Sarfraz Khurshid

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/11715675/publications.pdf

Version: 2024-02-01

87 papers	3,072 citations	623734 14 h-index	330143 37 g-index
87 all docs	87 docs citations	87 times ranked	1169 citing authors

#	Article	IF	Citations
1	Programming and training rate-independent chemical reaction networks. Proceedings of the National Academy of Sciences of the United States of America, 2022, 119, .	7.1	3
2	A study of learning likely data structure properties using machine learning models. International Journal on Software Tools for Technology Transfer, 2020, 22, 601-615.	1.9	1
3	A study of the learnability of relational properties: model counting meets machine learning (MCML). , 2020, , .		4
4	A Study of Symmetry Breaking Predicates and Model Counting. Lecture Notes in Computer Science, 2020, , 115-134.	1.3	4
5	The Java Pathfinder Workshop 2019. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2020, 45, 20-22.	0.7	0
6	AlloyMC: Alloy meets model counting. , 2020, , .		1
7	A synergistic approach to improving symbolic execution using test ranges. Innovations in Systems and Software Engineering, 2019, 15, 325-342.	2.1	4
8	Extension-Aware Automated Testing Based on Imperative Predicates. , 2019, , .		2
9	EdSketch: execution-driven sketching for Java. International Journal on Software Tools for Technology Transfer, 2019, 21, 249-265.	1.9	7
10	Learning Guided Enumerative Synthesis for Superoptimization. Lecture Notes in Computer Science, 2019, , 172-192.	1.3	2
11	A Study of Learning Data Structure Invariants Using Off-the-shelf Tools. Lecture Notes in Computer Science, 2019, , 226-243.	1.3	6
12	Incremental Analysis of Evolving Alloy Models. Lecture Notes in Computer Science, 2019, , 174-191.	1.3	5
13	Quantifying the Exploration of the Korat Solver for Imperative Constraints. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2019, 44, 15-15.	0.7	1
14	Using Test Ranges to Improve Symbolic Execution. Lecture Notes in Computer Science, 2018, , 416-434.	1.3	7
15	ASketch: a sketching framework for Alloy. , 2018, , .		10
16	Korat-API. , 2018, , .		7
17	Non-Semantics-Preserving Transformations for Higher-Coverage Test Generation Using Symbolic Execution. , 2017, , .		6
18	Boosting spectrum-based fault localization using PageRank. , 2017, , .		80

#	Article	IF	Citations
19	Optimizing parallel Korat using invalid ranges. , 2017, , .		7
20	A Sketching-Based Approach for Debugging Using Test Cases. Lecture Notes in Computer Science, 2016, , 463-478.	1.3	5
21	Compositional Symbolic Execution with Memoized Replay. , 2015, , .		19
22	Studying the influence of standard compiler optimizations on symbolic execution. , 2015, , .		21
23	Understanding the triaging and fixing processes of long lived bugs. Information and Software Technology, 2015, 65, 114-128.	4.4	34
24	An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes. , 2015, , .		61
25	Bounded exhaustive test input generation from hybrid invariants. , 2014, , .		13
26	Property differencing for incremental checking. , 2014, , .		25
27	Directed Incremental Symbolic Execution. ACM Transactions on Software Engineering and Methodology, 2014, 24, 1-42.	6.0	76
28	Bounded exhaustive test input generation from hybrid invariants. ACM SIGPLAN Notices, 2014, 49, 655-674.	0.2	3
29	Scaling symbolic execution using staged analysis. Innovations in Systems and Software Engineering, 2013, 9, 119-131.	2.1	4
30	Faster mutation testing inspired by test prioritization and reduction. , 2013, , .		72
31	Memoise: A tool for memoized symbolic execution. , 2013, , .		7
32	<scp>FaultTracer</scp>: a spectrumâ€based approach to localizing failureâ€inducing program edits. Journal of Software: Evolution and Process, 2013, 25, 1357-1383.	1.6	14
33	Ranger: Parallel analysis of alloy models by range partitioning. , 2013, , .		11
34	Improving bug localization using structured information retrieval. , 2013, , .		248
35	Injecting mechanical faults to localize developer faults for evolving software. ACM SIGPLAN Notices, 2013, 48, 765-784.	0.2	33
36	Staged symbolic execution. , 2012, , .		11

