

Cliff B Jones

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/2412918/publications.pdf>

Version: 2024-02-01

67
papers

1,610
citations

623574

14
h-index

315616

38
g-index

85
all docs

85
docs citations

85
times ranked

687
citing authors

#	ARTICLE	IF	CITATIONS
1	Formal methods. ACM Computing Surveys, 1996, 28, 626-643.	16.1	739
2	A logic covering undefinedness in program proofs. Acta Informatica, 1984, 21, 251-269.	0.5	146
3	Accommodating interference in the formal design of concurrent object-based programs. Formal Methods in System Design, 1996, 8, 105-122.	0.9	59
4	The early search for tractable ways of reasoning about programs. IEEE Annals of the History of Computing, 2003, 25, 26-49.	0.2	55
5	A typed logic of partial functions reconstructed classically. Acta Informatica, 1994, 31, 399-430.	0.5	52
6	An Early Program Proof by Alan Turing. IEEE Annals of the History of Computing, 1984, 6, 139-143.	0.2	51
7	A Structural Proof of the Soundness of Rely/guarantee Rules. Journal of Logic and Computation, 2007, 17, 807-841.	0.5	48
8	Roadmap for enhanced languages and methods to aid verification. , 2006, , .		41
9	Determining the Specification of a Control System from That of Its Environment. Lecture Notes in Computer Science, 2003, , 154-169.	1.0	33
10	Assessing the Long-Term Performance of Cross-Sectoral Strategies for National Infrastructure. Journal of Infrastructure Systems, 2014, 20, 04014014.	1.0	28
11	Partial functions and logics: A warning. Information Processing Letters, 1995, 54, 65-67.	0.4	24
12	Splitting atoms safely. Theoretical Computer Science, 2007, 375, 109-119.	0.5	23
13	Balancing expressiveness in formal approaches to concurrency. Formal Aspects of Computing, 2015, 27, 475-497.	1.4	18
14	Deriving Specifications for Systems That Are Connected to the Physical World. , 2007, , 364-390.		18
15	Constructing a theory of a data structure as an aid to program development. Acta Informatica, 1979, 11, 119.	0.5	17
16	Operational semantics: Concepts and their expression. Information Processing Letters, 2003, 88, 27-32.	0.4	17
17	Reasoning About Partial Functions in the Formal Development of Programs. Electronic Notes in Theoretical Computer Science, 2006, 145, 3-25.	0.9	17
18	Comparing Degrees of Non-Determinism in Expression Evaluation. Computer Journal, 2013, 56, 741-755.	1.5	15

#	ARTICLE	IF	CITATIONS
19	Scientific Decisions which Characterize VDM. Lecture Notes in Computer Science, 1999, , 28-47.	1.0	14
20	Understanding the differences between VDM and Z. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1994, 19, 75-81.	0.5	12
21	The meta-language: A reference manual. Lecture Notes in Computer Science, 1978, , 218-277.	1.0	12
22	Elucidating concurrent algorithms via layers of abstraction and reification. Formal Aspects of Computing, 2011, 23, 289-306.	1.4	11
23	Splitting Atoms with Rely/Guarantee Conditions Coupled with Data Reification. Lecture Notes in Computer Science, 2008, , 360-377.	1.0	10
24	The atomic manifesto. SIGMOD Record, 2005, 34, 63-69.	0.7	9
25	Possible values: Exploring a concept for concurrency. Journal of Logical and Algebraic Methods in Programming, 2016, 85, 972-984.	0.4	9
26	A formal definition of ALGOL 60 as described in the 1975 modified report. Lecture Notes in Computer Science, 1978, , 305-336.	1.0	9
27	Non-Interference Properties of a Concurrent Object-Based Language: Proofs Based on an Operational Semantics. Kluwer International Series in Engineering and Computer Science, 1996, , 1-22.	0.2	9
28	Reasoning about Separation Using Abstraction and Reification. Lecture Notes in Computer Science, 2015, , 3-19.	1.0	8
29	A Guide to Rely/Guarantee Thinking. Lecture Notes in Computer Science, 2018, , 1-38.	1.0	7
30	Dynamic syntax: A concept for the definition of the syntax of programming languages. Annual Review in Automatic Programming, 1973, 7, 115-142.	0.2	6
31	An Approach to Splitting Atoms Safely. Electronic Notes in Theoretical Computer Science, 2006, 155, 43-60.	0.9	5
32	Formal Modelling of Dynamic Coalitions, with an Application in Chemical Engineering. , 2006, , .		5
33	The connection between two ways of reasoning about partial functions. Information Processing Letters, 2008, 107, 128-132.	0.4	5
34	Some practical problems and their influence on semantics. Lecture Notes in Computer Science, 1996, , 1-17.	1.0	5
35	Some mistakes I have and what I have learned from them. Lecture Notes in Computer Science, 1998, , 7-20.	1.0	4
36	Understanding Programming Language Concepts Via Operational Semantics. Lecture Notes in Computer Science, 2007, , 177-235.	1.0	4

