Sumit Gulwani

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/12187456/publications.pdf

Version: 2024-02-01

69

papers

4,901

citations

279798

23

h-index

361022

35

g-index

69

all docs

69

docs citations

69

times ranked

1223

citing authors

#	Article	IF	Citations
1	Semantic programming by example with pre-trained models. , 2021, 5, 1-25.		13
2	FlashProfile: a framework for synthesizing data profiles. , 2018, 2, 1-28.		14
3	Automated clustering and program repair for introductory programming assignments. ACM SIGPLAN Notices, 2018, 53, 465-480.	0.2	36
4	Learning Syntactic Program Transformations from Examples. , 2017, , .		135
5	Program Synthesis. Foundations and Trends in Programming Languages, 2017, 4, 1-119.	1.8	213
6	Programming by Examples: PL Meets ML. Lecture Notes in Computer Science, 2017, , 3-20.	1.3	23
7	Semi-supervised verified feedback generation. , 2016, , .		44
8	Transforming spreadsheet data types using examples. , 2016, , .		29
9	FIDEX: filtering spreadsheet data using examples. ACM SIGPLAN Notices, 2016, 51, 195-213.	0.2	7
10	FlashRelate: extracting relational data from semi-structured spreadsheets using examples. , 2015, , .		54
11	How Can Automatic Feedback Help Students Construct Automata?. ACM Transactions on Computer-Human Interaction, 2015, 22, 1-24.	5.7	28
12	Inductive programming meets the real world. Communications of the ACM, 2015, 58, 90-99.	4.5	78
13	User Interaction Models for Disambiguation in Programming by Example. , 2015, , .		57
14	FlashMeta: a framework for inductive program synthesis. , 2015, , .		119
15	Example-based learning in computer-aided STEM education. Communications of the ACM, 2014, 57, 70-80.	4.5	26
16	FlashExtract. , 2014, , .		116
17	Test-driven synthesis. ACM SIGPLAN Notices, 2014, 49, 408-418.	0.2	20
18	Test-driven synthesis. , 2014, , .		41

#	Article	IF	Citations
19	Feedback generation for performance problems in introductory programming assignments. , 2014, , .		47
20	Template-based program verification and program synthesis. International Journal on Software Tools for Technology Transfer, 2013, 15, 497-518.	1.9	58
21	Automated feedback generation for introductory programming assignments. , 2013, , .		198
22	Recursive Program Synthesis. Lecture Notes in Computer Science, 2013, , 934-950.	1.3	64
23	Type-directed completion of partial expressions. , 2012, , .		61
24	Learning semantic string transformations from examples. Proceedings of the VLDB Endowment, 2012, 5, 740-751.	3.8	93
25	Type-directed completion of partial expressions. ACM SIGPLAN Notices, 2012, 47, 275-286.	0.2	15
26	Synthesis from Examples: Interaction Models and Algorithms. , 2012, , .		53
27	Spreadsheet data manipulation using examples. Communications of the ACM, 2012, 55, 97-105.	4.5	197
28	Synthesizing Number Transformations from Input-Output Examples. Lecture Notes in Computer Science, 2012, , 634-651.	1.3	74
29	Automating string processing in spreadsheets using input-output examples. ACM SIGPLAN Notices, 2011, 46, 317-330.	0.2	209
30	Synthesizing switching logic using constraint solving. International Journal on Software Tools for Technology Transfer, 2011, 13, 519-535.	1.9	17
31	Synthesizing geometry constructions. , 2011, , .		59
32	Synthesis of loop-free programs. , 2011, , .		179
33	Synthesis of loop-free programs. ACM SIGPLAN Notices, 2011, 46, 62-73.	0.2	68
34	Automating string processing in spreadsheets using input-output examples. , 2011, , .		363
35	Synthesizing geometry constructions. ACM SIGPLAN Notices, 2011, 46, 50-61.	0.2	30
36	Spreadsheet table transformations from examples. , 2011, , .		93

#	Article	IF	Citations
37	Path-based inductive synthesis for program inversion. , 2011, , .		36
38	Bound Analysis of Imperative Programs with the Size-Change Abstraction. Lecture Notes in Computer Science, 2011, , 280-297.	1.3	63
39	Spreadsheet table transformations from examples. ACM SIGPLAN Notices, 2011, 46, 317-328.	0.2	41
40	A simple inductive synthesis methodology and its applications. ACM SIGPLAN Notices, 2010, 45, 36-46.	0.2	14
41	A simple inductive synthesis methodology and its applications. , 2010, , .		27
42	From program verification to program synthesis. , 2010, , .		166
43	Oracle-guided component-based program synthesis. , 2010, , .		338
44	Synthesizing switching logic for safety and dwell-time requirements. , 2010, , .		37
45	Dimensions in program synthesis. , 2010, , .		154
46	A combination framework for tracking partition sizes. , 2009, , .		22
47	SPEED. , 2009, , .		132
48	Program verification using templates over predicate abstraction. , 2009, , .		76
49	Control-flow refinement and progress invariants for bound analysis. ACM SIGPLAN Notices, 2009, 44, 375-385.	0.2	30
50	SPEED. ACM SIGPLAN Notices, 2009, 44, 127-139.	0.2	96
51	VS3: SMT Solvers for Program Verification. Lecture Notes in Computer Science, 2009, , 702-708.	1.3	21
52	SPEED: Symbolic Complexity Bound Analysis. Lecture Notes in Computer Science, 2009, , 51-62.	1.3	45
53	Program analysis as constraint solving. , 2008, , .		123
54	Inferring locks for atomic sections. , 2008, , .		100

#	Article	IF	Citations
55	Lifting abstract interpreters to quantified logical domains. ACM SIGPLAN Notices, 2008, 43, 235-246.	0.2	29
56	Inferring locks for atomic sections. ACM SIGPLAN Notices, 2008, 43, 304-315.	0.2	14
57	Constraint-Based Approach for Analysis of Hybrid Systems. Lecture Notes in Computer Science, 2008, , 190-203.	1.3	86
58	A Numerical Abstract Domain Based on Expression Abstraction and Max Operator with Application in Timing Analysis. Lecture Notes in Computer Science, 2008, , 370-384.	1.3	60
59	Constraint-Based Invariant Inference over Predicate Abstraction. Lecture Notes in Computer Science, 2008, , 120-135.	1.3	37
60	Synthesizing Switching Logic Using Constraint Solving. Lecture Notes in Computer Science, 2008, , 305-319.	1.3	22
61	Program verification as probabilistic inference. , 2007, , .		22
62	Combining abstract interpreters. ACM SIGPLAN Notices, 2006, 41, 376-386.	0.2	16
63	Combining abstract interpreters. , 2006, , .		49
64	Assertion Checking over Combined Abstraction of Linear Arithmetic and Uninterpreted Functions. Lecture Notes in Computer Science, 2006, , 279-293.	1.3	20
65	A randomized satisfiability procedure for arithmetic and uninterpreted function symbols. Information and Computation, 2005, 199, 107-131.	0.7	0
66	Global value numbering using random interpretation. , 2004, , .		15
67	A Polynomial-Time Algorithm for Global Value Numbering. Lecture Notes in Computer Science, 2004, , 212-227.	1.3	29
68	Join Algorithms for the Theory of Uninterpreted Functions. Lecture Notes in Computer Science, 2004, , 311-323.	1.3	14
69	Discovering affine equalities using random interpretation. , 2003, , .		36