

Matthias Felleisen

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/10997676/publications.pdf>

Version: 2024-02-01

83
papers

4,599
citations

257101

24
h-index

214527

47
g-index

87
all docs

87
docs citations

87
times ranked

814
citing authors

#	ARTICLE	IF	CITATIONS
1	The revised report on the syntactic theories of sequential control and state. Theoretical Computer Science, 1992, 103, 235-271.	0.5	376
2	The essence of compiling with continuations. , 1993, , .		325
3	Classes and mixins. , 1998, , .		311
4	Contracts for higher-order functions. , 2002, , .		283
5	DrScheme: a programming environment for Scheme. Journal of Functional Programming, 2002, 12, .	0.5	214
6	The design and implementation of typed scheme. , 2008, , .		206
7	Reasoning about programs in continuation-passing style. Higher-Order and Symbolic Computation, 1993, 6, 289-360.	1.2	180
8	Hygienic macro expansion. , 1986, , .		173
9	On the expressive power of programming languages. Science of Computer Programming, 1991, 17, 35-75.	1.5	158
10	A syntactic theory of sequential control. Theoretical Computer Science, 1987, 52, 205-237.	0.5	154
11	Interlanguage migration. , 2006, , .		153
12	The call-by-need lambda calculus. Journal of Functional Programming, 1997, 7, 265-301.	0.5	118
13	Languages as libraries. , 2011, , .		100
14	Logical types for untyped languages. , 2010, , .		87
15	The TeachScheme! Project: Computing and Programming for Every Student. Computer Science Education, 2004, 14, 55-77.	2.7	77
16	DrScheme: A pedagogic programming environment for scheme. Lecture Notes in Computer Science, 1997, , 369-388.	1.0	74
17	Control delimiters and their hierarchies. Higher-Order and Symbolic Computation, 1990, 3, 67-99.	1.2	67
18	ViewpointWhy computer science doesn't matter. Communications of the ACM, 2009, 52, 37-40.	3.3	64

#	ARTICLE	IF	CITATIONS
19	Catching bugs in the web of program invariants. , 1996, , .		60
20	A Programmerâ€™s Reduction Semantics for Classes and Mixins. Lecture Notes in Computer Science, 1999, , 241-269.	1.0	60
21	A functional I/O system or, fun for freshman kids. , 2009, , .		58
22	Abstract continuations: a mathematical semantics for handling full jumps. , 1988, , .		57
23	A syntactic theory of sequential state. Theoretical Computer Science, 1989, 69, 243-287.	0.5	53
24	Correct blame for contracts. , 2011, , .		53
25	Componential set-based analysis. ACM Transactions on Programming Languages and Systems, 1999, 21, 370-416.	1.7	51
26	Implementation and use of the PLT scheme Web server. Higher-Order and Symbolic Computation, 2007, 20, 431-460.	0.3	51
27	Complete Monitors for Behavioral Contracts. Lecture Notes in Computer Science, 2012, , 214-233.	1.0	50
28	A programmable programming language. Communications of the ACM, 2018, 61, 62-71.	3.3	45
29	Scheme with Classes, Mixins, and Traits. Lecture Notes in Computer Science, 2006, , 270-289.	1.0	42
30	A functional I/O system or, fun for freshman kids. ACM SIGPLAN Notices, 2009, 44, 47-58.	0.2	40
31	On contract satisfaction in a higher-order world. ACM Transactions on Programming Languages and Systems, 2011, 33, 1-29.	1.7	40
32	Reasoning with continuations II: full abstraction for models of control. , 1990, , .		37
33	Automatically Restructuring Programs for the Web. Automated Software Engineering, 2004, 11, 337-364.	2.2	34
34	The structure and interpretation of the computer science curriculum. Journal of Functional Programming, 2004, 14, 365-378.	0.5	33
35	Programming languages as operating systems (or revenge of the son of the lisp machine). , 1999, , .		32
36	A tail-recursive machine with stack inspection. ACM Transactions on Programming Languages and Systems, 2004, 26, 1029-1052.	1.7	29

