Thomas Reps

List of Publications by Year in Descending Order

106 papers	5,917 citations	33 h-index	76 g-index
108 ext. papers	6,520 ext. citations	0.9 avg, IF	5.63 L-index

#	Paper	IF	Citations
106	On the complexity of bidirected interleaved Dyck-reachability **2021**, 5, 1-28		4
105	Interprocedural Context-Unbounded Program Analysis Using Observation Sequences. *ACM Transactions on Programming Languages and Systems*, **2021**, 42, 1-34	1.6	
104	Algebraic Program Analysis. *Lecture Notes in Computer Science*, **2021**, 46-83	0.9	1
103	Fast graph simplification for interleaved Dyck-reachability **2020**,		11
102	Refinement of path expressions for static analysis **2019**, 3, 1-29		10
101	A new abstraction framework for affine transformers. *Formal Methods in System Design*, **2019**, 54, 110-143		2
100	PMAF: an algebraic framework for static analysis of probabilistic programs. *ACM SIGPLAN Notices*, **2018**, 53, 513-528	0.2	6
99	Compositional recurrence analysis revisited **2017**,		22
98	Newtonian Program Analysis via Tensor Product. *ACM Transactions on Programming Languages and Systems*, **2017**, 39, 1-72	1.6	4
97	Compositional recurrence analysis revisited. *ACM SIGPLAN Notices*, **2017**, 52, 248-262	0.2	8
96	Sound Bit-Precise Numerical Domains. *Lecture Notes in Computer Science*, **2017**, 500-520	0.9	2
95	Automating Abstract Interpretation. *Lecture Notes in Computer Science*, **2016**, 3-40	0.9	11
94	Newtonian program analysis via tensor product **2016**,		10
93	Newtonian program analysis via tensor product. *ACM SIGPLAN Notices*, **2016**, 51, 663-677	0.2	2
92	Synthesis of machine code from semantics **2015**,		19
91	Partial evaluation of machine code **2015**,		8
90	Partial evaluation of machine code. *ACM SIGPLAN Notices*, **2015**, 50, 860-879	0.2	

89	Specialization Slicing. *ACM Transactions on Programming Languages and Systems*, **2014**, 36, 1-67	1.6	7
88	Abstract Domains of Affine Relations. *ACM Transactions on Programming Languages and Systems*, **2014**, 36, 1-73	1.6	13
87	TSL. *ACM Transactions on Programming Languages and Systems*, **2013**, 35, 1-59	1.6	28
86	OpenNWA: A Nested-Word Automaton Library. *Lecture Notes in Computer Science*, **2012**, 665-671	0.9	10
85	Bilateral Algorithms for Symbolic Abstraction. *Lecture Notes in Computer Science*, **2012**, 111-128	0.9	24
84	ConSeq. *Computer Architecture News*, **2011**, 39, 251-264		5
83	Symbolic analysis via semantic reinterpretation. *International Journal on Software Tools for Technology Transfer*, **2011**, 13, 61-87	1.3	9
82	A decision procedure for detecting atomicity violations for communicating processes with locks. *International Journal on Software Tools for Technology Transfer*, **2011**, 13, 37-60	1.3	2
81	Finding concurrency-related bugs using random isolation. *International Journal on Software Tools for Technology Transfer*, **2011**, 13, 495-518	1.3	9
80	ConSeq. *ACM SIGPLAN Notices*, **2011**, 46, 251-264	0.2	10
79	Abstract Domains of Affine Relations. *Lecture Notes in Computer Science*, **2011**, 198-215	0.9	6
78	Analysis Techniques for Information Security. *Synthesis Lectures on Information Security Privacy and Trust*, **2010**, 2, 1-164	1	0
77	A relational approach to interprocedural shape analysis. *ACM Transactions on Programming Languages and Systems*, **2010**, 32, 1-52	1.6	19
76	WYSINWYX. *ACM Transactions on Programming Languages and Systems*, **2010**, 32, 1-84	1.6	118
75	View-Augmented Abstractions. *Electronic Notes in Theoretical Computer Science*, **2010**, 267, 43-57	0.7	3
74	Reducing concurrent analysis under a context bound to sequential analysis. *Formal Methods in System Design*, **2009**, 35, 73-97	1.4	105
73	**2009**,		2
72	A Decision Procedure for Detecting Atomicity Violations for Communicating Processes with Locks. *Lecture Notes in Computer Science*, **2009**, 125-142	0.9	9

