

Viktor Vafeiadis

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/1646374/publications.pdf>

Version: 2024-02-01

90
papers

2,870
citations

331670

21
h-index

345221

36
g-index

92
all docs

92
docs citations

92
times ranked

584
citing authors

#	ARTICLE	IF	CITATIONS
1	Truly stateless, optimal dynamic partial order reduction. , 2022, 6, 1-28.		18
2	Extending Intel-x86 consistency and persistency: formalising the semantics of Intel-x86 memory types and non-temporal stores. , 2022, 6, 1-31.		3
3	The Decidability of Verification under PS 2.0. Lecture Notes in Computer Science, 2021, , 1-29.	1.3	7
4	GenMC: A Model Checker for Weak Memory Models. Lecture Notes in Computer Science, 2021, , 427-440.	1.3	19
5	VSync: push-button verification and optimization for synchronization primitives on weak memory models. , 2021, , .		20
6	PerSeVerE: persistency semantics for verification under ext4. , 2021, 5, 1-29.		9
7	Making weak memory models fair. , 2021, 5, 1-27.		10
8	Verifying and Optimizing the HMCS Lock for Arm Servers. Lecture Notes in Computer Science, 2021, , 240-260.	1.3	2
9	BAM: Efficient Model Checking for Barriers. Lecture Notes in Computer Science, 2021, , 223-239.	1.3	1
10	Persistency semantics of the Intel-x86 architecture. , 2020, 4, 1-31.		31
11	HMC. , 2020, , .		19
12	Promising 2.0: global optimizations in relaxed memory concurrency. , 2020, , .		29
13	Persistent Owicki-Gries reasoning: a program logic for reasoning about persistent programs on Intel-x86. , 2020, 4, 1-28.		8
14	Model checking for weakly consistent libraries. , 2019, , .		49
15	Bridging the gap between programming languages and hardware weak memory models. , 2019, 3, 1-31.		37
16	Grounding thin-air reads with event structures. , 2019, 3, 1-28.		35
17	On library correctness under weak memory consistency: specifying and verifying concurrent libraries under declarative consistency models. , 2019, 3, 1-31.		20
18	On the Semantics of Snapshot Isolation. Lecture Notes in Computer Science, 2019, , 1-23.	1.3	4

#	ARTICLE	IF	CITATIONS
19	Weak persistency semantics from the ground up: formalising the persistency semantics of ARMv8 and transactional models. , 2019, 3, 1-27.		31
20	Effective lock handling in stateless model checking. , 2019, 3, 1-26.		17
21	GPS\$\$\$++: Reasoning About Fences and Relaxed Atomics. International Journal of Parallel Programming, 2018, 46, 1157-1183.	1.5	6
22	Effective stateless model checking for C/C++ concurrency. , 2018, 2, 1-32.		69
23	Persistence semantics for weak memory: integrating epoch persistency with the TSO memory model. , 2018, 2, 1-27.		23
24	A Separation Logic for a Promising Semantics. Lecture Notes in Computer Science, 2018, , 357-384.	1.3	24
25	On Parallel Snapshot Isolation and Release/Acquire Consistency. Lecture Notes in Computer Science, 2018, , 940-967.	1.3	7
26	A promising semantics for relaxed-memory concurrency. , 2017, , .		93
27	Formalizing the concurrency semantics of an LLVM fragment. , 2017, , .		11
28	Program Verification Under Weak Memory Consistency Using Separation Logic. Lecture Notes in Computer Science, 2017, , 30-46.	1.3	5
29	Tackling Real-Life Relaxed Concurrency with FSL++. Lecture Notes in Computer Science, 2017, , 448-475.	1.3	31
30	Repairing sequential consistency in C/C++11. , 2017, , .		90
31	A promising semantics for relaxed-memory concurrency. ACM SIGPLAN Notices, 2017, 52, 175-189.	0.2	24
32	Repairing sequential consistency in C/C++11. ACM SIGPLAN Notices, 2017, 52, 618-632.	0.2	22
33	Reasoning about Fences and Relaxed Atomics. , 2016, , .		9
34	Validating optimizations of concurrent C/C++ programs. , 2016, , .		19
35	Explaining Relaxed Memory Models with Program Transformations. Lecture Notes in Computer Science, 2016, , 479-495.	1.3	20
36	A Program Logic for C11 Memory Fences. Lecture Notes in Computer Science, 2016, , 413-430.	1.3	33

