

# Mary Lou Soffa

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/10410614/publications.pdf>

Version: 2024-02-01

62  
papers

2,451  
citations

567281

15  
h-index

345221

36  
g-index

62  
all docs

62  
docs citations

62  
times ranked

1039  
citing authors

#	ARTICLE	IF	CITATIONS
1	A methodology for controlling the size of a test suite. ACM Transactions on Software Engineering and Methodology, 1993, 2, 270-285.	6.0	539
2	Bubble-Up. , 2011, , .		450
3	Coverage criteria for GUI testing. , 2001, , .		119
4	Efficient computation of interprocedural definition-use chains. ACM Transactions on Programming Languages and Systems, 1994, 16, 175-204.	2.1	106
5	The impact of memory subsystem resource sharing on datacenter applications. , 2011, , .		103
6	An approach for exploring code improving transformations. ACM Transactions on Programming Languages and Systems, 1997, 19, 1053-1084.	2.1	100
7	Contention aware execution. , 2010, , .		82
8	Automated test oracles for GUIs. , 2000, , .		77
9	Program Slicing-Based Regression Testing Techniques. Software Testing Verification and Reliability, 1996, 6, 83-111.	2.0	74
10	Feedback vertex sets and cyclically reducible graphs. Journal of the ACM, 1985, 32, 296-313.	2.2	71
11	Using a goal-driven approach to generate test cases for GUIs. , 1999, , .		66
12	Compiling for niceness. , 2012, , .		55
13	Coverage criteria for GUI testing. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2001, 26, 256-267.	0.7	38
14	Automated test oracles for GUIs. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2000, 25, 30-39.	0.7	36
15	Regression testing of GUIs. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2003, 28, 118-127.	0.7	34
16	On locating minimum feedback vertex sets. Journal of Computer and System Sciences, 1988, 37, 292-311.	1.2	30
17	Employing static information in the generation of test cases. Software Testing Verification and Reliability, 1993, 3, 29-48.	2.0	29
18	Predicting the memory bandwidth and optimal core allocations for multi-threaded applications on large-scale NUMA machines. , 2016, , .		28

#	ARTICLE	IF	CITATIONS
19	Refining data flow information using infeasible paths. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1997, 22, 361-377.	0.7	27
20	Performance analysis of thread mappings with a holistic view of the hardware resources. , 2012, , .		27
21	Automated test data generation using an iterative relaxation method. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1998, 23, 231-244.	0.7	26
22	Characterizing multi-threaded applications based on shared-resource contention. , 2011, , .		24
23	DraMon: Predicting memory bandwidth usage of multi-threaded programs with high accuracy and low overhead. , 2014, , .		24
24	Complete removal of redundant expressions. ACM SIGPLAN Notices, 1998, 33, 1-14.	0.2	22
25	Refining data flow information using infeasible paths. Lecture Notes in Computer Science, 1997, , 361-377.	1.3	20
26	Interprocedural data flow testing. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1989, 14, 158-167.	0.7	17
27	Incremental global reoptimization of programs. ACM Transactions on Programming Languages and Systems, 1992, 14, 173-200.	2.1	16
28	A family of test adequacy criteria for database-driven applications. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2003, 28, 98-107.	0.7	16
29	Low overhead program monitoring and profiling. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2006, 31, 28-34.	0.7	14
30	Distribution-Aware Testing of Neural Networks Using Generative Models. , 2021, , .		14
31	Catching and Identifying Bugs in Register Allocation. Lecture Notes in Computer Science, 2006, , 281-300.	1.3	14
32	Compilation techniques for a reconfigurable LIW architecture. Journal of Supercomputing, 1989, 3, 271-304.	3.6	11
33	Debugging and Testing Optimizers through Comparison Checking. Electronic Notes in Theoretical Computer Science, 2002, 65, 83-99.	0.9	11
34	Load-reuse analysis. ACM SIGPLAN Notices, 1999, 34, 64-76.	0.2	11
35	Referencing and Retention in Block-Structured Coroutines. ACM Transactions on Programming Languages and Systems, 1981, 3, 263-292.	2.1	10
36	Incremental register reallocation. Software - Practice and Experience, 1990, 20, 1015-1047.	3.6	10

#	ARTICLE	IF	CITATIONS
37	Compile-Time Planning for Overhead Reduction in Software Dynamic Translators. International Journal of Parallel Programming, 2005, 33, 103-114.	1.5	10
38	SIMCAL. , 1986, , .		9
39	Hybrid slicing. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1995, 20, 29-40.	0.7	9
40	Conversion of simulation processes to pascal constructs. Software - Practice and Experience, 1990, 20, 191-207.	3.6	6
41	Interprocedural conditional branch elimination. ACM SIGPLAN Notices, 1997, 32, 146-158.	0.2	6
42	An approach toward profit-driven optimization. Transactions on Architecture and Code Optimization, 2006, 3, 231-262.	2.0	6
43	Automatic generation of global optimizers. ACM SIGPLAN Notices, 1991, 26, 120-129.	0.2	6
44	The implementation of retention in a coroutine environment. Acta Informatica, 1983, 19, 221.	0.5	5
45	Compile-time techniques for efficient utilization of parallel memories. , 1988, , .		5
46	The design and implementation of genesis. Software - Practice and Experience, 1994, 24, 307-325.	3.6	5
47	GURRR. ACM SIGPLAN Notices, 1995, 30, 23-34.	0.2	4
48	The contraction of control implementations. Computer Languages, Systems and Structures, 1983, 8, 15-25.	0.3	3
49	Parallel generational garbage collection. ACM SIGPLAN Notices, 1991, 26, 16-32.	0.2	3
50	A shape matching approach for scheduling fine-grained parallelism. ACM SIGMICRO Newsletter, 1992, 23, 264-267.	0.4	3
51	Value prediction in VLIW machines. Computer Architecture News, 1999, 27, 258-269.	2.5	3
52	Virtual Execution Environments: Support and Tools. , 2007, , .		3
53	ReQoS. ACM SIGPLAN Notices, 2013, 48, 89-100.	0.2	3
54	Reverse execution in a generalized control regime. Computer Languages, Systems and Structures, 1984, 9, 183-192.	0.3	2

#	ARTICLE	IF	CITATIONS
55	Compile-time techniques for efficient utilization of parallel memories. ACM SIGPLAN Notices, 1988, 23, 235-246.	0.2	2
56	Debugging parallelized code using code liberation techniques. ACM SIGPLAN Notices, 1991, 26, 108-119.	0.2	2
57	The efficiency of storage management schemes for Ada programs. ACM SIGPLAN Notices, 1985, 20, 30-38.	0.2	2
58	An optimistic implementation of the stack-heap. Journal of Systems and Software, 1985, 5, 193-202.	4.5	1
59	DBT path selection for holistic memory efficiency and performance. ACM SIGPLAN Notices, 2010, 45, 145-156.	0.2	1
60	The efficiency of storage management schemes for Ada programs. ACM SIGAda Ada Letters, 1985, V, 164-172.	0.1	1
61	Control discipline necessity: Making the language as general as the implementation. BIT Numerical Mathematics, 1982, 22, 169-182.	2.0	0
62	VISUAL TRANSFORMATION SPECIFICATIONS. Series on Software Engineering and Knowledge Engineering, 1996, , 141-162.	0.1	0