

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 | Automating string processing in spreadsheets using input-output examples. , 2011, , . | | 363 |
| 2 | Oracle-guided component-based program synthesis. , 2010, , . | | 338 |
| 3 | Program Synthesis. Foundations and Trends in Programming Languages, 2017, 4, 1-119. | 1.8 | 213 |
| 4 | Automating string processing in spreadsheets using input-output examples. ACM SIGPLAN Notices, 2011, 46, 317-330. | 0.2 | 209 |
| 5 | Automated feedback generation for introductory programming assignments. , 2013, , . | | 198 |
| 6 | Spreadsheet data manipulation using examples. Communications of the ACM, 2012, 55, 97-105. | 4.5 | 197 |
| 7 | Synthesis of loop-free programs. , 2011, , . | | 179 |
| 8 | From program verification to program synthesis. , 2010, , . | | 166 |
| 9 | Dimensions in program synthesis. , 2010, , . | | 154 |
| 10 | Learning Syntactic Program Transformations from Examples. , 2017, , . | | 135 |
| 11 | SPEED. , 2009, , . | | 132 |
| 12 | Program analysis as constraint solving. , 2008, , . | | 123 |
| 13 | FlashMeta: a framework for inductive program synthesis. , 2015, , . | | 119 |
| 14 | FlashExtract. , 2014, , . | | 116 |
| 15 | Inferring locks for atomic sections. , 2008, , . | | 100 |
| 16 | SPEED. ACM SIGPLAN Notices, 2009, 44, 127-139. | 0.2 | 96 |
| 17 | Spreadsheet table transformations from examples. , 2011, , . | | 93 |
| 18 | Learning semantic string transformations from examples. Proceedings of the VLDB Endowment, 2012, 5, 740-751. | 3.8 | 93 |

| # | ARTICLE | IF | CITATIONS |
|----|---|-----|-----------|
| 19 | Constraint-Based Approach for Analysis of Hybrid Systems. Lecture Notes in Computer Science, 2008, , 190-203. | 1.3 | 86 |
| 20 | Inductive programming meets the real world. Communications of the ACM, 2015, 58, 90-99. | 4.5 | 78 |
| 21 | Program verification using templates over predicate abstraction. , 2009, , . | | 76 |
| 22 | Synthesizing Number Transformations from Input-Output Examples. Lecture Notes in Computer Science, 2012, , 634-651. | 1.3 | 74 |
| 23 | Synthesis of loop-free programs. ACM SIGPLAN Notices, 2011, 46, 62-73. | 0.2 | 68 |
| 24 | Recursive Program Synthesis. Lecture Notes in Computer Science, 2013, , 934-950. | 1.3 | 64 |
| 25 | Bound Analysis of Imperative Programs with the Size-Change Abstraction. Lecture Notes in Computer Science, 2011, , 280-297. | 1.3 | 63 |
| 26 | Type-directed completion of partial expressions. , 2012, , . | | 61 |
| 27 | 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 |
| 28 | Synthesizing geometry constructions. , 2011, , . | | 59 |
| 29 | Template-based program verification and program synthesis. International Journal on Software Tools for Technology Transfer, 2013, 15, 497-518. | 1.9 | 58 |
| 30 | User Interaction Models for Disambiguation in Programming by Example. , 2015, , . | | 57 |
| 31 | FlashRelate: extracting relational data from semi-structured spreadsheets using examples. , 2015, , . | | 54 |
| 32 | Synthesis from Examples: Interaction Models and Algorithms. , 2012, , . | | 53 |
| 33 | Combining abstract interpreters. , 2006, , . | | 49 |
| 34 | Feedback generation for performance problems in introductory programming assignments. , 2014, , . | | 47 |
| 35 | SPEED: Symbolic Complexity Bound Analysis. Lecture Notes in Computer Science, 2009, , 51-62. | 1.3 | 45 |
| 36 | Semi-supervised verified feedback generation. , 2016, , . | | 44 |

