

# Reiner Höhnle

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/1437427/publications.pdf>

Version: 2024-02-01

133  
papers

2,236  
citations

279798

23  
h-index

302126

39  
g-index

152  
all docs

152  
docs citations

152  
times ranked

721  
citing authors

#	ARTICLE	IF	CITATIONS
1	Finding Semantic Bugs Fast. Lecture Notes in Computer Science, 2022, , 145-154.	1.3	1
2	Dijkstraâ€™s Legacy on Program Verification. , 2022, , 105-140.		1
3	Automated model analysis tools and techniques presented at FASE 2019. International Journal on Software Tools for Technology Transfer, 2021, 23, 285-287.	1.9	0
4	An Architectural Pattern to Realize Multi Software Product Lines in Java. , 2021, , .		2
5	Automated model extraction: From non-deterministic C code to active objects. Science of Computer Programming, 2021, 204, 102597.	1.9	3
6	Variability modules for Java-like languages. , 2021, , .		1
7	Delta-based verification of software product families. , 2021, , .		1
8	Towards a Circular Economy of Industrial Software. Procedia CIRP, 2020, 90, 37-42.	1.9	4
9	Automating Software Re-engineering. Lecture Notes in Computer Science, 2020, , 3-8.	1.3	1
10	The Trace Modality. Lecture Notes in Computer Science, 2020, , 124-140.	1.3	1
11	Who Carries the Burden of Modularity?. Lecture Notes in Computer Science, 2020, , 3-21.	1.3	1
12	Behavioral Contracts for Cooperative Scheduling. Lecture Notes in Computer Science, 2020, , 85-121.	1.3	4
13	Automated Planning of ETCS Tracks. Lecture Notes in Computer Science, 2019, , 79-90.	1.3	4
14	Verifying OpenJDKâ€™s Sort Method for Generic Collections. Journal of Automated Reasoning, 2019, 62, 93-126.	1.4	19
15	The Symbolic Execution Debugger (SED): a platform for interactive symbolic execution, debugging, verification and more. International Journal on Software Tools for Technology Transfer, 2019, 21, 485-513.	1.9	9
16	A Program Logic for Dependence Analysis. Lecture Notes in Computer Science, 2019, , 83-100.	1.3	3
17	Deductive Software Verification: From Pen-and-Paper Proofs to Industrial Tools. Lecture Notes in Computer Science, 2019, , 345-373.	1.3	43
18	Asynchronous Cooperative Contracts for Cooperative Scheduling. Lecture Notes in Computer Science, 2019, , 48-66.	1.3	2

#	ARTICLE	IF	CITATIONS
19	A Survey of Active Object Languages. ACM Computing Surveys, 2018, 50, 1-39.	23.0	65
20	Interoperability of software product line variants. , 2018, , .		4
21	Modular, Correct Compilation with Automatic Soundness Proofs. Lecture Notes in Computer Science, 2018, , 424-447.	1.3	7
22	Same Same But Different: Interoperability of Software Product Line Variants. , 2018, , 99-117.		3
23	Formal modeling and analysis of railway operations with active objects. Science of Computer Programming, 2018, 166, 167-193.	1.9	20
24	Constraint-Based Behavioral Consistency of Evolving Software Systems. Lecture Notes in Computer Science, 2018, , 205-218.	1.3	3
25	Automatic detection and demonstrator generation for information flow leaks in object-oriented programs. Computers and Security, 2017, 67, 335-349.	6.0	4
26	Uniform Modeling of Railway Operations. Communications in Computer and Information Science, 2017, , 55-71.	0.5	13
27	A Unified and Formal Programming Model for Deltas and Traits. Lecture Notes in Computer Science, 2017, , 424-441.	1.3	11
28	Locally Abstract, Globally Concrete Semantics of Concurrent Programming Languages. Lecture Notes in Computer Science, 2017, , 22-43.	1.3	11
29	Inferring Secrets by Guided Experiments. Lecture Notes in Computer Science, 2017, , 269-287.	1.3	1
30	Deductive Verification of Railway Operations. Lecture Notes in Computer Science, 2017, , 131-147.	1.3	5
31	Abstraction Refinement for the Analysis of Software Product Lines. Lecture Notes in Computer Science, 2017, , 3-20.	1.3	2
32	An empirical evaluation of two user interfaces of an interactive program verifier. , 2016, , .		9
33	A UML profile for delta-oriented programming to support software product line engineering. , 2016, , .		8
34	Correctness-by-Construction and Post-hoc Verification: Friends or Foes?. Lecture Notes in Computer Science, 2016, , 723-729.	1.3	5
35	The interactive verification debugger: effective understanding of interactive proof attempts. , 2016, , .		9
36	A General Lattice Model for Merging Symbolic Execution Branches. Lecture Notes in Computer Science, 2016, , 57-73.	1.3	15

