Manuel V Hermenegildo

List of Publications by Year
in descending order

158	2,239	304368	315357
papers	citations	22 h-index	38 g-index

171	171	171	508
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	Analysis and Transformation of Constrained Horn Clauses for Program Verification. Theory and Practice of Logic Programming, 2022, 22, 974-1042.	1.1	19
2	Fifty Years of Prolog and Beyond. Theory and Practice of Logic Programming, 2022, 22, 776-858.	1.1	9
3	Testing Your (Static Analysis) Truths. Lecture Notes in Computer Science, 2021, , 271-292.	1.0	1
4	Incremental and Modular Context-sensitive Analysis. Theory and Practice of Logic Programming, 2021, 21, 196-243.	1.1	6
5	VeriFly: On-the-fly Assertion Checking via Incrementality. Theory and Practice of Logic Programming, 2021, 21, 768-784.	1.1	2
6	An Integrated Approach to Assertion-Based Random Testing in Prolog. Lecture Notes in Computer Science, 2020, , 159-176.	1.0	4
7	Incremental Analysis of Logic Programs with Assertions and Open Predicates. Lecture Notes in Computer Science, 2020, , 36-56.	1.0	6
8	Computing Abstract Distances in Logic Programs. Lecture Notes in Computer Science, 2020, , 57-72.	1.0	5
9	Cost Analysis of Smart Contracts Via Parametric Resource Analysis. Lecture Notes in Computer Science, 2020, , 7-31.	1.0	9
10	A General Framework for Static Cost Analysis of Parallel Logic Programs. Lecture Notes in Computer Science, 2020, , 19-35.	1.0	3
11	Some trade-offs in reducing the overhead of assertion run-time checks via static analysis. Science of Computer Programming, 2018, 155, 3-26.	1.5	5
12	Static Performance Guarantees for Programs with Runtime Checks. , 2018, , .		6
13	Inferring Energy Bounds via Static Program Analysis and Evolutionary Modeling of Basic Blocks. Lecture Notes in Computer Science, 2018, , 54-72.	1.0	2
14	Exploiting Term Hiding to Reduce Run-Time Checking Overhead. Lecture Notes in Computer Science, 2018, , 99-115.	1.0	3
15	Semantic code browsing. Theory and Practice of Logic Programming, 2016, 16, 721-737.	1.1	4
16	Reducing the overhead of assertion run-time checks via static analysis. , 2016, , .		7
17	ENTRA: Whole-systems energy transparency. Microprocessors and Microsystems, 2016, 47, 278-286.	1.8	13
18	Description and Optimization of Abstract Machines in a Dialect of Prolog. Theory and Practice of Logic Programming, 2016, 16, 1-58.	1.1	2

#	Article	IF	Citations
19	Practical run-time checking via unobtrusive property caching. Theory and Practice of Logic Programming, 2015, 15, 726-741.	1.1	2
20	Pre-indexed Terms for Prolog. Lecture Notes in Computer Science, 2015, , 317-331.	1.0	0
21	Assertion-based Debugging of Higher-Order (C)LP Programs. , 2014, , .		4
22	Resource Usage Analysis of Logic Programs via Abstract Interpretation Using Sized Types. Theory and Practice of Logic Programming, 2014, 14, 739-754.	1.1	21
23	Energy Consumption Analysis of Programs Based on XMOS ISA-Level Models. Lecture Notes in Computer Science, 2014, , 72-90.	1.0	27
24	A sharing-based approach to supporting adaptation in service compositions. Computing (Vienna/New) Tj ETQq0 0 0 rgBT /Overlock 10 T	3.2	1
25	Supporting Pruning in Tabled LP. Lecture Notes in Computer Science, 2013, , 60-76.	1.0	0
26	Analyzing service-oriented systems using their data and structure. , 2012, , .		0
27	Lightweight compilation of (C)LP to JavaScript. Theory and Practice of Logic Programming, 2012, 12, 755-773.	1.1	5
28	An overview of Ciao and its design philosophy. Theory and Practice of Logic Programming, 2012, 12, 219-252.	1.1	85
29	Exploring the impact of inaccuracy and imprecision of QoS assumptions on proactive constraint-based QoS prediction for service orchestrations. , 2012, , .		1
30	Certificate size reduction in abstraction-carrying code. Theory and Practice of Logic Programming, 2012, 12, 283-318.	1.1	5
31	A General Implementation Framework for Tabled CLP. Lecture Notes in Computer Science, 2012, , 104-119.	1.0	4
32	Interval-Based Resource Usage Verification: Formalization and Prototype. Lecture Notes in Computer Science, 2012, , 54-71.	1.0	5
33	A Segment-Swapping Approach for Executing Trapped Computations. Lecture Notes in Computer Science, 2012, , 138-152.	1.0	0
34	Automated Attribute Inference in Complex Service Workflows Based on Sharing Analysis. , 2011, , .		4
35	Efficient local unfolding with ancestor stacks. Theory and Practice of Logic Programming, 2011, 11, 1-32.	1.1	4
36	Parallel backtracking with answer memoing for independent and-parallelism. Theory and Practice of Logic Programming, 2011, 11, 555-574.	1.1	1

