

Willem Visser

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/11575257/publications.pdf>

Version: 2024-02-01

48
papers

3,267
citations

430874

18
h-index

434195

31
g-index

49
all docs

49
docs citations

49
times ranked

1270
citing authors

#	ARTICLE	IF	CITATIONS
1	Java Ranger at SV-COMP 2020 (Competition Contribution). Lecture Notes in Computer Science, 2020, , 393-397.	1.3	9
2	Java Ranger: statically summarizing regions for efficient symbolic execution of Java. , 2020, , .		14
3	Java Pathfinder at SV-COMP 2019 (Competition Contribution). Lecture Notes in Computer Science, 2019, , 224-228.	1.3	5
4	Symbolic Pathfinder for SV-COMP. Lecture Notes in Computer Science, 2019, , 239-243.	1.3	9
5	Monte Carlo Tree Search for Finding Costly Paths in Programs. Lecture Notes in Computer Science, 2018, , 123-138.	1.3	1
6	Veritesting Challenges in Symbolic Execution of Java. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2018, 42, 1-5.	0.7	2
7	Probabilistic programming for Java using symbolic execution and model counting. , 2017, , .		3
8	JFIX: semantics-based repair of Java programs via symbolic PathFinder. , 2017, , .		51
9	Addressing challenges in obtaining high coverage when model checking Android applications. , 2017, , .		1
10	StateComparator. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2017, 41, 1-5.	0.7	2
11	S3: syntax- and semantic-guided repair synthesis via programming by examples. , 2017, , .		126
12	What makes killing a mutant hard. , 2016, , .		17
13	BLISS: Improved Symbolic Execution by\ Bounded Lazy Initialization with SAT Support. IEEE Transactions on Software Engineering, 2015, , 1-1.	5.6	19
14	Generation of Library Models for Verification of Android Applications. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2015, 40, 1-5.	0.7	13
15	Model Counting for Complex Data Structures. Lecture Notes in Computer Science, 2015, , 222-241.	1.3	16
16	Environment Modeling Using Runtime Values for JPF-Android. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2015, 40, 1-5.	0.7	4
17	Compositional solution space quantification for probabilistic software analysis. , 2014, , .		25
18	Software engineering and automated deduction. , 2014, , .		5

#	ARTICLE	IF	CITATIONS
19	Statistical symbolic execution with informed sampling. , 2014, , .		26
20	Exact and approximate probabilistic symbolic execution for nondeterministic programs. , 2014, , .		29
21	Compositional solution space quantification for probabilistic software analysis. ACM SIGPLAN Notices, 2014, 49, 123-132.	0.2	18
22	Town hall discussion of SE 2004 revisions (panel). , 2013, , .		0
23	Symbolic PathFinder: integrating symbolic execution with model checking for Java bytecode analysis. Automated Software Engineering, 2013, 20, 391-425.	2.9	130
24	The hidden models of model checking. Software and Systems Modeling, 2012, 11, 541.	2.7	14
25	Symbolic execution for software testing in practice. , 2011, , .		222
26	Symbolic execution with mixed concrete-symbolic solving. , 2011, , .		61
27	Symbolic execution with abstraction. International Journal on Software Tools for Technology Transfer, 2009, 11, 53-67.	1.9	44
28	A survey of new trends in symbolic execution for software testing and analysis. International Journal on Software Tools for Technology Transfer, 2009, 11, 339-353.	1.9	167
29	Efficient Testing of Concurrent Programs with Abstraction-Guided Symbolic Execution. Lecture Notes in Computer Science, 2009, , 174-191.	1.3	28
30	Variably interprocedural program analysis for runtime error detection. , 2007, , .		39
31	JPFâ€“SE: A Symbolic Execution Extension to Java PathFinder. , 2007, , 134-138.		120
32	Verifying Multi-agent Programs by Model Checking. Autonomous Agents and Multi-Agent Systems, 2006, 12, 239-256.	2.1	118
33	Test input generation for java containers using state matching. , 2006, , .		129
34	Symbolic Execution with Abstract Subsumption Checking. Lecture Notes in Computer Science, 2006, , 163-181.	1.3	29
35	Combining test case generation and runtime verification. Theoretical Computer Science, 2005, 336, 209-234.	0.9	72
36	Concrete Model Checking with Abstract Matching and Refinement. Lecture Notes in Computer Science, 2005, , 52-66.	1.3	28

#	ARTICLE	IF	CITATIONS
37	Verifiable Multi-agent Programs. Lecture Notes in Computer Science, 2004, , 72-89.	1.3	19
38	Test input generation with java PathFinder. , 2004, , .		309
39	Verification of Java Programs Using Symbolic Execution and Invariant Generation. Lecture Notes in Computer Science, 2004, , 164-181.	1.3	63
40	Heuristics for model checking Java programs. International Journal on Software Tools for Technology Transfer, 2004, 6, 260-276.	1.9	68
41	Test input generation with java PathFinder. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2004, 29, 97-107.	0.7	100
42	Model Checking Programs. Automated Software Engineering, 2003, 10, 203-232.	2.9	697
43	Finding feasible abstract counter-examples. International Journal on Software Tools for Technology Transfer, 2003, 5, 34-48.	1.9	17
44	Generalized Symbolic Execution for Model Checking and Testing. Lecture Notes in Computer Science, 2003, , 553-568.	1.3	243
45	Model Checking Multi-Agent Programs with CASP. Lecture Notes in Computer Science, 2003, , 110-113.	1.3	25
46	Model checking Java programs using structural heuristics. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2002, , .	0.7	41
47	Heuristic Model Checking for Java Programs. Lecture Notes in Computer Science, 2002, , 242-245.	1.3	19
48	Verification of time partitioning in the DEOS scheduler kernel. , 2000, , .		51