| # | ARTICLE | IF | CITATIONS |
|----|---|-----|-----------|
| 37 | Test-driven synthesis. , 2014, , . | | 41 |
| 38 | Spreadsheet table transformations from examples. ACM SIGPLAN Notices, 2011, 46, 317-328. | 0.2 | 41 |
| 39 | Constraint-Based Invariant Inference over Predicate Abstraction. Lecture Notes in Computer Science, 2008, , 120-135. | 1.3 | 37 |
| 40 | Synthesizing switching logic for safety and dwell-time requirements. , 2010, , . | | 37 |
| 41 | Path-based inductive synthesis for program inversion. , 2011, , . | | 36 |
| 42 | Automated clustering and program repair for introductory programming assignments. ACM SIGPLAN Notices, 2018, 53, 465-480. | 0.2 | 36 |
| 43 | Discovering affine equalities using random interpretation. , 2003, , . | | 36 |
| 44 | Control-flow refinement and progress invariants for bound analysis. ACM SIGPLAN Notices, 2009, 44, 375-385. | 0.2 | 30 |
| 45 | Synthesizing geometry constructions. ACM SIGPLAN Notices, 2011, 46, 50-61. | 0.2 | 30 |
| 46 | Lifting abstract interpreters to quantified logical domains. ACM SIGPLAN Notices, 2008, 43, 235-246. | 0.2 | 29 |
| 47 | A Polynomial-Time Algorithm for Global Value Numbering. Lecture Notes in Computer Science, 2004, , 212-227. | 1.3 | 29 |
| 48 | Transforming spreadsheet data types using examples. , 2016, , . | | 29 |
| 49 | How Can Automatic Feedback Help Students Construct Automata?. ACM Transactions on Computer-Human Interaction, 2015, 22, 1-24. | 5.7 | 28 |
| 50 | A simple inductive synthesis methodology and its applications. , 2010, , . | | 27 |
| 51 | Example-based learning in computer-aided STEM education. Communications of the ACM, 2014, 57, 70-80. | 4.5 | 26 |
| 52 | Programming by Examples: PL Meets ML. Lecture Notes in Computer Science, 2017, , 3-20. | 1.3 | 23 |
| 53 | Program verification as probabilistic inference. , 2007, , . | | 22 |
| 54 | A combination framework for tracking partition sizes. , 2009, , . | | 22 |

| # | ARTICLE | IF | CITATIONS |
|----|--|-----|-----------|
| 55 | Synthesizing Switching Logic Using Constraint Solving. Lecture Notes in Computer Science, 2008, , 305-319. | 1.3 | 22 |
| 56 | VS3: SMT Solvers for Program Verification. Lecture Notes in Computer Science, 2009, , 702-708. | 1.3 | 21 |
| 57 | Test-driven synthesis. ACM SIGPLAN Notices, 2014, 49, 408-418. | 0.2 | 20 |
| 58 | Assertion Checking over Combined Abstraction of Linear Arithmetic and Uninterpreted Functions. Lecture Notes in Computer Science, 2006, , 279-293. | 1.3 | 20 |
| 59 | Synthesizing switching logic using constraint solving. International Journal on Software Tools for Technology Transfer, 2011, 13, 519-535. | 1.9 | 17 |
| 60 | Combining abstract interpreters. ACM SIGPLAN Notices, 2006, 41, 376-386. | 0.2 | 16 |
| 61 | Global value numbering using random interpretation. , 2004, , . | | 15 |
| 62 | Type-directed completion of partial expressions. ACM SIGPLAN Notices, 2012, 47, 275-286. | 0.2 | 15 |
| 63 | Inferring locks for atomic sections. ACM SIGPLAN Notices, 2008, 43, 304-315. | 0.2 | 14 |
| 64 | A simple inductive synthesis methodology and its applications. ACM SIGPLAN Notices, 2010, 45, 36-46. | 0.2 | 14 |
| 65 | FlashProfile: a framework for synthesizing data profiles. , 2018, 2, 1-28. | | 14 |
| 66 | Join Algorithms for the Theory of Uninterpreted Functions. Lecture Notes in Computer Science, 2004, , 311-323. | 1.3 | 14 |
| 67 | Semantic programming by example with pre-trained models. , 2021, 5, 1-25. | | 13 |
| 68 | FIDEX: filtering spreadsheet data using examples. ACM SIGPLAN Notices, 2016, 51, 195-213. | 0.2 | 7 |
| 69 | A randomized satisfiability procedure for arithmetic and uninterpreted function symbols. Information and Computation, 2005, 199, 107-131. | 0.7 | 0 |