#	ARTICLE	IF	CITATIONS
37	Lightweight verification of separate compilation. , 2016, , .		33
38	Taming release-acquire consistency. , 2016, , .		59
39	Lightweight verification of separate compilation. ACM SIGPLAN Notices, 2016, 51, 178-190.	0.2	9
40	Taming release-acquire consistency. ACM SIGPLAN Notices, 2016, 51, 649-662.	0.2	16
41	Mtac: A monad for typed tactic programming in Coq. Journal of Functional Programming, 2015, 25, .	0.8	15
42	Common Compiler Optimisations are Invalid in the C11 Memory Model and what we can do about it. ACM SIGPLAN Notices, 2015, 50, 209-220.	0.2	11
43	Common Compiler Optimisations are Invalid in the C11 Memory Model and what we can do about it. , 2015, , .		88
44	A formal C memory model supporting integer-pointer casts. , 2015, , .		32
45	Verifying read-copy-update in a logic for weak memory. , 2015, , .		29
46	Proving Lock-Freedom Easily and Automatically. , 2015, , .		5
47	Formal Reasoning about the C11 Weak Memory Model. , 2015, , .		2
48	Owicki-Gries Reasoning for Weak Memory Models. Lecture Notes in Computer Science, 2015, , 311-323.	1.3	62
49	Modular Verification of Concurrency-Aware Linearizability. Lecture Notes in Computer Science, 2015, , 371-387.	1.3	18
50	Pilsner: a compositionally verified compiler for a higher-order imperative language. , 2015, , .		54
51	Pilsner: a compositionally verified compiler for a higher-order imperative language. ACM SIGPLAN Notices, 2015, 50, 166-178.	0.2	8
52	Verifying read-copy-update in a logic for weak memory. ACM SIGPLAN Notices, 2015, 50, 110-120.	0.2	10
53	A formal C memory model supporting integer-pointer casts. ACM SIGPLAN Notices, 2015, 50, 326-335.	0.2	9
54	GPS. , 2014, , .		67

#	ARTICLE	IF	CITATIONS
55	GPS. ACM SIGPLAN Notices, 2014, 49, 691-707.	0.2	24
56	The power of parameterization in coinductive proof. , 2013, , .		43
57	CompCertTSO. Journal of the ACM, 2013, 60, 1-50.	2.2	108
58	Mtac. , 2013, , .		24
59	Relaxed separation logic. , 2013, , .		79
60	A Programming Language Approach to Fault Tolerance for Fork-Join Parallelism. , 2013, , .		3
61	Relaxed separation logic. ACM SIGPLAN Notices, 2013, 48, 867-884.	0.2	26
62	Mtac. ACM SIGPLAN Notices, 2013, 48, 87-100.	0.2	4
63	Adjustable References. Lecture Notes in Computer Science, 2013, , 328-337.	1.3	4
64	Aspect-Oriented Linearizability Proofs. Lecture Notes in Computer Science, 2013, , 242-256.	1.3	40
65	The power of parameterization in coinductive proof. ACM SIGPLAN Notices, 2013, 48, 193-206.	0.2	15
66	The marriage of bisimulations and Kripke logical relations. , 2012, , .		49
67	The marriage of bisimulations and Kripke logical relations. ACM SIGPLAN Notices, 2012, 47, 59-72.	0.2	5
68	Concurrent Separation Logic and Operational Semantics. Electronic Notes in Theoretical Computer Science, 2011, 276, 335-351.	0.9	54
69	Relaxed-memory concurrency and verified compilation. , 2011, , .		40
70	Separation Logic in the Presence of Garbage Collection. , 2011, , .		4
71	Relaxed-memory concurrency and verified compilation. ACM SIGPLAN Notices, 2011, 46, 43-54.	0.2	20
72	Verifying Fence Elimination Optimisations. Lecture Notes in Computer Science, 2011, , 146-162.	1.3	17

#	ARTICLE	IF	CITATIONS
73	Structuring the verification of heap-manipulating programs. , 2010, , .		46
74	Structuring the verification of heap-manipulating programs. ACM SIGPLAN Notices, 2010, 45, 261-274.	0.2	19
75	RGSep Action Inference. Lecture Notes in Computer Science, 2010, , 345-361.	1.3	25
76	Concurrent Abstract Predicates. Lecture Notes in Computer Science, 2010, , 504-528.	1.3	111
77	Automatically Proving Linearizability. Lecture Notes in Computer Science, 2010, , 450-464.	1.3	97
78	Proving that non-blocking algorithms don't block. , 2009, , .		30
79	Finding heap-bounds for hardware synthesis. , 2009, , .		18
80	Deny-Guarantee Reasoning. Lecture Notes in Computer Science, 2009, , 363-377.	1.3	82
81	Bi-abductive Resource Invariant Synthesis. Lecture Notes in Computer Science, 2009, , 259-274.	1.3	29
82	Proving that non-blocking algorithms don't block. ACM SIGPLAN Notices, 2009, 44, 16-28.	0.2	20
83	Shape-Value Abstraction for Verifying Linearizability. Lecture Notes in Computer Science, 2008, , 335-348.	1.3	65
84	Acute: High-level programming language design for distributed computation. Journal of Functional Programming, 2007, 17, 547-612.	0.8	29
85	Modular Safety Checking for Fine-Grained Concurrency. Lecture Notes in Computer Science, 2007, , 233-248.	1.3	42
86	A Marriage of Rely/Guarantee and Separation Logic. Lecture Notes in Computer Science, 2007, , 256-271.	1.3	138
87	Proving correctness of highly-concurrent linearisable objects. , 2006, , .		70
88	Acute. ACM SIGPLAN Notices, 2005, 40, 15-26.	0.2	52
89	Acute. , 2005, , .		37
90	Aspect-oriented linearizability proofs. Logical Methods in Computer Science, 0, Volume 11, Issue 1, .	0.4	17