#	ARTICLE	IF	CITATIONS
37	A formal verification framework for static analysis. <i>Software and Systems Modeling</i> , 2016, 15, 987-1012.	2.7	9
38	Array Abstraction with Symbolic Pivots. <i>Lecture Notes in Computer Science</i> , 2016, , 104-121.	1.3	2
39	Proof Repositories for Compositional Verification of Evolving Software Systems. <i>Lecture Notes in Computer Science</i> , 2016, , 130-156.	1.3	12
40	Towards Incremental Validation of Railway Systems. <i>Lecture Notes in Computer Science</i> , 2016, , 433-446.	1.3	4
41	Debugging and Visualization. <i>Lecture Notes in Computer Science</i> , 2016, , 383-413.	1.3	1
42	Abstract Interpretation. <i>Lecture Notes in Computer Science</i> , 2016, , 167-189.	1.3	2
43	Quo Vadis Formal Verification?. <i>Lecture Notes in Computer Science</i> , 2016, , 1-19.	1.3	1
44	Designing Resource-Aware Cloud Applications. <i>Computer</i> , 2015, 48, 72-75.	1.1	9
45	Exploit Generation for Information Flow Leaks in Object-Oriented Programs. <i>IFIP Advances in Information and Communication Technology</i> , 2015, , 401-415.	0.7	10
46	Testing abstract behavioral specifications. <i>International Journal on Software Tools for Technology Transfer</i> , 2015, 17, 107-119.	1.9	10
47	KeY-ABS: A Deductive Verification Tool for the Concurrent Modelling Language ABS. <i>Lecture Notes in Computer Science</i> , 2015, , 517-526.	1.3	30
48	OpenJDK's Java.util.Collection.sort() Is Broken: The Good, the Bad and the Worst Case. <i>Lecture Notes in Computer Science</i> , 2015, , 273-289.	1.3	48
49	History-Based Specification and Verification of Scalable Concurrent and Distributed Systems. <i>Lecture Notes in Computer Science</i> , 2015, , 217-233.	1.3	14
50	A Dynamic Logic with Traces and Coinduction. <i>Lecture Notes in Computer Science</i> , 2015, , 307-322.	1.3	7
51	The KeY Platform for Verification and Analysis of Java Programs. <i>Lecture Notes in Computer Science</i> , 2014, , 55-71.	1.3	37
52	Formal modeling and analysis of resource management for cloud architectures: an industrial case study using Real-Time ABS. <i>Service Oriented Computing and Applications</i> , 2014, 8, 323-339.	1.6	48
53	Analysis of Executable Software Models. <i>Lecture Notes in Computer Science</i> , 2014, , 1-25.	1.3	7
54	Visualizing Unbounded Symbolic Execution. <i>Lecture Notes in Computer Science</i> , 2014, , 82-98.	1.3	10

#	ARTICLE	IF	CITATIONS
55	An Interactive Verification Tool Meets an IDE. Lecture Notes in Computer Science, 2014, , 55-70.	1.3	10
56	Resource Analysis of Complex Programs with Cost Equations. Lecture Notes in Computer Science, 2014, , 275-295.	1.3	49
57	Fully Abstract Operation Contracts. Lecture Notes in Computer Science, 2014, , 120-134.	1.3	15
58	Introduction to Track on Engineering Virtualized Services. Lecture Notes in Computer Science, 2014, , 1-4.	1.3	0
59	HATS Abstract Behavioral Specification: The Architectural View. Lecture Notes in Computer Science, 2013, , 109-132.	1.3	8
60	Reuse in Software Verification by Abstract Method Calls. Lecture Notes in Computer Science, 2013, , 300-314.	1.3	24
61	Program Transformation Based on Symbolic Execution and Deduction. Lecture Notes in Computer Science, 2013, , 289-304.	1.3	5
62	The Abstract Behavioral Specification Language: A Tutorial Introduction. Lecture Notes in Computer Science, 2013, , 1-37.	1.3	29
63	Engineering virtualized services. , 2013, , .		11
64	Verified Resource Guarantees for Heap Manipulating Programs. Lecture Notes in Computer Science, 2012, , 130-145.	1.3	9
65	Formal Modeling of Resource Management for Cloud Architectures: An Industrial Case Study. Lecture Notes in Computer Science, 2012, , 91-106.	1.3	15
66	A Liskov Principle for Delta-Oriented Programming. Lecture Notes in Computer Science, 2012, , 32-46.	1.3	30
67	Adaptable and Evolving Software for Eternal Systems. Lecture Notes in Computer Science, 2012, , 1-3.	1.3	1
68	Formal Methods in Software Product Line Engineering. Computer, 2011, 44, 82-85.	1.1	38
69	Preface: Special Issue of Selected Extended Papers of IJCAR 2010. Journal of Automated Reasoning, 2011, 47, 337-339.	1.4	0
70	Verified resource guarantees using COSTA and KeY. , 2011, , .		17
71	Modeling Spatial and Temporal Variability with the HATS Abstract Behavioral Modeling Language. Lecture Notes in Computer Science, 2011, , 417-457.	1.3	35
72	ABS: A Core Language for Abstract Behavioral Specification. Lecture Notes in Computer Science, 2011, , 142-164.	1.3	236

