Keijo Heljanko

List of Publications by Year
in descending order

86	1,449	489802	445137
papers	citations	18 h-index	33 g-index

90	90	90	1190
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	Surrogateâ€based optimization of a periodic rescheduling algorithm. AICHE Journal, 2022, 68, .	1.8	2
2	Progress in Certifying Hardware Model Checking Results. Lecture Notes in Computer Science, 2021, , 363-386.	1.0	2
3	Dynamic Process Intensification via Data-Driven Dynamic Optimization: Concept and Application to Ternary Distillation. Industrial & Engineering Chemistry Research, 2021, 60, 10265-10275.	1.8	1
4	Distributed hybrid-indexing of compressed pan-genomes for scalable and fast sequence alignment. PLoS ONE, 2021, 16, e0255260.	1.1	1
5	An optimal cut-off algorithm for parameterised refinement checking. Science of Computer Programming, 2020, 198, 102517.	1.5	0
6	Synergistic and Intelligent Process Optimization: First Results and Open Challenges. Industrial & Engineering Chemistry Research, 2020, 59, 16684-16694.	1.8	4
7	Data-Driven Approach to Grade Change Scheduling Optimization in a Paper Machine. Industrial & Engineering Chemistry Research, 2020, 59, 8281-8294.	1.8	4
8	Reinforcement learning of adaptive online rescheduling timing and computing time allocation. Computers and Chemical Engineering, 2020, 141, 106994.	2.0	14
9	IoTEF: A Federated Edge-Cloud Architecture for Fault-Tolerant IoT Applications. Journal of Grid Computing, 2020, 18, 57-80.	2.5	36
10	Dartagnan: Bounded Model Checking for Weak Memory Models (Competition Contribution). Lecture Notes in Computer Science, 2020, , 378-382.	1.0	8
11	Scalable Reference Genome Assembly from Compressed Pan-Genome Index with Spark. Lecture Notes in Computer Science, 2020, , 68-84.	1.0	2
12	Exploiting Event Log Event Attributes in RNN Based Prediction. Lecture Notes in Business Information Processing, 2020, , 67-85.	0.8	3
13	SparkBeagle. , 2020, , .		1
14	Access Time Improvement Framework for Standardized IoT Gateways. , 2019, , .		6
15	Classifying Process Instances Using Recurrent Neural Networks. Lecture Notes in Business Information Processing, 2019, , 313-324.	0.8	17
16	BMC for Weak Memory Models: Relation Analysis for Compact SMT Encodings. Lecture Notes in Computer Science, 2019, , 355-365.	1.0	28
17	Exploiting Event Log Event Attributes in RNN Based Prediction. Communications in Computer and Information Science, 2019, , 405-416.	0.4	8
18	Certifying Hardware Model Checking Results. Lecture Notes in Computer Science, 2019, , 498-502.	1.0	3