#	Article	IF	Citations
37	CLP projection for constraint handling rules. , 2011, , .		2
38	Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations. Lecture Notes in Computer Science, 2011, , 62-76.	1.0	29
39	An Overview of the Ciao System. Lecture Notes in Computer Science, 2011, , 2-2.	1.0	1
40	Introduction to the 26th international conference on logic programming special issue. Theory and Practice of Logic Programming, 2010, 10, 361-364.	1.1	1
41	Automatic Inference of Determinacy and Mutual Exclusion for Logic Programs Using Mode and Type Analyses. New Generation Computing, 2010, 28, 177-206.	2.5	9
42	Lock-free parallel dynamic programming. Journal of Parallel and Distributed Computing, 2010, 70, 839-848.	2.7	33
43	Towards Data-Aware QoS-driven Adaptation for Service Orchestrations. , 2010, , .		26
44	Automatic Fragment Identification in Workflows Based on Sharing Analysis. Lecture Notes in Computer Science, 2010, , 350-364.	1.0	8
45	An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring. Lecture Notes in Computer Science, 2010, , 414-424.	1.0	5
46	Program Parallelization Using Synchronized Pipelining. Lecture Notes in Computer Science, 2010, , 173-187.	1.0	0
47	Non-strict independence-based program parallelization using sharing and freeness information. Theoretical Computer Science, 2009, 410, 4704-4723.	0.5	2
48	User-Definable Resource Usage Bounds Analysis for Java Bytecode. Electronic Notes in Theoretical Computer Science, 2009, 253, 65-82.	0.9	20
49	Integrating Software Testing and Run-Time Checking in an Assertion Verification Framework. Lecture Notes in Computer Science, 2009, , 281-295.	1.0	27
50	Identification of logically related heap regions. , 2009, , .		7
51	A Tabling Implementation Based on Variables with Multiple Bindings. Lecture Notes in Computer Science, 2009, , 190-204.	1.0	1
52	Abstraction-Carrying Code: a Model for Mobile Code Safety. New Generation Computing, 2008, 26, 171-204.	2.5	5
53	Comparing tag scheme variations using an abstract machine generator. , 2008, , .		4
54	A practical type analysis for verification of modular prolog programs. , 2008, , .		6

#	Article	IF	Citations
55	Sharing analysis of arrays, collections, and recursive structures. , 2008, , .		20
56	An Overview of the Ciao Multiparadigm Language and Program Development Environment and Its Design Philosophy. Lecture Notes in Computer Science, 2008, , 209-237.	1.0	12
57	An Improved Continuation Call-Based Implementation of Tabling. , 2008, , 197-213.		7
58	A Flexible, (C)LP-Based Approach to the Analysis of Object-Oriented Programs. Lecture Notes in Computer Science, 2008, , 154-168.	1.0	41
59	Efficient Context-Sensitive Shape Analysis with Graph Based Heap Models. , 2008, , 245-259.		14
60	Identification of Heapâ€“Carried Data Dependence Via Explicit Store Heap Models. Lecture Notes in Computer Science, 2008, , 94-108.	1.0	10
61	A High-Level Implementation of Non-deterministic, Unrestricted, Independent And-Parallelism. Lecture Notes in Computer Science, 2008, , 651-666.	1.0	5
62	Towards a Complete Scheme for Tabled Execution Based on Program Transformation. Lecture Notes in Computer Science, 2008, , 224-238.	1.0	3
63	A Sketch of a Complete Scheme for Tabled Execution Based on Program Transformation. Lecture Notes in Computer Science, 2008, , 795-800.	1.0	0
64	Annotation Algorithms for Unrestricted Independent And-Parallelism in Logic Programs. Lecture Notes in Computer Science, 2008, , 138-153.	1.0	4
65	Negative Ternary Set-Sharing. Lecture Notes in Computer Science, 2008, , 301-316.	1.0	1
66	Precise Set Sharing Analysis for Java-Style Programs. , 2008, , 172-187.		7
67	Heap analysis in the presence of collection libraries. , 2007, , .		12
68	User-Definable Resource Bounds Analysis for Logic Programs. , 2007, , 348-363.		30
69	An Efficient, Parametric Fixpoint Algorithm for Analysis of Java Bytecode. Electronic Notes in Theoretical Computer Science, 2007, 190, 51-66.	0.9	7
70	Automatic Binding-Related Error Diagnosis in Logic Programs. , 2007, , 333-347.		0
71	A Syntactic Approach to Combining Functional Notation, Lazy Evaluation, and Higher-Order in LP Systems. Lecture Notes in Computer Science, 2006, , 146-162.	1.0	21
72	Reduced Certificates for Abstraction-Carrying Code. Lecture Notes in Computer Science, 2006, , 163-178.	1.0	11