#	ARTICLE	IF	CITATIONS
37	What Can the pi-calculus Tell Us About the Mondex Purse System?. , 2007, , .		3
38	Reasoning about programs via operational semantics: requirements for a support system. Automated Software Engineering, 2008, 15, 299-312.	2.2	3
39	Investigating the limits of rely/guarantee relations based on a concurrent garbage collector example. Formal Aspects of Computing, 2019, 31, 353-374.	1.4	3
40	General Lessons from a Rely/Guarantee Development. Lecture Notes in Computer Science, 2017, , 3-22.	1.0	3
41	Abstraction as a Unifying Link for Formal Approaches to Concurrency. Lecture Notes in Computer Science, 2012, , 1-15.	1.0	3
42	The atomic manifesto. Operating Systems Review (ACM), 2005, 39, 41-46.	1.5	2
43	Turing's 1949 Paper in Context. Lecture Notes in Computer Science, 2017, , 32-41.	1.0	2
44	Deriving Specifications of Control Programs for Cyber Physical Systems. Computer Journal, 2020, 63, 774-790.	1.5	2
45	The role of proof obligations in software design. Lecture Notes in Computer Science, 1985, , 27-41.	1.0	2
46	Formal Semantics of ALGOL 60: Four Descriptions in their Historical Context. Philosophical Studies Series, 2018, , 83-152.	1.3	2
47	Structured Handling of Online Interface Upgrades in Integrating Dependable Systems of Systems. Lecture Notes in Computer Science, 2003, , 73-86.	1.0	2
48	Ours Is to Reason Why. Lecture Notes in Computer Science, 2013, , 227-243.	1.0	2
49	Muffin: A user interface design experiment for a Theorem Proving Assistant. Lecture Notes in Computer Science, 1988, , 337-375.	1.0	2
50	Challenges for Formal Semantic Description: Responses from the Main Approaches. Lecture Notes in Computer Science, 2018, , 176-217.	1.0	2
51	Panelist position statement: reasoning about the design of programs. Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences, 2005, 363, 2395-2396.	1.6	1
52	Revising basic theorem proving algorithms to cope with the logic of partial functions. Science of Computer Programming, 2014, 94, 238-252.	1.5	1
53	The Turing Guide. Formal Aspects of Computing, 2017, 29, 1121-1122.	1.4	1
54	A Formal Basis for Some Dependability Notions. Lecture Notes in Computer Science, 2003, , 191-206.	1.0	1

#	ARTICLE	IF	CITATIONS
55	Thinking Tools for the Future of Computing Science. Lecture Notes in Computer Science, 2001, , 112-130.	1.0	1
56	Some Interdisciplinary Observations about Getting the “Right” Specification. Lecture Notes in Computer Science, 2008, , 64-69.	1.0	1
57	Experience of Deployment in the Automotive Industry. , 2013, , 13-26.		1
58	A Semantic Analysis of Logics That Cope with Partial Terms. Lecture Notes in Computer Science, 2012, , 252-265.	1.0	1
59	The role of structure: a dependability perspective. , 2006, , 3-15.		1
60	Specification Before Satisfaction: The Case for Research into Obtaining the Right Specification “Extended Abstract”. Lecture Notes in Computer Science, 2005, , 1-5.	1.0	0
61	Reflections on, and Predictions for, Support Systems for the Development of Programs. , 2008, , .		0
62	John McCarthy (1927–2011). Formal Aspects of Computing, 2012, 24, 305-306.	1.4	0
63	Expressiveness of Notations for Reasoning about Concurrency. , 2013, , .		0
64	Digital communications and information systems. , 0, , 181-202.		0
65	Concurrency: Handling Interference Formally. Lecture Notes in Computer Science, 2018, , 26-43.	1.0	0
66	A Model for Capturing and Replaying Proof Strategies. Lecture Notes in Computer Science, 2014, , 183-199.	1.0	0
67	Reasoning About Shared-Variable Concurrency: Interactions Between Research Threads. Lecture Notes in Computer Science, 2020, , 54-72.	1.0	0