

Philip Wadler

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/1254127/publications.pdf>

Version: 2024-02-01

29
papers

2,303
citations

623734

14
h-index

552781

26
g-index

30
all docs

30
docs citations

30
times ranked

691
citing authors

#	ARTICLE	IF	CITATIONS
1	Programming language foundations in Agda. <i>Science of Computer Programming</i> , 2020, 194, 102440.	1.9	6
2	Featherweight go. , 2020, 4, 1-29.		6
3	System F in Agda, for Fun and Profit. <i>Lecture Notes in Computer Science</i> , 2019, , 255-297.	1.3	8
4	Unraveling Recursion: Compiling an IR with Recursion to System F. <i>Lecture Notes in Computer Science</i> , 2019, , 414-443.	1.3	5
5	Programming Language Foundations in Agda. <i>Lecture Notes in Computer Science</i> , 2018, , 56-73.	1.3	6
6	Quantified class constraints. <i>ACM SIGPLAN Notices</i> , 2017, 52, 148-161.	0.2	2
7	Propositions as types. <i>Communications of the ACM</i> , 2015, 58, 75-84.	4.5	100
8	Propositions as sessions. <i>Journal of Functional Programming</i> , 2014, 24, 384-418.	0.8	81
9	Idioms are Oblivious, Arrows are Meticulous, Monads are Promiscuous. <i>Electronic Notes in Theoretical Computer Science</i> , 2011, 229, 97-117.	0.9	31
10	Threesomes, with and without blame. <i>ACM SIGPLAN Notices</i> , 2010, 45, 365-376.	0.2	12
11	The arrow calculus. <i>Journal of Functional Programming</i> , 2010, 20, 51-69.	0.8	19
12	Monadic constraint programming. <i>Journal of Functional Programming</i> , 2009, 19, 663-697.	0.8	26
13	The Girardâ€“Reynolds isomorphism. <i>Information and Computation</i> , 2003, 186, 260-284.	0.7	10
14	Featherweight Java. <i>ACM Transactions on Programming Languages and Systems</i> , 2001, 23, 396-450.	2.1	672
15	Featherweight Java. , 1999, , .		146
16	Featherweight Java. <i>ACM SIGPLAN Notices</i> , 1999, 34, 132-146.	0.2	56
17	Making the future safe for the past. , 1998, , .		276
18	Making the future safe for the past. <i>ACM SIGPLAN Notices</i> , 1998, 33, 183-200.	0.2	39

#	ARTICLE	IF	CITATIONS
19	The call-by-need lambda calculus. <i>Journal of Functional Programming</i> , 1998, 8, 275-317.	0.8	84
20	Why no one uses functional languages. <i>ACM SIGPLAN Notices</i> , 1998, 33, 23-27.	0.2	58
21	EDITORIAL: A HOT opportunity. <i>Journal of Functional Programming</i> , 1997, 7, 127-128.	0.8	1
22	A practical subtyping system for Erlang. <i>ACM SIGPLAN Notices</i> , 1997, 32, 136-149.	0.2	12
23	Call-by-name, Call-by-value, Call-by-need, and the Linear Lambda Calculus. <i>Electronic Notes in Theoretical Computer Science</i> , 1995, 1, 370-392.	0.9	23
24	Monads and composable continuations. <i>Higher-Order and Symbolic Computation</i> , 1994, 7, 39-55.	0.6	47
25	Type systems for object-oriented programming. <i>Journal of Functional Programming</i> , 1994, 4, 125-125.	0.8	3
26	Functional programming in education – Introduction. <i>Journal of Functional Programming</i> , 1993, 3, 3-4.	0.8	3
27	Report on the programming language Haskell. <i>ACM SIGPLAN Notices</i> , 1992, 27, 1-164.	0.2	539
28	Is there a use for linear logic?. <i>ACM SIGPLAN Notices</i> , 1991, 26, 255-273.	0.2	7
29	Fixing some space leaks with a garbage collector. <i>Software - Practice and Experience</i> , 1987, 17, 595-608.	3.6	25