

# Peter Sewell

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/2084759/publications.pdf>

Version: 2024-02-01

94  
papers

3,711  
citations

304743

22  
h-index

233421

45  
g-index

94  
all docs

94  
docs citations

94  
times ranked

883  
citing authors

#	ARTICLE	IF	CITATIONS
1	x86-TSO. Communications of the ACM, 2010, 53, 89-97.	4.5	327
2	A Better x86 Memory Model: x86-TSO. Lecture Notes in Computer Science, 2009, , 391-407.	1.3	275
3	Mathematizing C++ concurrency. , 2011, , .		217
4	Mechanized Metatheory for the Masses: The PoplMark Challenge. Lecture Notes in Computer Science, 2005, , 50-65.	1.3	200
5	Understanding POWER multiprocessors. , 2011, , .		159
6	CompCertTSO. Journal of the ACM, 2013, 60, 1-50.	2.2	108
7	Fences in Weak Memory Models. Lecture Notes in Computer Science, 2010, , 258-272.	1.3	106
8	Ott: Effective tool support for the working semanticist. Journal of Functional Programming, 2010, 20, 71-122.	0.8	99
9	Simplifying ARM concurrency: multicopy-atomic axiomatic and operational models for ARMv8. , 2018, 2, 1-29.		87
10	Modelling the ARMv8 architecture, operationally: concurrency and ISA. , 2016, , .		85
11	An Axiomatic Memory Model for POWER Multiprocessors. Lecture Notes in Computer Science, 2012, , 495-512.	1.3	82
12	The semantics of x86-CC multiprocessor machine code. , 2009, , .		80
13	Clarifying and compiling C/C++ concurrency. , 2012, , .		77
14	Mathematizing C++ concurrency. ACM SIGPLAN Notices, 2011, 46, 55-66.	0.2	75
15	The Problem of Programming Language Concurrency Semantics. Lecture Notes in Computer Science, 2015, , 283-307.	1.3	70
16	Synchronising C/C++ and POWER. , 2012, , .		67
17	<i>Mutatis Mutandis</i>. ACM Transactions on Programming Languages and Systems, 2007, 29, 22.	2.1	65
18	The semantics of power and ARM multiprocessor machine code. , 2009, , .		65

#	ARTICLE	IF	CITATIONS
19	ISA semantics for ARMv8-a, RISC-v, and CHERI-MIPS. , 2019, 3, 1-31.		61
20	Litmus: Running Tests against Hardware. Lecture Notes in Computer Science, 2011, , 41-44.	1.3	55
21	From rewrite rules to bisimulation congruences. Theoretical Computer Science, 2002, 274, 183-230.	0.9	53
22	Lem. , 2014, , .		53
23	Acute. ACM SIGPLAN Notices, 2005, 40, 15-26.	0.2	52
24	Into the depths of C: elaborating the de facto standards. , 2016, , .		51
25	A concurrency semantics for relaxed atomics that permits optimisation and avoids thin-air executions. , 2016, , .		49
26	Understanding POWER multiprocessors. ACM SIGPLAN Notices, 2011, 46, 175-186.	0.2	45
27	Global/local subtyping and capability inference for a distributed $\lambda$ -calculus. Lecture Notes in Computer Science, 1998, , 695-706.	1.3	42
28	Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and sockets. , 2005, , .		42
29	The semantics of x86-CC multiprocessor machine code. ACM SIGPLAN Notices, 2009, 44, 379-391.	0.2	42
30	Ott. , 2007, , .		40
31	Relaxed-memory concurrency and verified compilation. , 2011, , .		40
32	The java module system. , 2007, , .		38
33	Acute. , 2005, , .		37
34	SibylFS. , 2015, , .		35
35	Nomadic pict. , 2001, , .		31
36	Location-Independent Communication for Mobile Agents: A Two-Level Architecture. Lecture Notes in Computer Science, 1999, , 1-31.	1.3	30

#	ARTICLE	IF	CITATIONS
37	Modules, abstract types, and distributed versioning. , 2001, , .		29
38	Acute: High-level programming language design for distributed computation. Journal of Functional Programming, 2007, 17, 547-612.	0.8	29
39	Nonaxiomatisability of equivalences over finite state processes. Annals of Pure and Applied Logic, 1997, 90, 163-191.	0.5	28
40	An integrated concurrency and core-ISA architectural envelope definition, and test oracle, for IBM POWER multiprocessors. , 2015, , .		27
41	Ott. ACM SIGPLAN Notices, 2007, 42, 1-12.	0.2	26
42	Secure composition of untrusted code: box i€, wrappers, and causality types. Journal of Computer Security, 2003, 11, 135-187.	0.8	25
43	Fences in weak memory models (extended version). Formal Methods in System Design, 2012, 40, 170-205.	0.8	24
44	Exploring C semantics and pointer provenance. , 2019, 3, 1-32.		24
45	Engineering with logic. , 2006, , .		24
46	An operational semantics for C/C++11 concurrency. , 2016, , .		24
47	Dynamic rebinding for marshalling and update, with destruct-time ?. , 2003, , .		21
48	Models for name-passing processes: interleaving and causal. Information and Computation, 2004, 190, 136-178.	0.7	21
49	Relaxed-memory concurrency and verified compilation. ACM SIGPLAN Notices, 2011, 46, 43-54.	0.2	20
50	Nomadic pict. ACM Transactions on Programming Languages and Systems, 2010, 32, 1-63.	2.1	19
51	Modelling the ARMv8 architecture, operationally: concurrency and ISA. ACM SIGPLAN Notices, 2016, 51, 608-621.	0.2	19
52	Mixed-size concurrency: ARM, POWER, C/C++11, and SC. , 2017, , .		19
53	From rewrite to bisimulation congruences. Lecture Notes in Computer Science, 1998, , 269-284.	1.3	18
54	Global abstraction-safe marshalling with hash types. , 2003, , .		18

