Maurizio Proietti

List of Publications by Year
in descending order

74	918	623734	552781
papers	citations	14 h-index	26 g-index

81	81	81	220
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	Transformation of logic programs: Foundations and techniques. The Journal of Logic Programming, 1994, 19-20, 261-320.	1.7	143
2	Rules and strategies for transforming functional and logic programs. ACM Computing Surveys, 1996, 28, 360-414.	23.0	108
3	VeriMAP: A Tool for Verifying Programs through Transformations. Lecture Notes in Computer Science, 2014, , 568-574.	1.3	52
4	Unfolding-definition-folding, in this order, for avoiding unnecessary variables in logic programs. Theoretical Computer Science, 1995, 142, 89-124.	0.9	47
5	Synthesis and transformation of logic programs using unfold/fold proofs. The Journal of Logic Programming, 1999, 41, 197-230.	1.7	36
6	Generalization strategies for the verification of infinite state systems. Theory and Practice of Logic Programming, 2013, 13, 175-199.	1.5	32
7	The loop absorption and the generalization strategies for the development of logic programs and partial deduction. The Journal of Logic Programming, 1993, 16, 123-161.	1.7	29
8	Unfolding â€" definition â€" folding, in this order, for avoiding unnecessary variables in logic programs. Lecture Notes in Computer Science, 1991, , 347-358.	1.3	23
9	Relational Verification Through Horn Clause Transformation. Lecture Notes in Computer Science, 2016, , 147-169.	1.3	22
10	Solving Horn Clauses on Inductive Data Types Without Induction. Theory and Practice of Logic Programming, 2018, 18, 452-469.	1.5	19
11	Analysis and Transformation of Constrained Horn Clauses for Program Verification. Theory and Practice of Logic Programming, 2022, 22, 974-1042.	1.5	19
12	Ontology-Based Querying of Composite Services. Lecture Notes in Computer Science, 2012, , 159-180.	1.3	17
13	Automated Strategies for Specializing Constraint Logic Programs. Lecture Notes in Computer Science, 2001, , 125-146.	1.3	17
14	Reducing nondeterminism while specializing logic programs. , 1997, , .		16
15	Semantics preserving transformation rules for Prolog. , 1991, , .		15
16	Proving correctness of imperative programs by linearizing constrained Horn clauses. Theory and Practice of Logic Programming, 2015, 15, 635-650.	1.5	15
17	A comparative revisitation of some program transformation techniques. Lecture Notes in Computer Science, 1996, , 355-385.	1.3	15
18	A Semantic Framework for Knowledge Management in Virtual Innovation Factories. International Journal of Information System Modeling and Design, 2013, 4, 70-92.	1.1	14

#	Article	IF	Citations
19	Semantics preserving transformation rules for Prolog. ACM SIGPLAN Notices, 1991, 26, 274-284.	0.2	12
20	Verifying programs via iterated specialization. , 2013, , .		11
21	Querying Semantically Enriched Business Processes. Lecture Notes in Computer Science, 2011, , 294-302.	1.3	10
22	Verification of Sets of Infinite State Processes Using Program Transformation. Lecture Notes in Computer Science, 2002, , 111-128.	1.3	10
23	A Rule-based Verification Strategy for Array Manipulating Programs. Fundamenta Informaticae, 2015, 140, 329-355.	0.4	9
24	Predicate Pairing for program verification. Theory and Practice of Logic Programming, 2018, 18, 126-166.	1.5	9
25	Verifying Array Programs by Transforming Verification Conditions. Lecture Notes in Computer Science, 2014, , 182-202.	1.3	9
26	Improving Reachability Analysis of Infinite State Systems by Specialization. Fundamenta Informaticae, 2012, 119, 281-300.	0.4	8
27	Removing Algebraic Data Types from Constrained Horn Clauses Using Difference Predicates. Lecture Notes in Computer Science, 2020, , 83-102.	1.3	8
28	Program Derivation = Rules + Strategies. Lecture Notes in Computer Science, 2002, , 273-309.	1.3	7
29	Deciding Full Branching Time Logic by Program Transformation. Lecture Notes in Computer Science, 2010, , 5-21.	1.3	7
30	A theory of logic program specialization and generalization for dealing with input data properties. Lecture Notes in Computer Science, 1996, , 386-408.	1.3	6
31	Synthesis of Programs from Unfold/Fold Proofs. Workshops in Computing, 1994, , 141-158.	0.4	6
32	Behavioral Reasoning on Semantic Business Processes in a Rule-Based Framework. Communications in Computer and Information Science, 2014, , 293-313.	0.5	6
33	Program Specialization for Verifying Infinite State Systems: An Experimental Evaluation. Lecture Notes in Computer Science, 2011, , 164-183.	1.3	6
34	Verification of Imperative Programs by Constraint Logic Program Transformation. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 129, 186-210.	0.8	6
35	Verifying Catamorphism-Based Contracts using Constrained Horn Clauses. Theory and Practice of Logic Programming, 0, , 1-18.	1.5	6
36	Transformations of logic programs on infinite lists. Theory and Practice of Logic Programming, 2010, 10, 383-399.	1.5	5

