

# Wolfram Schulte

## List of Publications by Year in Descending Order

**Source:** <https://exaly.com/author-pdf/10651797/wolfram-schulte-publications-by-year.pdf>

**Version:** 2024-04-28

This document has been generated based on the publications and citations recorded by exaly.com. For the latest version of this publication list, visit the link given above.

The third column is the impact factor (IF) of the journal, and the fourth column is the number of citations of the article.

26  
papers

1,557  
citations

16  
h-index

26  
g-index

26  
ext. papers

1,696  
ext. citations

1  
avg, IF

4.2  
L-index

#	Paper	IF	Citations
26	Automated migration of build scripts using dynamic analysis and search-based refactoring. <i>ACM SIGPLAN Notices</i> , <b>2014</b> , 49, 599-616	0.2	7
25	Formalizing and Verifying a Modern Build Language. <i>Lecture Notes in Computer Science</i> , <b>2014</b> , 643-657	0.9	16
24	Specification and verification. <i>Communications of the ACM</i> , <b>2011</b> , 54, 81-91	2.5	133
23	Automatic verification of Java programs with dynamic frames. <i>Formal Aspects of Computing</i> , <b>2010</b> , 22, 423-457	1.2	7
22	HOL-Boogie: An Interactive Prover-Backend for the Verifying C Compiler. <i>Journal of Automated Reasoning</i> , <b>2010</b> , 44, 111-144	1	14
21	Local Verification of Global Invariants in Concurrent Programs. <i>Lecture Notes in Computer Science</i> , <b>2010</b> , 480-494	0.9	34
20	Reusing Model Transformations While Preserving Properties. <i>Lecture Notes in Computer Science</i> , <b>2010</b> , 44-58	0.9	3
19	Separation Logic Verification of C Programs with an SMT Solver. <i>Electronic Notes in Theoretical Computer Science</i> , <b>2009</b> , 254, 5-23	0.7	14
18	A Precise Yet Efficient Memory Model For C. <i>Electronic Notes in Theoretical Computer Science</i> , <b>2009</b> , 254, 85-103	0.7	35
17	Fitness-guided path exploration in dynamic symbolic execution <b>2009</b> ,		109
16	VCC: A Practical System for Verifying Concurrent C. <i>Lecture Notes in Computer Science</i> , <b>2009</b> , 23-42	0.9	285
15	A programming model for concurrent object-oriented programs. <i>ACM Transactions on Programming Languages and Systems</i> , <b>2008</b> , 31, 1-48	1.6	19
14	Automating Software Testing Using Program Analysis. <i>IEEE Software</i> , <b>2008</b> , 25, 30-37	1.5	70
13	An Automatic Verifier for Java-Like Programs Based on Dynamic Frames. <i>Lecture Notes in Computer Science</i> , <b>2008</b> , 261-275	0.9	17
12	A Simple Sequential Reasoning Approach for Sound Modular Verification of Mainstream Multithreaded Programs. <i>Electronic Notes in Theoretical Computer Science</i> , <b>2007</b> , 174, 23-47	0.7	7
11	Using History Invariants to Verify Observers. <i>Lecture Notes in Computer Science</i> , <b>2007</b> , 80-94	0.9	15
10	Allowing State Changes in Specifications. <i>Lecture Notes in Computer Science</i> , <b>2006</b> , 321-336	0.9	8

9	A Statically Verifiable Programming Model for Concurrent Object-Oriented Programs. <i>Lecture Notes in Computer Science</i> , <b>2006</b> , 420-439	0.9	22
8	The Spec# Programming System: An Overview. <i>Lecture Notes in Computer Science</i> , <b>2005</b> , 49-69	0.9	407
7	Towards a Tool Environment for Model-Based Testing with AsmL. <i>Lecture Notes in Computer Science</i> , <b>2004</b> , 252-266	0.9	20
6	Optimal strategies for testing nondeterministic systems. <i>Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM</i> , <b>2004</b> , 29, 55-64	0.4	13
5	Optimal strategies for testing nondeterministic systems <b>2004</b> ,		28
4	Verification of Object-Oriented Programs with Invariants.. <i>Journal of Object Technology</i> , <b>2004</b> , 3, 27	1.4	168
3	Runtime verification of .NET contracts. <i>Journal of Systems and Software</i> , <b>2003</b> , 65, 199-208	3.3	38
2	Generating finite state machines from abstract state machines. <i>Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM</i> , <b>2002</b> ,	0.4	61
1	Massive parallelization of divide-and-conquer algorithms over powerlists. <i>Science of Computer Programming</i> , <b>1996</b> , 26, 59-78	1.1	7