#	Article	IF	Citations
37	Memoized symbolic execution. , 2012, , .		74
38	Scaling symbolic execution using ranged analysis. , 2012, , .		30
39	Specification-Based Test Repair Using a Lightweight Formal Method. Lecture Notes in Computer Science, 2012, , 455-470.	1.3	12
40	Scaling symbolic execution using ranged analysis. ACM SIGPLAN Notices, 2012, 47, 523-536.	0.2	5
41	Dynamic Shape Analysis Using Spectral Graph Properties. , 2012, , .		1
42	Shared Execution for Efficiently Testing Product Lines. , 2012, , .		36
43	Regression mutation testing. , 2012, , .		61
44	Annotations for Alloy: Automated Incremental Analysis Using Domain Specific Solvers. Lecture Notes in Computer Science, 2012, , 414-429.	1.3	12
45	Efficiently Running Test Suites Using Abstract Undo Operations. , 2011, , .		8
46	An Empirical Study of JUnit Test-Suite Reduction. , 2011, , .		55
47	Systematic Testing of Database Engines Using a Relational Constraint Solver. , 2011, , .		13
48	Directed incremental symbolic execution. ACM SIGPLAN Notices, 2011, 46, 504-515.	0.2	49
49	A case for alloy annotations for efficient incremental analysis via domain specific solvers. , 2011, , .		2
50	Constraint-based program debugging using data structure repair. , 2011, , .		19
51	Localizing failure-inducing program edits based on spectrum information. , 2011, , .		87
52	Directed incremental symbolic execution. , 2011, , .		122
53	Specification-Based Program Repair Using SAT. Lecture Notes in Computer Science, 2011, , 173-188.	1.3	54
54	Symbolic Execution of Alloy Models. Lecture Notes in Computer Science, 2011, , 340-355.	1.3	1

#	Article	IF	Citations
55	ParSym: Parallel symbolic execution. , 2010, , .		27
56	Test generation through programming in UDITA. , 2010, , .		115
57	Incremental Test Generation for Software Product Lines. IEEE Transactions on Software Engineering, 2010, 36, 309-322.	5.6	79
58	Optimizing Incremental Scope-Bounded Checking with Data-Flow Analysis. , 2010, , .		7
59	Contract-Based Data Structure Repair Using Alloy. Lecture Notes in Computer Science, 2010, , 577-598.	1.3	14
60	A Case for Automated Debugging Using Data Structure Repair. , 2009, , .		28
61	PKorat: Parallel Generation of Structurally Complex Test Inputs. , 2009, , .		17
62	An Incremental Approach to Scope-Bounded Checking Using a Lightweight Formal Method. Lecture Notes in Computer Science, 2009, , 757-772.	1.3	7
63	An Empirical Study of Structural Constraint Solving Techniques. Lecture Notes in Computer Science, 2009, , 88-106.	1.3	13
64	Query-Aware Test Generation Using a Relational Constraint Solver. , 2008, , .		34
65	Testing Software Product Lines Using Incremental Test Generation. , 2008, , .		34
66	Juzi. , 2008, , .		40
67	Deryaft. , 2008, , .		6
68	Assertion-based repair of complex data structures. , 2007, , .		47
69	Parallel test generation and execution with Korat. , 2007, , .		48
70	Evaluation of Semantic Interference Detection in Parallel Changes: an Exploratory Experiment. Conference on Software Maintenance, Proceedings of the, 2007, , .	0.0	12
71	A Case for White-box Testing Using Declarative Specifications Poster Abstract. , 2007, , .		5
72	A Case for White-box Testing Using Declarative Specifications Poster Abstract. , 2007, , .		0

#	Article	IF	Citations
73	Whispec. , 2007, , .		18
74	Generalizing symbolic execution to library classes. , 2005, , .		23
75	Repairing Structurally Complex Data. Lecture Notes in Computer Science, 2005, , 123-138.	1.3	31
76	Software assurance by bounded exhaustive testing. , 2004, , .		57
77	Software assurance by bounded exhaustive testing. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2004, 29, 133-142.	0.7	4
78	TestEra: Specification-Based Testing of Java Programs Using SAT. Automated Software Engineering, 2004, 11, 403-434.	2.9	137
79	Exploring very large state spaces using genetic algorithms. International Journal on Software Tools for Technology Transfer, 2004, 6, 117-127.	1.9	55
80	A Case for Efficient Solution Enumeration. Lecture Notes in Computer Science, 2004, , 272-286.	1.3	16
81	Generalized Symbolic Execution for Model Checking and Testing. Lecture Notes in Computer Science, 2003, , 553-568.	1.3	243
82	Korat. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2002, , .	0.7	271
83	Korat. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2002, 27, 123-133.	0.7	173
84	Checking Java Implementation of a Naming Architecture Using Testera. Electronic Notes in Theoretical Computer Science, 2001, 55, 322-342.	0.9	10
85	Testing an Intentional Naming Scheme Using Genetic Algorithms. Lecture Notes in Computer Science, 2001, , 358-372.	1.3	2
86	Is the Java type system sound?. Theory and Practice of Object Systems, 1999, 5, 3-24.	0.7	29
87	Is the Java type system sound?. , 1999, 5, 3.		15