#	ARTICLE	IF	CITATIONS
37	Parameter-passing and the lambda calculus. , 1991, , .		28
38	The design and implementation of typed scheme. ACM SIGPLAN Notices, 2008, 43, 395-406.	0.2	28
39	Reasoning about programs in continuation-passing style.. ACM SIGPLAN Lisp Pointers, 1992, V, 288-298.	0.1	26
40	Languages as libraries. ACM SIGPLAN Notices, 2011, 46, 132-141.	0.2	26
41	Modeling an Algebraic Stepper. Lecture Notes in Computer Science, 2001, , 320-334.	1.0	24
42	Modular set-based analysis from contracts. , 2006, , .		24
43	A Visual Environment for Developing Context-Sensitive Term Rewriting Systems. Lecture Notes in Computer Science, 2004, , 301-311.	1.0	23
44	The Call-by-Need Lambda Calculus, Revisited. Lecture Notes in Computer Science, 2012, , 128-147.	1.0	22
45	Super 8 languages for making movies (functional pearl). , 2017, 1, 1-29.		22
46	Adding delimited and composable control to a production programming environment. , 2007, , .		20
47	A reduction semantics for imperative higher-order languages. Lecture Notes in Computer Science, 1987, , 206-223.	1.0	20
48	Optimization coaching. , 2012, , .		19
49	Typing the Numeric Tower. Lecture Notes in Computer Science, 2012, , 289-303.	1.0	19
50	Reflections on Landin's J-operator: A partly historical note. Computer Languages, Systems and Structures, 1987, 12, 197-207.	0.3	18
51	$\hat{\lambda}$ -V-CS: an extended $\hat{\lambda}$ -calculus for scheme. , 1988, , .		17
52	ACL2 in DrScheme. , 2006, , .		17
53	Debugging hygienic macros. Science of Computer Programming, 2010, 75, 496-515.	1.5	15
54	Oh Lord, please don't let contracts be misunderstood (functional pearl). , 2016, , .		12

#	ARTICLE	IF	CITATIONS
55	Toward a formal theory of extensible software. , 1998, , .		10
56	Contracts for First-Class Classes. ACM Transactions on Programming Languages and Systems, 2013, 35, 1-58.	1.7	10
57	Catching bugs in the web of program invariants. ACM SIGPLAN Notices, 1996, 31, 23-32.	0.2	9
58	Componential set-based analysis. ACM SIGPLAN Notices, 1997, 32, 235-248.	0.2	9
59	The essence of compiling with continuations. ACM SIGPLAN Notices, 2004, 39, 502-514.	0.2	8
60	Adding delimited and composable control to a production programming environment. ACM SIGPLAN Notices, 2007, 42, 165-176.	0.2	6
61	Optimization coaching. ACM SIGPLAN Notices, 2012, 47, 163-178.	0.2	5
62	Is continuation-passing useful for data flow analysis?. ACM SIGPLAN Notices, 1994, 29, 1-12.	0.2	4
63	Continuations from generalized stack inspection. ACM SIGPLAN Notices, 2005, 40, 216-227.	0.2	4
64	Debugging macros. , 2007, , .		4
65	Automatic verification for interactive graphical programs. , 2009, , .		4
66	SPCF: its model, calculus, and computational power. Lecture Notes in Computer Science, 1993, , 318-347.	1.0	4
67	Programming languages as operating systems (orrevange of the son of the lisp machine). ACM SIGPLAN Notices, 1999, 34, 138-147.	0.2	4
68	Complete monitors for gradual types. , 2019, 3, 1-29.		4
69	Modeling Web Interactions and Errors. , 2006, , 255-275.		3
70	Feature-Specific Profiling. Lecture Notes in Computer Science, 2015, , 49-68.	1.0	3
71	Fortifying macros. ACM SIGPLAN Notices, 2010, 45, 235-246.	0.2	2
72	TeachScheme!. , 2010, , .		2

#	ARTICLE	IF	CITATIONS
73	ICFP 2002. ACM SIGPLAN Notices, 2013, 48, 34-45.	0.2	2
74	Toward a formal theory of extensible software. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1998, 23, 88-98.	0.5	1
75	Environmental acquisition revisited. ACM SIGPLAN Notices, 2005, 40, 14-25.	0.2	1
76	Nested and Dynamic Contract Boundaries. Lecture Notes in Computer Science, 2010, , 141-158.	1.0	1
77	Program Units as Higher-Order Modules. Electronic Notes in Theoretical Computer Science, 1998, 10, 204-229.	0.9	0
78	TeachScheme!. ACM SIGPLAN Notices, 2010, 45, 129-130.	0.2	0
79	Languages as libraries. ACM SIGPLAN Notices, 2012, 47, 132.	0.2	0
80	Modular set-based analysis from contracts. ACM SIGPLAN Notices, 2006, 41, 218-231.	0.2	0
81	Option contracts. ACM SIGPLAN Notices, 2013, 48, 475-494.	0.2	0
82	On the Orthogonality of Assignments and Procedures in Algol. , 1997, , 101-124.		0
83	Oh Lord, please don't let contracts be misunderstood (functional pearl). ACM SIGPLAN Notices, 2016, 51, 117-131.	0.2	0