Jeremy G Siek

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/7130592/publications.pdf

Version: 2024-02-01

72 papers	1,499 citations	759233 12 h-index	642732 23 g-index
76 all docs	76 docs citations	76 times ranked	401 citing authors

#	Article	IF	Citations
1	Blame and coercion: Together again for the first time. Journal of Functional Programming, 2021, 31, .	0.8	1
2	Parameterized cast calculi and reusable meta-theory for gradually typed lambda calculi. Journal of Functional Programming, 2021, 31, .	0.8	2
3	Programming language foundations in Agda. Science of Computer Programming, 2020, 194, 102440.	1.9	6
4	Extrinsically typed operational semantics for functional languages. , 2020, , .		8
5	Toward efficient gradual typing for structural types via coercions. , 2019, , .		10
6	Gradual typing: a new perspective. , 2019, 3, 1-32.		22
7	Optimizing and evaluating transient gradual typing. , 2019, , .		11
8	A space-efficient call-by-value virtual machine for gradual set-theoretic types. , 2019, , .		0
9	Gradually typed symbolic expressions. , 2018, , .		1
10	Automatically generating the dynamic semantics of gradually typed languages. , 2017, , .		17
11	Big types in little runtime: open-world soundness and collaborative blame for gradual type systems. , 2017, , .		22
12	Theorems for free for free: parametricity, with and without types. , 2017, 1, 1-28.		28
13	Gradually typed symbolic expressions. , 2017, , .		3
14	Sound gradual typing: only mostly dead. , 2017, 1, 1-24.		17
15	Big types in little runtime: open-world soundness and collaborative blame for gradual type systems. ACM SIGPLAN Notices, 2017, 52, 762-774.	0.2	6
16	Automatically generating the dynamic semantics of gradually typed languages. ACM SIGPLAN Notices, 2017, 52, 789-803.	0.2	4
17	Fractional Permissions for Race-Free Mutable References in a Dataflow Intermediate Language. , 2016, , .		0
18	The gradualizer: a methodology and algorithm for generating gradual type systems. , 2016, , .		36

#	Article	IF	Citations
19	The Recursive Union of Some Gradual Types. Lecture Notes in Computer Science, 2016, , 388-410.	1.3	9
20	The gradualizer: a methodology and algorithm for generating gradual type systems. ACM SIGPLAN Notices, 2016, 51, 443-455.	0.2	4
21	Reliable Generation of High-Performance Matrix Algebra. ACM Transactions on Mathematical Software, 2015, 41, 1-27.	2.9	8
22	Blame and coercion: together again for the first time. , 2015, , .		26
23	Monotonic References for Efficient Gradual Typing. Lecture Notes in Computer Science, 2015, , 432-456.	1.3	35
24	Design and evaluation of gradual typing for python. ACM SIGPLAN Notices, 2015, 50, 45-56.	0.2	26
25	Pycket: a tracing JIT for a functional language. , 2015, , .		29
26	Blame and coercion: together again for the first time. ACM SIGPLAN Notices, 2015, 50, 425-435.	0.2	9
27	Region-based memory management for GPU programming languages. , 2014, , .		5
28	Design and evaluation of gradual typing for python. , 2014, , .		76
29	Compile-time reflection and metaprogramming for Java. , 2014, , .		5
30	Modular type-safety proofs in Agda. , 2013, , .		11
31	Well-Typed Islands Parse Faster. Lecture Notes in Computer Science, 2013, , 69-84.	1.3	4
32	Visualizing transactional memory. , 2012, , .		7
33	19th international workshop on foundations of object-oriented languages (FOOL'12). , 2012, , .		1
34	Interpretations of the gradually-typed lambda calculus. , 2012, , .		13
35	Pattern-based traits. , 2012, , .		4
36	The C++0x â€œConceptsâ€• Effort. Lecture Notes in Computer Science, 2012, , 175-216.	1.3	6

#	Article	IF	Citations
37	Blame for all. ACM SIGPLAN Notices, 2011, 46, 201-214.	0.2	15
38	A language for generic programming in the large. Science of Computer Programming, 2011, 76, 423-465.	1.9	16
39	Blame for all. , 2011, , .		71
40	2011 international workshop on foundations of object-oriented languages (fool'11). , 2011, , .		0
41	Parallel memory prediction for fused linear algebra kernels. Performance Evaluation Review, 2011, 38, 43-49.	0.6	4
42	Incremental type-checking for type-reflective metaprograms. ACM SIGPLAN Notices, 2011, 46, 167-176.	0.2	1
43	Threesomes, with and without blame. ACM SIGPLAN Notices, 2010, 45, 365-376.	0.2	12
44	Understanding memory effects in the automated generation of optimized matrix algebra kernels. Procedia Computer Science, 2010, 1, 1873-1881.	2.0	1
45	2010 international workshop on foundations of object-oriented languages (FOOL'10). , 2010, , .		0
46	Incremental type-checking for type-reflective metaprograms. , 2010, , .		9
47	Threesomes, with and without blame. , 2010, , .		64
48	General purpose languages should be metalanguages. , 2010, , .		7
49	An efficient software transactional memory using commit-time invalidation. , 2010, , .		30
50	An efficient lock-aware transactional memory implementation. , 2009, , .		2
51	Automating the generation of composed linear algebra kernels. , 2009, , .		37
52	In Pursuit of Real Answers. , 2009, , .		0
53	Exploring the Design Space of Higher-Order Casts. Lecture Notes in Computer Science, 2009, , 17-31.	1.3	57
54	Generating Empirically Optimized Composed Matrix Kernels from MATLAB Prototypes. Lecture Notes in Computer Science, 2009, , 248-258.	1.3	5

#	Article	IF	Citations
55	Threesomes, with and without blame. , 2009, , .		8
56	Build to order linear algebra kernels. Parallel and Distributed Processing Symposium (IPDPS), Proceedings of the International Conference on, 2008, , .	1.0	20
57	Gradual typing with unification-based inference. , 2008, , .		55
58	An extended comparative study of language support for generic programming. Journal of Functional Programming, 2007, 17, 145-205.	0.8	45
59	Gradual Typing for Objects. Lecture Notes in Computer Science, 2007, , 2-27.	1.3	176
60	Improving the lazy Krivine machine. Higher-Order and Symbolic Computation, 2007, 20, 271-293.	0.3	19
61	Concoqtion. , 2007, , .		42
62	Algorithm specialization in generic programming. ACM SIGPLAN Notices, 2006, 41, 272-282.	0.2	8
63	LCSD. , 2006, , .		1
64	Algorithm specialization in generic programming. , 2006, , .		12
65	Essential language support for generic programming. ACM SIGPLAN Notices, 2005, 40, 73-84.	0.2	6
66	Essential language support for generic programming. , 2005, , .		34
67	Language Requirements for Large-Scale Generic Libraries. Lecture Notes in Computer Science, 2005, , 405-421.	1.3	11
68	Modular generics. , 2004, , .		0
69	A comparative study of language support for generic programming. , 2003, , .		77
70	A Modern Framework for Portable High-Performance Numerical Linear Algebra. Lecture Notes in Computational Science and Engineering, 2000, , 1-55.	0.3	16
71	The generic graph component library. , 1999, , .		23
72	The generic graph component library. ACM SIGPLAN Notices, 1999, 34, 399-414.	0.2	13