Yanhong A Liu

List of Publications by Year
in descending order

44 papers	650 citations	840776 11 h-index	713466 21 g-index
48 all docs	48 docs citations	48 times ranked	170 citing authors

#	Article	IF	Citations
1	Static caching for incremental computation. ACM Transactions on Programming Languages and Systems, 1998, 20, 546-585.	2.1	74
2	Systematic derivation of incremental programs. Science of Computer Programming, 1995, 24, 1-39.	1.9	64
3	Model Checking Linearizability via Refinement. Lecture Notes in Computer Science, 2009, , 321-337.	1.3	43
4	From recursion to iteration. , 1999, , .		39
5	From datalog rules to efficient programs with time and space guarantees. ACM Transactions on Programming Languages and Systems, 2009, 31, 1-38.	2.1	34
6	Efficiency by Incrementalization: An Introduction. Higher-Order and Symbolic Computation, 2000, 13, 289-313.	0.3	33
7	Incrementalization across object abstraction. , 2005, , .		30
8	From clarity to efficiency for distributed algorithms. , 2012, , .		25
9	Dynamic Programming via Static Incrementalization. Higher-Order and Symbolic Computation, 2003, 16, 37-62.	0.3	24
10	Alias analysis for optimization of dynamic languages. , 2010, , .		23
11	Optimizing aggregate array computations in loops. ACM Transactions on Programming Languages and Systems, 2005, 27, 91-125.	2.1	22
12	Automatic accurate time-bound analysis for high-level languages. Lecture Notes in Computer Science, 1998, , 31-40.	1.3	21
13	Verifying Linearizability via Optimized Refinement Checking. IEEE Transactions on Software Engineering, 2013, 39, 1018-1039.	5.6	19
14	From recursion to iteration. ACM SIGPLAN Notices, 1999, 34, 73-82.	0.2	18
15	More efficient datalog queries. , 2011, , .		18
16	Caching intermediate results for program improvement. , 1995, , .		17
17	Discovering auxiliary information for incremental computation. , 1996, , .		17
18	Eliminating dead code on recursive data. Science of Computer Programming, 2003, 47, 221-242.	1.9	14

#	Article	IF	Citations
19	Principled strength reduction. IFIP Advances in Information and Communication Technology, 1997, , 357-381.	0.7	12
20	Strengthening invariants for efficient computation. Science of Computer Programming, 2001, 41, 139-172.	1.9	11
21	Founded Semantics and Constraint Semantics of Logic Rules. Lecture Notes in Computer Science, 2018, , 221-241.	1.3	11
22	A language and framework for invariant-driven transformations. , 2009, , .		10
23	Founded semantics and constraint semantics of logic rules. Journal of Logic and Computation, 2020, 30, 1609-1668.	0.8	9
24	Precise complexity analysis for efficient datalog queries. , 2010, , .		9
25	Eliminating Dead Code on Recursive Data. Lecture Notes in Computer Science, 1999, , 211-231.	1.3	7
26	Alias analysis for optimization of dynamic languages. ACM SIGPLAN Notices, 2010, 45, 27-42.	0.2	5
27	Knowledge of Uncertain Worlds: Programming with Logical Constraints. Lecture Notes in Computer Science, 2020, , 111-127.	1.3	5
28	A systematic incrementalization technique and its application to hardware design. International Journal on Software Tools for Technology Transfer, 2003, 4, 211-223.	1.9	4
29	Logic programming applications: what are the abstractions and implementations?. , 2018, , 519-548.		4
30	Moderately Complex Paxos Made Simple. , 2019, , .		4
31	Dynamic Programming via Static Incrementalization. Lecture Notes in Computer Science, 1999, , 288-305.	1.3	4
32	Automatic Accurate Live Memory Analysis for Garbage-Collected Languages. ACM SIGPLAN Notices, 2001, 36, 102-111.	0.2	3
33	Analysis and Transformations for Efficient Query-Based Debugging. , 2008, , .		3
34	Knowledge of uncertain worlds: programming with logical constraints. Journal of Logic and Computation, 2021, 31, 193-212.	0.8	3
35	Optimizing Ackermann's function by incrementalization. ACM SIGPLAN Notices, 2003, 38, 85-91.	0.2	2
36	Precise complexity guarantees for pointer analysis via Datalog with extensions. Theory and Practice of Logic Programming, 2016, 16, 916-932.	1.5	2

#	Article	IF	Citations
37	Dynamic Programming via Static Incrementalization. , 2008, , 71-92.		2
38	Removing runtime overhead for optimized object queries. , 2016, , .		1
39	Solving Regular Tree Grammar Based Constraints. Lecture Notes in Computer Science, 2001, , 213-233.	1.3	1
40	Program optimization using indexed and recursive data structures. ACM SIGPLAN Notices, 2002, 37, 108-118.	0.2	1
41	A Systematic Incrementalization Technique and Its Application to Hardware Design. Lecture Notes in Computer Science, 1999, , 334-337.	1.3	1
42	Incrementalization across object abstraction. ACM SIGPLAN Notices, 2005, 40, 473-486.	0.2	0
43	Discrete Math with Programming. , 2021, , .		0
44	Automatic time-bound analysis for a higher-order language. ACM SIGPLAN Notices, 2002, 37, 75-86.	0.2	0