#	Article	IF	Citations
73	Using Combined Static Analysis and Profiling for Logic Program Execution Time Estimation. Lecture Notes in Computer Science, 2006, , 431-432.	1.0	3
74	Context-Sensitive Multivariant Assertion Checking in Modular Programs. Lecture Notes in Computer Science, 2006, , 392-406.	1.0	6
75	Combining Static Analysis and Profiling for Estimating Execution Times. Lecture Notes in Computer Science, 2006, , 140-154.	1.0	7
76	Towards Description and Optimization of Abstract Machines in an Extension of Prolog. , 2006, , 77-93.		1
77	Removing Superfluous Versions in Polyvariant Specialization of Prolog Programs. Lecture Notes in Computer Science, 2006, , 80-97.	1.0	2
78	Experiments in Context-Sensitive Analysis of Modular Programs. Lecture Notes in Computer Science, 2006, , 163-178.	1.0	4
79	Abstract Interpretation with Specialized Definitions. Lecture Notes in Computer Science, 2006, , 107-126.	1.0	11
80	An Abstract Interpretation-based Approach to Mobile Code Safety. Electronic Notes in Theoretical Computer Science, 2005, 132, 113-129.	0.9	5
81	Integrated program debugging, verification, and optimization using abstract interpretation (and the) Tj ETQq1 1 0.784314 rgBT /Over	1.5	118
82	Efficient Local Unfolding with Ancestor Stacks for Full Prolog. Lecture Notes in Computer Science, 2005, , 149-165.	1.0	11
83	Determinacy Analysis for Logic Programs Using Mode and Type Information. Lecture Notes in Computer Science, 2005, , 19-35.	1.0	9
84	Abstraction carrying code and resource-awareness. , 2005, , .		9
85	A Generator of Efficient Abstract Machine Implementations and Its Application to Emulator Minimization. Lecture Notes in Computer Science, 2005, , 21-36.	1.0	7
86	Efficient Top-Down Set-Sharing Analysis Using Cliques. Lecture Notes in Computer Science, 2005, , 183-198.	1.0	8
87	Abstraction-Carrying Code. Lecture Notes in Computer Science, 2005, , 380-397.	1.0	31
88	A Generic Framework for Context-Sensitive Analysis of Modular Programs. Lecture Notes in Computer Science, 2004, , 233-260.	1.0	6
89	Multivariant Non-failure Analysis via Standard Abstract Interpretation. Lecture Notes in Computer Science, 2004, , 100-116.	1.0	10
90	Improved Compilation of Prolog to C Using Moded Types and Determinism Information. Lecture Notes in Computer Science, 2004, , 86-103.	1.0	13