#	Article	IF	Citations
37	Controlling Polyvariance for Specialization-based Verification. Fundamenta Informaticae, 2013, 124, 483-502.	0.4	5
38	Future directions in program transformation. ACM Computing Surveys, 1996, 28, 171.	23.0	5
39	Proving Properties of Constraint Logic Programs by Eliminating Existential Variables. Lecture Notes in Computer Science, 2006, , 179-195.	1.3	5
40	Program Derivation via List Introduction. IFIP Advances in Information and Communication Technology, 1997, , 296-323.	0.7	5
41	Rules and Strategies for Contextual Specialization of Constraint Logic Programs. Electronic Notes in Theoretical Computer Science, 2000, 30, 129-144.	0.9	4
42	The List Introduction Strategy for the Derivation of Logic Programs. Formal Aspects of Computing, 2002, 13, 233-251.	1.8	4
43	Transformations of logic programs with goals as arguments. Theory and Practice of Logic Programming, 2004, 4, 495-537.	1.5	4
44	Synthesizing Concurrent Programs Using Answer Set Programming. Fundamenta Informaticae, 2012, 120, 205-229.	0.4	4
45	Constraint-based correctness proofs for logic program transformations. Formal Aspects of Computing, 2012, 24, 569-594.	1.8	4
46	Specialization with Constrained Generalization for Software Model Checking. Lecture Notes in Computer Science, 2013, , 51-70.	1.3	4
47	Proving Theorems by Program Transformation. Fundamenta Informaticae, 2013, 127, 115-134.	0.4	3
48	Program Verification using Constraint Handling Rules and Array Constraint Generalizations*. Fundamenta Informaticae, 2017, 150, 73-117.	0.4	3
49	Enhancing partial deduction via unfold/fold rules. Lecture Notes in Computer Science, 1997, , 146-168.	1.3	3
50	The Transformational Approach to Program Development. Lecture Notes in Computer Science, 2010, , 112-135.	1.3	3
51	Improving Reachability Analysis of Infinite State Systems by Specialization. Lecture Notes in Computer Science, 2011, , 165-179.	1.3	3
52	Lemma Generation for Horn Clause Satisfiability: A Preliminary Study. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 299, 4-18.	0.8	3
53	Satisfiability of constrained Horn clauses on algebraic data types: A transformation-based approach. Journal of Logic and Computation, 0, , .	0.8	3
54	Developing correct and efficient logic programs by transformation. Knowledge Engineering Review, 1996, 11, 347-360.	2.6	2

#	Article	IF	Citations
55	Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism. Higher-Order and Symbolic Computation, 2005, 18, 121-210.	0.3	2
56	Semantics and Controllability of Time-Aware Business Processes*. Fundamenta Informaticae, 2019, 165, 205-244.	0.4	2
57	Predicate Pairing with Abstraction for Relational Verification. Lecture Notes in Computer Science, 2018, , 289-305.	1.3	2
58	Verification of Programs by Combining Iterated Specialization with Interpolation. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 169, 3-18.	0.8	2
59	Verification of Time-Aware Business Processes Using Constrained Horn Clauses. Lecture Notes in Computer Science, 2017, , 38-55.	1.3	2
60	Verifying Controllability of Time-Aware Business Processes. Lecture Notes in Computer Science, 2017, , 103-118.	1.3	2
61	Property-Based Test Case Generators for Free. Lecture Notes in Computer Science, 2019, , 186-206.	1.3	2
62	Proving Properties of Sorting Programs: A Case Study in Horn Clause Verification. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 296, 48-75.	0.8	2
63	Program specialization via algorithmic unfold/fold transformations. ACM Computing Surveys, 1998, 30, 6.	23.0	1
64	Totally correct logic program transformations viaÂwell-founded annotations. Higher-Order and Symbolic Computation, 2008, 21, 193-234.	0.3	1
65	Solving Horn Clauses on Inductive Data Types Without Induction â€“ ERRATUM. Theory and Practice of Logic Programming, 2019, 19, 629.	1.5	1
66	Transformation Rules for Logic Programs with Goals as Arguments. Lecture Notes in Computer Science, 2000, , 176-195.	1.3	1
67	Transformational Verification of Parameterized Protocols Using Array Formulas. Lecture Notes in Computer Science, 2006, , 23-43.	1.3	1
68	A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs. Lecture Notes in Computer Science, 2008, , 284-300.	1.3	1
69	Bounded Symbolic Execution for Runtime Error Detection of Erlang Programs. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 278, 19-26.	0.8	1
70	A Folding Rule for Eliminating Existential Variables from Constraint Logic Programs. Fundamenta Informaticae, 2009, 96, 373-393.	0.4	0
71	Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism. , 2008, , 130-177.		0
72	Using Real Relaxations during Program Specialization. Lecture Notes in Computer Science, 2012, , 106-122.	1.3	0

#	Article	IF	Citations
73	Removing Unnecessary Variables from Horn Clause Verification Conditions. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 219, 49-55.	0.8	0
74	Transformational Verification of Quicksort. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 320, 95-109.	0.8	0