#	ARTICLE	IF	CITATIONS
73	Specifying Imperative ML-Like Programs Using Dynamic Logic. Lecture Notes in Computer Science, 2011, , 122-137.	1.3	0
74	Program Specialization via a Software Verification Tool. Lecture Notes in Computer Science, 2011, , 80-101.	1.3	3
75	Tests and Proofs. Journal of Automated Reasoning, 2010, 45, 327-329.	1.4	0
76	A visual interactive debugger based on symbolic execution. , 2010, , .		14
77	Abstract Interpretation of Symbolic Execution with Explicit State Updates. Lecture Notes in Computer Science, 2009, , 247-277.	1.3	14
78	Integrated and Tool-Supported Teaching of Testing, Debugging, and Verification. Lecture Notes in Computer Science, 2009, , 125-143.	1.3	3
79	Integration of a security type system into a program logic. Theoretical Computer Science, 2008, 402, 172-189.	0.9	5
80	KeY: A Formal Method for Object-Oriented Systems. Lecture Notes in Computer Science, 2007, , 32-43.	1.3	7
81	The KeY system 1.0 (Deduction Component). Lecture Notes in Computer Science, 2007, , 379-384.	1.3	8
82	KeY-C: A Tool for Verification of C Programs. Lecture Notes in Computer Science, 2007, , 385-390.	1.3	7
83	Generating Unit Tests from Formal Proofs. Lecture Notes in Computer Science, 2007, , 169-188.	1.3	38
84	Verification by Parallelization of Parametric Code. Lecture Notes in Computer Science, 2007, , 138-159.	1.3	0
85	Verifying Object-Oriented Programs with KeY: AÄTutorial. Lecture Notes in Computer Science, 2007, , 70-101.	1.3	4
86	Intelligent Systems and Formal Methods in Software Engineering. IEEE Intelligent Systems, 2006, 21, 71-81.	4.0	24
87	Integration of a Security Type System into a Program Logic. , 2006, , 116-131.		4
88	The KeY tool. Software and Systems Modeling, 2005, 4, 32-54.	2.7	161
89	Integration of informal and formal development of object-oriented safety-critical software. International Journal on Software Tools for Technology Transfer, 2005, 7, 197-211.	1.9	2
90	A Theorem Proving Approach to Analysis of Secure Information Flow. Lecture Notes in Computer Science, 2005, , 193-209.	1.3	102