#	Article	IF	Citations
19	ViraPipe: scalable parallel pipeline for viral metagenome analysis from next generation sequencing reads. Bioinformatics, 2018, 34, 928-935.	1.8	14
20	CEFIoT: A fault-tolerant IoT architecture for edge and cloud. , 2018, , .		43
21	Dynamic Cut-Off Algorithm for Parameterised Refinement Checking. Lecture Notes in Computer Science, 2018, , 256-276.	1.0	2
22	Structural Feature Selection for Event Logs. Lecture Notes in Business Information Processing, 2018, , 20-35.	0.8	7
23	Testing Programs with Contextual Unfoldings. Transactions on Embedded Computing Systems, 2018, 17, 1-25.	2.1	1
24	BMC with Memory Models as Modules. , 2018, , .		10
25	Minimizing Test Suites with Unfoldings of Multithreaded Programs. Transactions on Embedded Computing Systems, 2017, 16, 1-24.	2.1	7
26	Hardware model checking competition 2017. , 2017, , .		19
27	The FMCAD 2017 graduate student forum. , 2017, , .		3
28	Portability Analysis for Weak Memory Models porthos: One Tool for all Models. Lecture Notes in Computer Science, 2017, , 299-320.	1.0	9
29	When Do We Not Need Complex Assume-Guarantee Rules?. Transactions on Embedded Computing Systems, 2017, 16, 1-25.	2.1	2
30	Hardware Model Checking Competition 2014: An Analysis and Comparison of Model Checkers and Benchmarks. Journal of Satisfiability, Boolean Modeling and Computation, 2016, 9, 135-172.	1.2	12
31	Synchronous counting and computational algorithm design. Journal of Computer and System Sciences, 2016, 82, 310-332.	0.9	13
32	Assessing Big Data SQL Frameworks for Analyzing Event Logs. , 2016, , .		6
33	LCTD: Test-guided proofs for C programs on LLVM. Journal of Logical and Algebraic Methods in Programming, 2016, 85, 1292-1317.	0.4	1
34	LCTD: Tests-Guided Proofs for C Programs on LLVM. Lecture Notes in Computer Science, 2016, , 927-929.	1.0	2
35	Unfolding Based Minimal Test Suites for Testing Multithreaded Programs. , 2015, , .		3
36	Verifying large modular systems using iterative abstraction refinement. Reliability Engineering and System Safety, 2015, 139, 120-130.	5.1	3

#	Article	IF	Citations
37	When Do We (Not) Need Complex Assume-Guarantee Rules?. , 2015, , .		2
38	Unfolding based automated testing of multithreaded programs. Automated Software Engineering, 2015, 22, 475-515.	2.2	8
39	Unfolding-Based Process Discovery. Lecture Notes in Computer Science, 2015, , 31-47.	1.0	12
40	Parametrised Modal Interface Automata. Transactions on Embedded Computing Systems, 2015, 14, 1-25.	2.1	5
41	SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop. Bioinformatics, 2014, 30, 119-120.	1.8	85
42	Testing Multithreaded Programs with Contextual Unfoldings and Dynamic Symbolic Execution. , 2014, , .		8
43	A symbolic model checking approach to verifying satellite onboard software. Science of Computer Programming, 2014, 82, 44-55.	1.5	19
44	Lightweight State Capturing for Automated Testing of Multithreaded Programs. Lecture Notes in Computer Science, 2014, , 187-203.	1.0	2
45	Parametrised Compositional Verification with Multiple Process and Data Types. , 2013, , .		4
46	LCT: A Parallel Distributed Testing Tool for Multithreaded Java Programs. Electronic Notes in Theoretical Computer Science, 2013, 296, 253-259.	0.9	5
47	Increasing Confidence in Liveness Model Checking Results with Proofs. Lecture Notes in Computer Science, 2013, , 32-43.	1.0	7
48	Asynchronous Multi-core Incremental SAT Solving. Lecture Notes in Computer Science, 2013, , 139-153.	1.0	6
49	Scripting for large-scale sequencing based on Hadoop. EMBnet Journal, 2013, 19, 84.	0.2	1
50	Concurrent Clause Strengthening. Lecture Notes in Computer Science, 2013, , 116-132.	1.0	5
51	Using unfoldings in automated testing of multithreaded programs. , 2012, , .		18
52	Message from the Programme Co-chairs. , 2012, , .		0
53	Hadoop-BAM: directly manipulating next generation sequencing data in the cloud. Bioinformatics, 2012, 28, 876-877.	1.8	118
54	Improving Dynamic Partial Order Reductions for Concolic Testing. , 2012, , .		23