#	ARTICLE	IF	CITATIONS
55	Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and sockets. <i>Computer Communication Review</i> , 2005, 35, 265-276.	1.8	17
56	Type-safe distributed programming for OCaml. , 2006, , .		16
57	Lem. <i>ACM SIGPLAN Notices</i> , 2014, 49, 175-188.	0.2	16
58	Clarifying and compiling C/C++ concurrency. <i>ACM SIGPLAN Notices</i> , 2012, 47, 509-520.	0.2	14
59	The missing link: explaining ELF static linking, semantically. , 2016, , .		13
60	Lem: A Lightweight Tool for Heavyweight Semantics. <i>Lecture Notes in Computer Science</i> , 2011, , 363-369.	1.3	12
61	Into the depths of C: elaborating the de facto standards. <i>ACM SIGPLAN Notices</i> , 2016, 51, 1-15.	0.2	12
62	ARMv8-A System Semantics: Instruction Fetch in Relaxed Architectures. <i>Lecture Notes in Computer Science</i> , 2020, , 626-655.	1.3	11
63	A Rigorous Approach to Networking: TCP, from Implementation to Protocol to Service. , 2008, , 294-309.		11
64	An operational semantics for C/C++11 concurrency. <i>ACM SIGPLAN Notices</i> , 2016, 51, 111-128.	0.2	10
65	Synchronising C/C++ and POWER. <i>ACM SIGPLAN Notices</i> , 2012, 47, 311-322.	0.2	8
66	The UDP Calculus: Rigorous Semantics for Real Networking. <i>Lecture Notes in Computer Science</i> , 2001, , 535-559.	1.3	8
67	On implementations and semantics of a concurrent programming language. <i>Lecture Notes in Computer Science</i> , 1997, , 391-405.	1.3	8
68	A concurrency semantics for relaxed atomics that permits optimisation and avoids thin-air executions. <i>ACM SIGPLAN Notices</i> , 2016, 51, 622-633.	0.2	8
69	Islaris: verification of machine code against authoritative ISA semantics. , 2022, , .		8
70	Nomadic pict. <i>ACM SIGPLAN Notices</i> , 2001, 36, 116-127.	0.2	7
71	Rigour is good for you and feasible. , 2002, , .		7
72	Passive-attack analysis for connection-based anonymity systems. <i>International Journal of Information Security</i> , 2005, 4, 172-180.	3.4	7

#	ARTICLE	IF	CITATIONS
73	Engineering with Logic. Journal of the ACM, 2019, 66, 1-77.	2.2	7
74	Isla: Integrating Full-Scale ISA Semantics and Axiomatic Concurrency Models. Lecture Notes in Computer Science, 2021, , 303-316.	1.3	7
75	Cerberus-BMC: A Principled Reference Semantics and Exploration Tool for Concurrent and Sequential C. Lecture Notes in Computer Science, 2019, , 387-397.	1.3	7
76	Engineering with logic. ACM SIGPLAN Notices, 2006, 41, 55-66.	0.2	6
77	The java module system. ACM SIGPLAN Notices, 2007, 42, 499-514.	0.2	6
78	Relaxed virtual memory in Armv8-A. Lecture Notes in Computer Science, 2022, , 143-173.	1.3	6
79	Mixed-size concurrency: ARM, POWER, C/C++11, and SC. ACM SIGPLAN Notices, 2017, 52, 429-442.	0.2	5
80	Timing UDP: Mechanized Semantics for Sockets, Threads, and Failures. Lecture Notes in Computer Science, 2002, , 278-294.	1.3	4
81	Dynamic rebinding for marshalling and update, via redex-time and destruct-time reduction. Journal of Functional Programming, 2008, 18, .	0.8	3
82	The missing link: explaining ELF static linking, semantically. ACM SIGPLAN Notices, 2016, 51, 607-623.	0.2	3
83	Rigorous Protocol Design in Practice: An Optical Packet-Switch MAC in HOL. , 2006, , .		2
84	It Is Time to Mechanize Programming Language Metatheory. Lecture Notes in Computer Science, 2008, , 26-30.	1.3	2
85	Global abstraction-safe marshalling with hash types. ACM SIGPLAN Notices, 2003, 38, 87-98.	0.2	1
86	Dynamic rebinding for marshalling and update, with destruct-time ?. ACM SIGPLAN Notices, 2003, 38, 99-110.	0.2	1
87	Process Calculi: The End of the Beginning?. Electronic Notes in Theoretical Computer Science, 2006, 162, 317-321.	0.9	1
88	POPL 2014 program chair's report. ACM SIGPLAN Notices, 2014, 49, 10-26.	0.2	1
89	VIP: verifying real-world C idioms with integer-pointer casts. , 2022, 6, 1-32.		1
90	Modules, abstract types, and distributed versioning. ACM SIGPLAN Notices, 2001, 36, 236-247.	0.2	0

#	ARTICLE	IF	CITATIONS
91	Tales from the jungle. , 2012, , .		0
92	Principles of POPL. ACM SIGPLAN Notices, 2013, 48, 12-16.	0.2	0
93	Memory, an elusive abstraction. ACM SIGPLAN Notices, 2010, 45, 51-52.	0.2	0
94	False Concurrency and Strange-but-True Machines. Lecture Notes in Computer Science, 2012, , 37-38.	1.3	0