71	Verifying Information Flow Control over Unbounded Processes. *Lecture Notes in Computer Science*, **2009**, 773-789	0.9	11
70	Language Strength Reduction. *Lecture Notes in Computer Science*, **2008**, 283-298	0.9	10
69	Solving Multiple Dataflow Queries Using WPDSs. *Lecture Notes in Computer Science*, **2008**, 93-109	0.9	5
68	Interprocedural Analysis of Concurrent Programs Under a Context Bound **2008**, 282-298		37
67	Finding Concurrency-Related Bugs Using Random Isolation. *Lecture Notes in Computer Science*, **2008** , 198-213	0.9	12
66	A Next-Generation Platform for Analyzing Executables **2007**, 43-61		2
65	Low-Level Library Analysis and Summarization **2007**, 68-81		16
64	Abstract Error Projection. *Lecture Notes in Computer Science*, **2007**, 200-217	0.9	10
63	Program Analysis Using Weighted Pushdown Systems. *Lecture Notes in Computer Science*, **2007**, 23-51	0.9	13
62	Extracting Output Formats from Executables **2006**,		28
61	Intermediate-representation recovery from low-level code **2006**,		33
60	Weighted Pushdown Systems and Trust-Management Systems. *Lecture Notes in Computer Science*, **2006**, 1-26	0.9	7
59	Improving Pushdown System Model Checking. *Lecture Notes in Computer Science*, **2006**, 343-357	0.9	26
58	Recency-Abstraction for Heap-Allocated Storage. *Lecture Notes in Computer Science*, **2006**, 221-239	0.9	69
57	Analysis of recursive state machines. *ACM Transactions on Programming Languages and Systems*, **2005**, 27, 786-818	1.6	135
56	CodeSurfer/x86 Platform for Analyzing x86 Executables. *Lecture Notes in Computer Science*, **2005**, 250-254	0.9	75
55	Extended Weighted Pushdown Systems. *Lecture Notes in Computer Science*, **2005**, 434-448	0.9	41
54	Weighted pushdown systems and their application to interprocedural dataflow analysis. *Science of Computer Programming*, **2005**, 58, 206-263	1.1	132

53	Automatic Assume/Guarantee Reasoning for Heap-Manipulating Programs: Ongoing Work. *Electronic Notes in Theoretical Computer Science*, **2005**, 131, 125-138	0.7	4
52	A semantics for procedure local heaps and its abstractions. *ACM SIGPLAN Notices*, **2005**, 40, 296-309	0.2	10
51	A framework for numeric analysis of array operations. *ACM SIGPLAN Notices*, **2005**, 40, 338-350	0.2	36
50	Symbolic Implementation of the Best Transformer. *Lecture Notes in Computer Science*, **2004**, 252-266	0.9	84
49	Analyzing Memory Accesses in x86 Executables. *Lecture Notes in Computer Science*, **2004**, 5-23	0.9	136
48	Interprocedural slicing using dependence graphs. *ACM SIGPLAN Notices*, **2004**, 39, 229-243	0.2	18
47	Numeric Domains with Summarized Dimensions. *Lecture Notes in Computer Science*, **2004**, 512-529	0.9	42
46	Symbolically Computing Most-Precise Abstract Operations for Shape Analysis. *Lecture Notes in Computer Science*, **2004**, 530-545	0.9	61
45	A Relational Approach to Interprocedural Shape Analysis. *Lecture Notes in Computer Science*, **2004**, 246-264	0.9	24
44	Finite Differencing of Logical Formulas for Static Analysis. *Lecture Notes in Computer Science*, **2003**, 380-398	0.9	31
43	Parametric shape analysis via 3-valued logic. *ACM Transactions on Programming Languages and Systems*, **2002**, 24, 217-298	1.6	479
42	Model Checking of Unrestricted Hierarchical State Machines. *Lecture Notes in Computer Science*, **2001**, 652-666	0.9	26
41	Safety checking of machine code. *ACM SIGPLAN Notices*, **2000**, 35, 70-82	0.2	6
40	Interconvertibility of a class of set constraints and context-free-language reachability. *Theoretical Computer Science*, **2000**, 248, 29-98	1.1	67
39	Undecidability of context-sensitive data-dependence analysis. *ACM Transactions on Programming Languages and Systems*, **2000**, 22, 162-186	1.6	61
38	Putting static analysis to work for verification **2000**,		60
37	Physical type checking for C. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1999**, 24, 66-75	0.4	9
36	Coping with type casts in C. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1999**, 24, 180-198	0.4	6

