facchinetti

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
August 2020

2020-05-01T14:46:00.321Z

Table of Contents

Bibliography

1. Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley. 2006.
2. *Babel*. `https://babeljs.io`. Accessed 2020-04-06.
3. Emery Berger, Celeste Hollenbeck, Petr Maj, Olga Vitek, and Jan Vitek. *On the Impact of Programming Languages on Code Quality: A Reproduction Study*. ACM Transactions on Programming Languages and Systems. 2019. `https://doi.org/10.1145/3340571`.
4. Gavin Bierman, Martín Abadi, and Mads Torgersen. *Understanding TypeScript*. European Conference on Object-Oriented Programming (ECOOP). 2014.
5. **Leandro Facchinetti**. *Collections Deep Equal*. `https://github.com/leafac/collections-deep-equal`. Accessed 2020-04-01.
6. Matthias Felleisen. *On the Expressive Power of Programming Languages*. Science of Computer Programming. 1991. `https://doi.org/10.1016/0167-6423(91)90036-W`.
7. Matthias Felleisen, Robert Bruce Findler, and Matthew Flatt. *Semantics Engineering with PLT Redex*. The MIT Press. 2009.
8. Matthew Flatt, Robert Bruce Findler, and PLT. *The Racket Guide*. `https://docs.racket-lang.org/guide/`. Accessed 2020-04-13.
9. Mike Grant, Zachary Palmer, and Scott Smith. *Principles of Programming Languages*. 2020.
10. JetBrains. *The State of Developer Ecosystem 2019*. `https://www.jetbrains.com/lp/devecosystem-2019/`. Accessed 2020-01-14.

11. Gilles Kahn. *Natural Semantics*. Annual Symposium on Theoretical Aspects of Computer Science. 1987.
12. Peter Landin. *The Mechanical Evaluation of Expressions*. The Computer Journal. 1964.
13. Bil Lewis, Dan LaLiberte, and Richard Stallman. *GNU Emacs Lisp Reference Manual*. 2015.
14. John McCarthy. *History of LISP*. History of Programming Languages. 1978. https://doi.org/10.1145/800025.1198360.
15. John McCarthy. *Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I*. Communications of the ACM. 1960. https://doi.org/10.1145/367177.367199.
16. Mozilla. *Arrow Function Expressions*. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions. Accessed 2020-01-16.
17. Mozilla. *Destructuring Assignment*. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment. Accessed 2020-01-27.
18. Mozilla. `eval()`. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval. Accessed 2020-02-13.
19. Mozilla. `JSON.stringify()`. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify. Accessed 2020-04-13.
20. Mozilla. Spread Syntax. https://developer.mozilla.org/en-US/

docs/Web/JavaScript/Reference/Operators/Spread_syntax. Accessed 2020-02-03.

21. Matthew Might. *The Language of Languages*. http://matt.might.net/articles/grammars-bnf-ebnf/. Accessed 2020-01-17.
22. Matthew Might, Yannis Smaragdakis, and David Van Horn. *Resolving and Exploiting the k-CFA Paradox: Illuminating Functional vs Object-Oriented Program Analysis*. Programming Language Design and Implementation (PLDI). 2010. https://doi.org/10.1145/1806596.1806631.
23. Gordon Plotkin. *Call-By-Name, Call-By-Value and the λ-Calculus*. Theoretical Computer Science. 1975. https://doi.org/10.1016/0304-3975(75)90017-1.
24. *Prettier*. https://prettier.io. Accessed 2020-02-18.
25. John Reynolds. *Definitional Interpreters for Higher-Order Programming Languages*. Proceedings of the ACM Annual Conference. 1972. https://doi.org/10.1145/800194.805852.
26. Olin Shivers. *Control-Flow Analysis of Higher-Order Languages*. PhD Dissertation, Carnegie Mellon University. 1991.
27. Stack Overflow. *Developer Survey Results 2019*. https://insights.stackoverflow.com/survey/2019. Accessed 2020-01-14.
28. Tom Stuart. *Understanding Computation: From Simple Machines to Impossible Programs*. O'Reilly Media. 2013.
29. Basarat Ali Syed. *TypeScript Deep Dive*. https://basarat.gitbook.io/typescript/. Accessed 2020-01-17.
30. *TypeScript Documentation*. https://www.typescriptlang.org/

`docs`. Accessed 2020-01-17.