#	Article	IF	Citations
91	Hiord: A Type-Free Higher-Order Logic Programming Language with Predicate Abstraction. Lecture Notes in Computer Science, 2004, , 93-108.	1.0	3
92	Abstract Interpretation-Based Mobile Code Certification. Lecture Notes in Computer Science, 2004, , 446-447.	1.0	2
93	Abstract specialization and its applications. ACM SIGPLAN Notices, 2003, 38, 29-43.	0.2	5
94	Abstract Verification and Debugging of Constraint Logic Programs. Lecture Notes in Computer Science, 2003, , 1-14.	1.0	4
95	Program Development Using Abstract Interpretation (And the Ciao System Preprocessor). Lecture Notes in Computer Science, 2003, , 127-152.	1.0	18
96	Abstract specialization and its applications. , 2003, , .		7
97	A Generic Persistence Model for (C)LP Systems. Lecture Notes in Computer Science, 2003, , 481-482.	1.0	1
98	Special issue on â€˜Logic Programming and the INTERNETâ€™. Theory and Practice of Logic Programming, 2001, 1, 249-250.	1.1	0
99	Distributed WWW programming using (Ciao-)Prolog and the PiLLoW library. Theory and Practice of Logic Programming, 2001, 1, 251-282.	1.1	28
100	Parallel execution of prolog programs. ACM Transactions on Programming Languages and Systems, 2001, 23, 472-602.	1.7	103
101	A Model for Inter-module Analysis and Optimizing Compilation. Lecture Notes in Computer Science, 2001, , 86-102.	1.0	15
102	The Ciao Modular, Standalone Compiler and Its Generic Program Processing Library. Electronic Notes in Theoretical Computer Science, 2000, 30, 144-162.	0.9	6
103	The Ciao Module System: A New Module System for Prolog. Electronic Notes in Theoretical Computer Science, 2000, 30, 122-142.	0.9	5
104	Some Issues in Analysis and Specialization of Modular Ciao-Prolog Programs. Electronic Notes in Theoretical Computer Science, 2000, 30, 163-187.	0.9	12
105	A System for Automatically Generating Documentation for (C)LP Programs. Electronic Notes in Theoretical Computer Science, 2000, 30, 289-307.	0.9	1
106	Parallelizing irregular and pointer-based computations automatically: Perspectives from logic and constraint programming. Parallel Computing, 2000, 26, 1685-1708.	1.3	12
107	Independence in CLP languages. ACM Transactions on Programming Languages and Systems, 2000, 22, 296-339.	1.7	10
108	Incremental analysis of constraint logic programs. ACM Transactions on Programming Languages and Systems, 2000, 22, 187-223.	1.7	63

#	Article	IF	Citations
109	Combined Static and Dynamic Assertion-Based Debugging of Constraint Logic Programs. Lecture Notes in Computer Science, 2000, , 273-292.	1.0	35
110	Tools for Search-Tree Visualisation: The APT Tool. Lecture Notes in Computer Science, 2000, , 237-252.	1.0	7
111	Tools for Constraint Visualisation: The VIFID/TRIFID Tool. Lecture Notes in Computer Science, 2000, , 253-272.	1.0	7
112	An Assertion Language for Constraint Logic Programs. Lecture Notes in Computer Science, 2000, , 23-61.	1.0	54
113	A Generic Preprocessor for Program Validation and Debugging. Lecture Notes in Computer Science, 2000, , 63-107.	1.0	13
114	A New Module System for Prolog. Lecture Notes in Computer Science, 2000, , 131-148.	1.0	21
115	A Documentation Generator for (C)LP Systems. Lecture Notes in Computer Science, 2000, , 1345-1361.	1.0	9
116	Effectivness of abstract interpretation in automatic parallelization. ACM Transactions on Programming Languages and Systems, 1999, 21, 189-239.	1.7	33
117	Automatic compile-time parallelization of logic programs for restricted, goal level, independent and parallelism. The Journal of Logic Programming, 1999, 38, 165-218.	1.9	29
118	Abstract multiple specialization and its application to program parallelization. The Journal of Logic Programming, 1999, 41, 279-316.	1.9	26
119	Using Global Analysis, Partial Specifications, and an Extensible Assertion Language for Program Validation and Debugging. Artificial Intelligence, 1999, , 161-192.	0.7	35
120	Partial order and contextual net semantics for atomic and locally atomic CC programs. Science of Computer Programming, 1998, 30, 51-82.	1.5	16
121	A Framework for Assertion-based Debugging in Constraint Logic Programming. Lecture Notes in Computer Science, 1998, , 472-472.	1.0	7
122	Automatic parallelization of irregular and pointer-based computations: Perspectives from logic and constraint programming. Lecture Notes in Computer Science, 1997, , 31-45.	1.0	7
123	Some Challenges for Constraint Programming. Constraints, 1997, 2, 63-69.	0.4	2
124	Exploiting goal independence in the analysis of logic programs. The Journal of Logic Programming, 1997, 32, 247-261.	1.9	10
125	Abstract specialization and its application to program parallelization. Lecture Notes in Computer Science, 1997, , 169-186.	1.0	7
126	Flexible scheduling for non-deterministic, and-parallel execution of logic programs. Lecture Notes in Computer Science, 1996, , 635-639.	1.0	7

