Mohammad Reza Mousavi

List of Publications by Year
in descending order

49 papers	522 citations	840776 11 h-index	839539 18 g-index
54 all docs	54 docs citations	54 times ranked	296 citing authors

#	Article	IF	Citations
1	Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review. ACM Transactions on Software Engineering and Methodology, 2023, 32, 1-61.	6.0	5
2	Learning by sampling: learning behavioral family models from software product lines. Empirical Software Engineering, 2021, 26, 1.	3.9	8
3	Efficient state synchronisation in model-based testing through reinforcement learning. , 2021, , .		3
4	Connected Automated Driving: A Model-Based Approach to the Analysis of Basic Awareness Services. , 2020, , .		1
5	Conformance-Based Doping Detection for Cyber-Physical Systems. Lecture Notes in Computer Science, 2020, , 59-77.	1.3	3
6	Learning from Difference. , 2019, , .		8
7	Modal transition system encoding of featured transition systems. Journal of Logical and Algebraic Methods in Programming, 2019, 106, 1-28.	0.5	5
8	Trusted Autonomous Vehicles: an Interactive Exhibit. , 2019, , .		3
9	Hierarchical featured state machines. Science of Computer Programming, 2019, 171, 67-88.	1.9	6
10	Learning to Reuse: Adaptive Model Learning for Evolving Systems. Lecture Notes in Computer Science, 2019, , 138-156.	1.3	10
11	Comparative Expressiveness of Product Line Calculus of Communicating Systems and 1-Selecting Modal Transition Systems. Lecture Notes in Computer Science, 2019, , 490-503.	1.3	1
12	Sound conformance testing for cyber-physical systems: Theory and implementation. Science of Computer Programming, 2018, 162, 35-54.	1.9	13
13	A classification of product sampling for software product lines. , 2018, , .		47
14	Basic behavioral models for software product lines: Revisited. Science of Computer Programming, 2018, 168, 171-185.	1.9	8
15	Model Learning and Model-Based Testing. Lecture Notes in Computer Science, 2018, , 74-100.	1.3	32
16	Validated Test Models for Software Product Lines: Featured Finite State Machines. Lecture Notes in Computer Science, 2017, , 210-227.	1.3	8
17	Generation of Failure Models through Automata Learning. , 2016, , .		5
18	Basic behavioral models for software product lines: Expressiveness and testing pre-orders. Science of Computer Programming, 2016, 123, 42-60.	1.9	24

#	Article	IF	Citations
19	Input–output conformance testing for software product lines. Journal of Logical and Algebraic Methods in Programming, 2016, 85, 1131-1153.	0.5	12
20	Synchrony and asynchrony in conformance testing. Software and Systems Modeling, 2015, 14, 149-172.	2.7	3
21	Automatic Consequence Analysis of Automotive Standards (AUTO-CAAS). , 2015, , .		3
22	Input-output conformance testing based on featured transition systems. , 2014, , .		8
23	Preface: Special section on foundations of coordination languages and software architectures (selected papers from FOCLASA'10). Science of Computer Programming, 2014, 80, 1-2.	1.9	1
24	On the Complexity of Input Output Conformance Testing. Lecture Notes in Computer Science, 2014, , 291-309.	1.3	2
25	Early Fault Detection in DSLs Using SMT Solving and Automated Debugging. Lecture Notes in Computer Science, 2013, , 182-196.	1.3	8
26	Modular Semantics for Transition System Specifications with Negative Premises. Lecture Notes in Computer Science, 2013, , 46-60.	1.3	1
27	Integrating Model-Based and Constraint-Based Testing Using SpecExplorer. , 2012, , .		7
28	Formal modeling of evolving self-adaptive systems. Science of Computer Programming, 2012, 78, 3-26.	1.9	38
29	Rule formats for determinism and idempotence. Science of Computer Programming, 2012, 77, 889-907.	1.9	9
30	Mechanized Extraction of Topology Anti-patterns in Wireless Networks. Lecture Notes in Computer Science, 2012, , 158-173.	1.3	7
31	Process Algebra as a Common Framework for Hardware/Software Coverification. IEEE Embedded Systems Letters, 2011, 3, 9-12.	1.9	0
32	Formal Analysis of SystemC Designs in Process Algebra. Fundamenta Informaticae, 2011, 107, 19-42.	0.4	2
33	Symmetry and partial order reduction techniques in model checking Rebeca. Acta Informatica, 2010, 47, 33-66.	0.5	27
34	Lifting non-finite axiomatizability results to extensions of process algebras. Acta Informatica, 2010, 47, 147-177.	0.5	8
35	Towards model checking executable UML specifications in mCRL2. Innovations in Systems and Software Engineering, 2010, 6, 83-90.	2.1	33
36	On Rule Formats for Zero and Unit Elements. Electronic Notes in Theoretical Computer Science, 2010, 265, 145-160.	0.9	1

#	Article	IF	Citations
37	PobSAM: Policy-based Managing of Actors in Self-Adaptive Systems. Electronic Notes in Theoretical Computer Science, 2010, 263, 129-143.	0.9	22
38	Semantics and expressiveness of ordered SOS. Information and Computation, 2009, 207, 85-119.	0.7	12
39	On Well-Foundedness and Expressiveness of Promoted Tyft. Electronic Notes in Theoretical Computer Science, 2007, 175, 45-56.	0.9	1
40	A Congruence Rule Format with Universal Quantification. Electronic Notes in Theoretical Computer Science, 2007, 192, 109-124.	0.9	2
41	Prototyping SOS Meta-theory in Maude. Electronic Notes in Theoretical Computer Science, 2006, 156, 135-150.	0.9	9
42	Formal Semantics and Analysis of Component Connectors in Reo. Electronic Notes in Theoretical Computer Science, 2006, 154, 83-99.	0.9	22
43	A Hierarchy of SOS Rule Formats. Electronic Notes in Theoretical Computer Science, 2006, 156, 3-25.	0.9	11
44	A syntactic commutativity format for SOS. Information Processing Letters, 2005, 93, 217-223.	0.6	23
45	Notions of bisimulation and congruence formats for SOS with data. Information and Computation, 2005, 200, 107-147.	0.7	32
46	Extending HSI Test Generation Method for Software Product Lines. Computer Journal, 0, , .	2.4	6
47	Spinal Test Suites for Software Product Lines. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 141, 44-55.	0.8	7
48	Decomposability in Input Output Conformance Testing. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 111, 51-66.	0.8	5
49	Algebraic Meta-Theory of Processes with Data. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 120, 63-77.	0.8	2