#	ARTICLE	IF	CITATIONS
91	Many-Valued Logic, Partiality, and Abstraction in Formal Specification Languages. Logic Journal of the IGPL, 2005, 13, 415-433.	1.5	20
92	Normal Forms for Knowledge Compilation. Lecture Notes in Computer Science, 2005, , 304-313.	1.3	7
93	Verification of Safety Properties in the Presence of Transactions. Lecture Notes in Computer Science, 2005, , 151-171.	1.3	11
94	Linearity and regularity with negation normal form. Theoretical Computer Science, 2004, 328, 325-354.	0.9	9
95	Using a Software Testing Technique to Improve Theorem Proving. Lecture Notes in Computer Science, 2004, , 30-41.	1.3	4
96	Integration of Informal and Formal Development of Object-Oriented Safety-Critical Software. Electronic Notes in Theoretical Computer Science, 2003, 80, 1-23.	0.9	1
97	Complexity of Many-valued Logics. Studies in Fuzziness and Soft Computing, 2003, , 211-233.	0.8	3
98	Fair Constraint Merging Tableaux in Lazy Functional Programming Style. Lecture Notes in Computer Science, 2003, , 252-256.	1.3	0
99	An Authoring Tool for Informal and Formal Requirements Specifications. Lecture Notes in Computer Science, 2002, , 233-248.	1.3	25
100	The Key System: Integrating Object-Oriented Design and Formal Methods. Lecture Notes in Computer Science, 2002, , 327-330.	1.3	17
101	Advanced Many-Valued Logics. , 2001, , 297-395.		43
102	Tableaux and Related Methods. , 2001, , 101-178.		62
103	Proof Theory of Many-Valued Logic and Linear Optimization. , 2001, , 15-33.		1
104	Ordered Resolution vs. Connection Graph resolution. Lecture Notes in Computer Science, 2001, , 182-194.	1.3	2
105	The Approach: Integrating Object Oriented Design and Formal Verification. Lecture Notes in Computer Science, 2000, , 21-36.	1.3	34
106	Model Generation Theorem Proving with Finite Interval Constraints. Lecture Notes in Computer Science, 2000, , 285-299.	1.3	3
107	Entwurfsmustergesteuerte Erzeugung von OCL-Constraints. Informatik Aktuell, 2000, , 389-404.	0.6	10
108	The SAT Problem of Signed CNF Formulas. Applied Logic Series, 2000, , 59-80.	0.3	26

#	ARTICLE	IF	CITATIONS
109	Tableaux for Many-Valued Logics. , 1999, , 529-580.		34
110	Commodious Axiomatization of Quantifiers in Multiple-Valued Logic. Studia Logica, 1998, 61, 101-121.	0.6	17
111	Some Remarks on Completeness, Connection Graph Resolution, and Link Deletion. Lecture Notes in Computer Science, 1998, , 172-186.	1.3	1
112	Completeness for linear regular negation normal form inference systems. Lecture Notes in Computer Science, 1997, , 590-599.	1.3	4
113	Ordered tableaux: Extensions and applications. Lecture Notes in Computer Science, 1997, , 173-187.	1.3	6
114	Fast Subsumption Checks Using Anti-Links. Journal of Automated Reasoning, 1997, 18, 47-83.	1.4	5
115	Restart tableaux with selection function. Lecture Notes in Computer Science, 1997, , 219-232.	1.3	0
116	The tableau-based theorem prover 3 T A P Version 4.0. Lecture Notes in Computer Science, 1996, , 303-307.	1.3	14
117	A-ordered tableaux. Journal of Logic and Computation, 1996, 6, 819-833.	0.8	10
118	Deduction by combining semantic tableaux and integer programming. Lecture Notes in Computer Science, 1996, , 52-63.	1.3	1
119	Automated Deduction and Integer Programming. Journal of Zoological Systematics and Evolutionary Research, 1995, , 67-86.	1.4	1
120	Short Conjunctive Normal Forms in Finitely Valued Logics. Journal of Logic and Computation, 1994, 4, 905-927.	0.8	65
121	Many-valued logic and mixed integer programming. Annals of Mathematics and Artificial Intelligence, 1994, 12, 231-263.	1.3	67
122	The liberalized ?-rule in free variable semantic tableaux. Journal of Automated Reasoning, 1994, 13, 211-221.	1.4	47
123	Improving temporal logic tableaux using integer constraints. , 1994, , 535-539.		7
124	On anti-links. Lecture Notes in Computer Science, 1994, , 275-289.	1.3	1
125	The even more liberalized $\hat{\tau}$ -rule in free variable Semantic Tableaux. , 1993, , 108-119.		25
126	Short CNF in finitely-valued logics. Lecture Notes in Computer Science, 1993, , 49-58.	1.3	5



#	ARTICLE	IF	CITATIONS
127	Verification of Switch-level designs with many-valued logic. Lecture Notes in Computer Science, 1993, , 158-169.	1.3	21
128	An improved method for adding equality to free variable semantic tableaux. Lecture Notes in Computer Science, 1992, , 507-521.	1.3	15
129	The tableau-based theorem prover 3 T A P for multiple-valued logics. Lecture Notes in Computer Science, 1992, , 758-760.	1.3	3
130	Towards an efficient tableau proof procedure for multiple-valued logics. Lecture Notes in Computer Science, 1991, , 248-260.	1.3	32
131	Prototyping Formal System Models with Active Objects. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 279, 52-67.	0.8	3
132	Analysis of SLA Compliance in the Cloud - An Automated, Model-based Approach. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 302, 1-15.	0.8	2
133	24 Challenges in Deductive Software Verification. , 0, , .		1