#	Article	IF	Citations
127	High-level characteristics of or- and independent and-parallelism in prolog. International Journal of Parallel Programming, 1996, 24, 433-478.	1.1	3
128	Improving the efficiency of nondeterministic independent and-parallel systems. Computer Languages, Systems and Structures, 1996, 22, 115-142.	0.3	15
129	Relating data-parallelism and (and-) parallelism in logic programs. Computer Languages, Systems and Structures, 1996, 22, 143-163.	0.3	7
130	A Methodology for Granularity-Based Control of Parallelism in Logic Programs. Journal of Symbolic Computation, 1996, 21, 715-734.	0.5	30
131	Optimized algorithms for incremental analysis of logic programs. Lecture Notes in Computer Science, 1996, , 270-284.	1.0	23
132	Towards independent and-parallelism in CLP. Lecture Notes in Computer Science, 1996, , 77-91.	1.0	10
133	Global analysis of constraint logic programs. ACM Transactions on Programming Languages and Systems, 1996, 18, 564-614.	1.7	44
134	Some challenges for constraint programming. ACM Computing Surveys, 1996, 28, 64.	16.1	1
135	Independence in dynamically scheduled logic languages. Lecture Notes in Computer Science, 1996, , 47-61.	1.0	0
136	Automatic optimization of dynamic scheduling in logic programs. Lecture Notes in Computer Science, 1996, , 475-476.	1.0	2
137	Relating data-parallelism and (and-) parallelism in logic programs. Lecture Notes in Computer Science, 1995, , 27-41.	1.0	4
138	Strict and nonstrict independent and-parallelism in logic programs: Correctness, efficiency, and compile-time conditions. The Journal of Logic Programming, 1995, 22, 1-45.	1.9	51
139	Special section: Ten Years of Logic Programming. The Journal of Logic Programming, 1995, 23, 87-88.	1.9	0
140	Improving abstract interpretations by combining domains. ACM Transactions on Programming Languages and Systems, 1995, 17, 28-44.	1.7	44
141	Implementation of multiple specialization in logic programs. , 1995, , .		21
142	Analyzing logic programs with dynamic scheduling. , 1994, , .		19
143	Extracting Non-strict independent and-parallelism using sharing and freeness information. Lecture Notes in Computer Science, 1994, , 297-313.	1.0	8
144	Analytic model of a Cache Only Memory Architecture. Lecture Notes in Computer Science, 1994, , 336-350.	1.0	0

#	Article	IF	Citations
145	And-Or parallel Prolog: A recomputation based approach. New Generation Computing, 1993, 11, 297-321.	2.5	9
146	Improving abstract interpretations by combining domains. , 1993, , .		19
147	Compile-time derivation of variable dependency using abstract interpretation. The Journal of Logic Programming, 1992, 13, 315-347.	1.9	149
148	Global flow analysis as a practical compilation tool. The Journal of Logic Programming, 1992, 13, 349-366.	1.9	91
149	The &-Prolog system: Exploiting independent and-parallelism. New Generation Computing, 1991, 9, 233-256.	2.5	58
150	Task granularity analysis in logic programs. ACM SIGPLAN Notices, 1990, 25, 174-188.	0.2	18
151	Memory referencing characteristics and caching performance of AND-Parallel Prolog on shared-memory multiprocessors. New Generation Computing, 1989, 7, 37-58.	2.5	3
152	Constructs and evaluations strategies for intelligent speculative parallelism---armageddon revisited. , 1988, , .		0
153	Designing a high performance parallel logic programming system. Computer Architecture News, 1987, 15, 43-52.	2.5	8
154	&ACE: a high-performance parallel Prolog system. , 0, , .		22
155	Regular Path Clauses and Their Application in Solving Loops. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 344, 22-35.	0.8	0
156	From Big-Step to Small-Step Semantics and Back with Interpreter Specialisation. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 320, 50-64.	0.8	6
157	Energy Consumption Analysis and Verification by Transformation into Horn Clauses and Abstract Interpretation. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 253, 4-6.	0.8	0
158	Parallel Logic Programming: A Sequel. Theory and Practice of Logic Programming, 0, , 1-69.	1.1	3