35	Pointer analysis for programs with structures and casting. *ACM SIGPLAN Notices*, **1999**, 34, 91-103	0.2	12
34	Program analysis via graph reachability. *Information and Software Technology*, **1998**, 40, 701-726	3.4	168
33	Solving shape-analysis problems in languages with destructive updating. *ACM Transactions on Programming Languages and Systems*, **1998**, 20, 1-50	1.6	203
32	Interconvertbility of set constraints and context-free language reachability. *ACM SIGPLAN Notices*, **1997**, 32, 74-89	0.2	7
31	Program generalization for software reuse **1996**,		14
30	On the sequential nature of interprocedural program-analysis problems. *Acta Informatica*, **1996**, 33, 739-757	0.7	14
29	On the sequential nature of interprocedural program-analysis problems. *Acta Informatica*, **1996**, 33, 739-757	0.7	12
28	Precise interprocedural dataflow analysis with applications to constant propagation. *Theoretical Computer Science*, **1996**, 167, 131-170	1.1	159
27	Program specialization via program slicing. *Lecture Notes in Computer Science*, **1996**, 409-429	0.9	32
26	Demand interprocedural dataflow analysis **1995**,		97
25	Shape analysis as a generalized path problem **1995**,		41
24	Precise interprocedural chopping. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1995**, 20, 41-52	0.4	14
23	Demand interprocedural dataflow analysis. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1995**, 20, 104-115	0.4	20
22	Semantic foundations of binding-time analysis for imperative programs **1995**,		8
21	Precise interprocedural dataflow analysis via graph reachability **1995**,		642
20	Speeding up slicing. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1994**, 19, 11-20	0.4	35
19	Speeding up slicing **1994**,		88
18	On competitive on-line algorithms for the dynamic priority-ordering problem. *Information Processing Letters*, **1994**, 51, 155-161	0.8	13

17	Solving demand versions of interprocedural analysis problems. *Lecture Notes in Computer Science*, **1994**, 389-403	0.9	28
16	A program integration algorithm that accommodates semantics-preserving transformations. *ACM Transactions on Software Engineering and Methodology*, **1992**, 1, 310-354	3.3	33
15	Algebraic properties of program integration. *Science of Computer Programming*, **1991**, 17, 139-215	1.1	17
14	Efficient comparison of program slices. *Acta Informatica*, **1991**, 28, 713-732	0.9	11
13	Interprocedural slicing using dependence graphs. *ACM Transactions on Programming Languages and Systems*, **1990**, 12, 26-60	1.6	851
12	Integrating noninterfering versions of programs. *ACM Transactions on Programming Languages and Systems*, **1989**, 11, 345-387	1.6	218
11	Incremental evaluation for attribute grammars with unrestricted movement between tree modifications. *Acta Informatica*, **1988**, 25, 155-178	0.9	
10	Sublinear-space evaluation algorithms for attribute grammars. *ACM Transactions on Programming Languages and Systems*, **1987**, 9, 408-440	1.6	2
9	The synthesizer generator **1984**,		105
8	The synthesizer generator. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **1984**, 9, 42-48	0.4	17
7	The synthesizer generator. *ACM SIGPLAN Notices*, **1984**, 19, 42-48	0.2	40
6	Incremental Context-Dependent Analysis for Language-Based Editors. *ACM Transactions on Programming Languages and Systems*, **1983**, 5, 449-477	1.6	145
5	Optimal-time incremental semantic analysis for syntax-directed editors **1982**,		39
4	The why and wherefore of the Cornell Program Synthesizer. *ACM SIGPLAN Notices*, **1981**, 16, 8-16	0.2	24
3	The why and wherefore of the Cornell Program Synthesizer. *ACM SIGOA Newsletter*, **1981**, 2, 8-16		1
2	Incremental evaluation for attribute grammars with application to syntax-directed editors **1981**,		99
1	The Cornell program synthesizer. *Communications of the ACM*, **1981**, 24, 563-573	2.5	386