Georg Weissenbacher

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/2463216/publications.pdf

Version: 2024-02-01

46	849	687363 13	552781 26
papers	citations	h-index	g-index
48	48	48	585
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	A Survey of Automated Techniques for Formal Software Verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27, 1165-1178.	2.7	237
2	Interpolant Strength. Lecture Notes in Computer Science, 2010, , 129-145.	1.3	70
3	Boolean Satisfiability Solvers and Their Applications in Model Checking. Proceedings of the IEEE, 2015, 103, 2021-2035.	21.3	67
4	Model checking concurrent linux device drivers. , 2007, , .		64
5	Mutation-Based Test Case Generation for Simulink Models. Lecture Notes in Computer Science, 2010, , 208-227.	1.3	43
6	Counterexample to Induction-Guided Abstraction-Refinement (CTIGAR). Lecture Notes in Computer Science, 2014, , 831-848.	1.3	39
7	Interpolation-Based Software Verification with Wolverine. Lecture Notes in Computer Science, 2011, , 573-578.	1.3	31
8	Randomized testing of distributed systems with probabilistic guarantees. , 2018, 2, 1-28.		23
9	Counterexamples with Loops for Predicate Abstraction. Lecture Notes in Computer Science, 2006, , 152-165.	1.3	21
10	SAT-based techniques for determining backbones for post-silicon fault localisation. , 2011, , .		18
11	Under-Approximating Loops in C Programs for Fast Counterexample Detection. Lecture Notes in Computer Science, 2013, , 381-396.	1.3	18
12	Model-based, Mutation-driven Test-case Generation Via Heuristic-guided Branching Search. Transactions on Embedded Computing Systems, 2019, 18, 1-28.	2.9	15
13	Verification and falsification of programs with loops using predicate abstraction. Formal Aspects of Computing, 2010, 22, 105-128.	1.8	14
14	Under-approximating loops in C programs for fast counterexample detection. Formal Methods in System Design, 2015, 47, 75-92.	0.8	14
15	Abstraction and Mining of Traces to Explain Concurrency Bugs. Lecture Notes in Computer Science, 2014, , 162-177.	1.3	14
16	Advanced SAT Techniques for Abstract Argumentation. Lecture Notes in Computer Science, 2013, , 138-154.	1.3	13
17	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		12
18	Incremental bounded software model checking. , 2014, , .		12

#	Article	IF	Citations
19	Silicon fault diagnosis using sequence interpolation with backbones. , 2014, , .		9
20	Model-based, mutation-driven test case generation via heuristic-guided branching search. , 2017, , .		8
21	SAT-Based Summarization for Boolean Programs. , 2007, , 131-148.		8
22	An Interpolating Decision Procedure for Transitive Relations with Uninterpreted Functions. Lecture Notes in Computer Science, 2011, , 150-168.	1.3	8
23	Interpolant Strength Revisited. Lecture Notes in Computer Science, 2012, , 312-326.	1.3	8
24	Proving Safety with Trace Automata and Bounded Model Checking. Lecture Notes in Computer Science, 2015, , 325-341.	1.3	7
25	Dynamic Reductions for Model Checking Concurrent Software. Lecture Notes in Computer Science, 2017, , 246-265.	1.3	6
26	A Separation Logic with Data: Small Models and Automation. Lecture Notes in Computer Science, 2018, , 455-471.	1.3	6
27	Rely-Guarantee Reasoning for Automated Bound Analysis of Lock-Free Algorithms. , 2018, , .		5
28	Model-Based Diagnosis with Multiple Observations. , 2019, , .		5
29	Abstraction and mining of traces to explain concurrency bugs. Formal Methods in System Design, 2016, 49, 1-32.	0.8	4
30	Error Invariants for Concurrent Traces. Lecture Notes in Computer Science, 2016, , 370-387.	1.3	3
31	Labelled Interpolation Systems for Hyper-Resolution, Clausal, and Local Proofs. Journal of Automated Reasoning, 2016, 57, 3-36.	1.4	3
32	Mutation testing with hyperproperties. Software and Systems Modeling, 2021, 20, 405-427.	2.7	3
33	Mutation Testing with Hyperproperties. Lecture Notes in Computer Science, 2019, , 203-221.	1.3	3
34	Reduction of Resolution Refutations and Interpolants via Subsumption. Lecture Notes in Computer Science, 2014, , 188-203.	1.3	3
35	Wolverine: Battling Bugs with Interpolants. Lecture Notes in Computer Science, 2012, , 556-558.	1.3	3
36	Extracting Safe Thread Schedules from Incomplete Model Checking Results. Lecture Notes in Computer Science, 2019, , 153-171.	1.3	3

#	Article	IF	Citations
37	Extracting safe thread schedules from incomplete model checking results. International Journal on Software Tools for Technology Transfer, 2020, 22, 565-581.	1.9	2
38	Rely-guarantee bound analysis of parameterized concurrent shared-memory programs. Formal Methods in System Design, 2021, 57, 270-302.	0.8	2
39	Multi-linear Strategy Extraction for QBF Expansion Proofs via Local Soundness. Lecture Notes in Computer Science, 2020, , 429-446.	1.3	2
40	Preface of the Special Issue in Memoriam Helmut Veith. Formal Methods in System Design, 2017, 51, 267-269.	0.8	1
41	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		1
42	Preface of the special issue on the Conference on Formal Methods in Computer-Aided Design 2017. Formal Methods in System Design, 2021, 57, 303-304.	0.8	0
43	Preface of the special issue on the conference on computer-aided verification 2018. Formal Methods in System Design, 2021, 57, 1.	0.8	0
44	Parallel Assertions for Architectures with Weak Memory Models. Lecture Notes in Computer Science, 2012, , 254-268.	1.3	0
45	Language Inclusion for Finite Prime Event Structures. Lecture Notes in Computer Science, 2020, , 314-336.	1.3	0
46	RAT Elimination. , 0, , .		0