#	Article	IF	Citations
55	Model checking of safety-critical software in the nuclear engineering domain. Reliability Engineering and System Safety, 2012, 105, 104-113.	5.1	51
56	Exploiting step semantics for efficient bounded model checking of asynchronous systems. Science of Computer Programming, 2012, 77, 1095-1121.	1.5	8
57	Solving parity games by a reduction to SAT. Journal of Computer and System Sciences, 2012, 78, 430-440.	0.9	11
58	Efficient model checking of PSL safety properties. IET Computers and Digital Techniques, 2011, 5, 479.	0.9	3
59	Efficient Model Checking of PSL Safety Properties. , 2010, , .		6
60	Experimental Comparison of Concolic and Random Testing for Java Card Applets. Lecture Notes in Computer Science, 2010, , 22-39.	1.0	4
61	The LIME Interface Specification Language and Runtime Monitoring Tool. Lecture Notes in Computer Science, 2009, , 93-100.	1.0	12
62	Analyzing Context-Free Grammars Using an Incremental SAT Solver. Lecture Notes in Computer Science, 2008, , 410-422.	1.0	35
63	Symbolic Step Encodings for Object Based Communicating State Machines. Lecture Notes in Computer Science, 2008, , 96-112.	1.0	8
64	Planning as satisfiability: parallel plans and algorithms for plan search. Artificial Intelligence, 2006, 170, 1031-1080.	3.9	110
65	Linear Encodings of Bounded LTL Model Checking. Logical Methods in Computer Science, 2006, 2, .	0.4	116
66	Incremental and Complete Bounded Model Checking for Full PLTL. Lecture Notes in Computer Science, 2005, , 98-111.	1.0	47
67	BMC via on-the-fly determinization. International Journal on Software Tools for Technology Transfer, 2005, 7, 89-101.	1.7	13
68	Simple Is Better: Efficient Bounded Model Checking for Past LTL. Lecture Notes in Computer Science, 2005, , 380-395.	1.0	30
69	Parallel Encodings of Classical Planning as Satisfiability. Lecture Notes in Computer Science, 2004, , 307-319.	1.0	7
70	Simple Bounded LTL Model Checking. Lecture Notes in Computer Science, 2004, , 186-200.	1.0	49
71	BMC via On-the-Fly Determinization. Electronic Notes in Theoretical Computer Science, 2003, 89, 561-577.	0.9	3
72	Bounded LTL model checking with stable models. Theory and Practice of Logic Programming, 2003, 3, 519-550.	1.1	52

#	Article	IF	Citations
73	Testing LTL formula translation into BÃ¼chi automata. International Journal on Software Tools for Technology Transfer, 2002, 4, 57-70.	1.7	32
74	Parallelisation of the Petri Net Unfolding Algorithm. Lecture Notes in Computer Science, 2002, , 371-385.	1.0	21
75	Bounded Reachability Checking with Process Semantics. Lecture Notes in Computer Science, 2001, , 218-232.	1.0	33
76	Implementing LTL model checking with net unfoldings. Lecture Notes in Computer Science, 2001, , 37-56.	1.0	25
77	Bounded LTL Model Checking with Stable Models. Lecture Notes in Computer Science, 2001, , 200-212.	1.0	14
78	Coping With Strong Fairness. Fundamenta Informaticae, 2000, 43, 175-193.	0.3	18
79	Model Checking with Finite Complete Prefixes Is PSPACE-Complete. Lecture Notes in Computer Science, 2000, , 108-122.	1.0	5
80	A New Unfolding Approach to LTL Model Checking. Lecture Notes in Computer Science, 2000, , 475-486.	1.0	19
81	Using Logic Programs with Stable Model Semantics to Solve Deadlock and Reachability Problems for 1-Safe Petri Nets. Fundamenta Informaticae, 1999, 37, 247-268.	0.3	24
82	Using Logic Programs with Stable Model Semantics to Solve Deadlock and Reachability Problems for 1-Safe Petri Nets. Lecture Notes in Computer Science, 1999, , 240-254.	1.0	13
83	Prod 3.2 An advanced tool for efficient reachability analysis. Lecture Notes in Computer Science, 1997, , 472-475.	1.0	19
84	Specification coverage aided test selection. , 0, , .		5
85	Complexity Results for Checking Distributed Implementability. , 0, , .		5
86	Tarmo: A Framework for Parallelized Bounded Model Checking. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 14, 62-76.	0.8	13