

Fausto Spoto

List of Publications by Year in Descending Order

Source: <https://exaly.com/author-pdf/8604519/fausto-spoto-publications-by-year.pdf>
Version: 2024-04-10

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.

69 papers	694 citations	14 h-index	24 g-index
74 ext. papers	814 ext. citations	1.3 avg, IF	4.45 L-index

#	Paper	IF	Citations
69	On the Termination of Borrow Checking in Featherweight Rust. <i>Lecture Notes in Computer Science</i> , 2022 , 411-430	0.9	0
68	Intents Analysis of Android Apps for Confidentiality Leakage Detection. <i>Advances in Intelligent Systems and Computing</i> , 2021 , 43-65	0.4	1
67	Static analysis for discovering IoT vulnerabilities. <i>International Journal on Software Tools for Technology Transfer</i> , 2021 , 23, 71-88	1.3	11
66	On-Chain Smart Contract Verification over Tendermint. <i>Lecture Notes in Computer Science</i> , 2021 , 333-347	0.9	0
65	Analysing omics data sets with weighted nodes networks (WNNets). <i>Scientific Reports</i> , 2021 , 11, 14447	4.9	0
64	Static Privacy Analysis by Flow Reconstruction of Tainted Data. <i>International Journal of Software Engineering and Knowledge Engineering</i> , 2021 , 31, 973-1016	1	0
63	From CIL to Java bytecode: Semantics-based translation for static analysis leveraging. <i>Science of Computer Programming</i> , 2020 , 191, 102392	1.1	2
62	Cross-program taint analysis for IoT systems 2020 ,		2
61	(mathsf {BackFlow}): Backward Context-Sensitive Flow Reconstruction of Taint Analysis Results. <i>Lecture Notes in Computer Science</i> , 2020 , 23-43	0.9	3
60	A Java Framework for Smart Contracts. <i>Lecture Notes in Computer Science</i> , 2020 , 122-137	0.9	2
59	Enforcing Determinism of Java Smart Contracts. <i>Lecture Notes in Computer Science</i> , 2020 , 568-583	0.9	2
58	Static analysis of Android Auto infotainment and on-board diagnostics II apps. <i>Software - Practice and Experience</i> , 2019 , 49, 1131	2.5	8
57	Static Identification of Injection Attacks in Java. <i>ACM Transactions on Programming Languages and Systems</i> , 2019 , 41, 1-58	1.6	8
56	Checking Array Bounds by Abstract Interpretation and Symbolic Expressions. <i>Lecture Notes in Computer Science</i> , 2018 , 706-722	0.9	0
55	Tailoring Taint Analysis to GDPR. <i>Lecture Notes in Computer Science</i> , 2018 , 63-76	0.9	6
54	Static Analysis of Android Apps Interaction with Automotive CAN. <i>Lecture Notes in Computer Science</i> , 2018 , 114-123	0.9	2
53	SDLI: Static Detection of Leaks Across Intents 2018 ,		2

52	Vulnerability analysis of Android auto infotainment apps 2018 ,		8
51	CIL to Java-bytecode translation for static analysis leveraging 2018 ,		1
50	Creating, generating and comparing random network models with Network Randomizer. <i>F1000Research</i> , 2016 , 5, 2524	3.6	19
49	Creating, generating and comparing random network models with NetworkRandomizer. <i>F1000Research</i> , 2016 , 5, 2524	3.6	11
48	The Julia Static Analyzer for Java. <i>Lecture Notes in Computer Science</i> , 2016 , 39-57	0.9	15
47	Semantics for Locking Specifications. <i>Lecture Notes in Computer Science</i> , 2016 , 355-372	0.9	2
46	Locking discipline inference and checking 2016 ,		6
45	Finding the shortest path with PesCa: a tool for network reconstruction. <i>F1000Research</i> , 2015 , 4, 484	3.6	22
44	Finding the shortest path with PesCa: a tool for network reconstruction. <i>F1000Research</i> , 2015 , 4, 484	3.6	17
43	Boolean Formulas for the Static Identification of Injection Attacks in Java. <i>Lecture Notes in Computer Science</i> , 2015 , 130-145	0.9	10
42	Field-sensitive unreachability and non-cyclicity analysis. <i>Science of Computer Programming</i> , 2014 , 95, 359-375	1	
41	Biological network analysis with CentiScaPe: centralities and experimental dataset integration. <i>F1000Research</i> , 2014 , 3, 139	3.6	53
40	An operational semantics for android activities 2014 ,		14
39	Biological network analysis with CentiScaPe: centralities and experimental dataset integration. <i>F1000Research</i> , 2014 , 3, 139	3.6	71
38	A Thread-Safe Library for Binary Decision Diagrams. <i>Lecture Notes in Computer Science</i> , 2014 , 35-49	0.9	8
37	Inferring complete initialization of arrays. <i>Theoretical Computer Science</i> , 2013 , 484, 16-40	1.1	3
36	Reachability analysis of program variables. <i>ACM Transactions on Programming Languages and Systems</i> , 2013 , 35, 1-68	1.6	9
35	Static analysis of Android programs. <i>Information and Software Technology</i> , 2012 , 54, 1192-1201	3.4	46

34	Automaton-Based Array Initialization Analysis. <i>Lecture Notes in Computer Science</i> , 2012 , 420-432	0.9	2
33	Reachability Analysis of Program Variables. <i>Lecture Notes in Computer Science</i> , 2012 , 423-438	0.9	5
32	Definite Expression Aliasing Analysis for Java Bytecode. <i>Lecture Notes in Computer Science</i> , 2012 , 74-89	0.9	13
31	Precise null-pointer analysis. <i>Software and Systems Modeling</i> , 2011 , 10, 219-252	1.9	22
30	Inference of field initialization 2011 ,		8
29	Static Analysis of Android Programs. <i>Lecture Notes in Computer Science</i> , 2011 , 439-445	0.9	2
28	A termination analyzer for Java bytecode based on path-length. <i>ACM Transactions on Programming Languages and Systems</i> , 2010 , 32, 1-70	1.6	79
27	Magic-sets for localised analysis of Java bytecode. <i>Higher-Order and Symbolic Computation</i> , 2010 , 23, 29-86		
26	The Nullness Analyser of julia. <i>Lecture Notes in Computer Science</i> , 2010 , 405-424	0.9	3
25	Experiments with Non-Termination Analysis for Java Bytecode. <i>Electronic Notes in Theoretical Computer Science</i> , 2009 , 253, 83-96	0.7	8
24	Using CLP Simplifications to Improve Java Bytecode Termination Analysis. <i>Electronic Notes in Theoretical Computer Science</i> , 2009 , 253, 129-144	0.7	3
23	Nullness Analysis in Boolean Form 2008 ,		13
22	Optimality and condensing of information flow through linear refinement. <i>Theoretical Computer Science</i> , 2007 , 388, 53-82	1.1	1
21	Magic-Sets Transformation for the Analysis of Java Bytecode. <i>Lecture Notes in Computer Science</i> , 2007 , 452-467	0.9	5
20	Deriving escape analysis by abstract interpretation. <i>Higher-Order and Symbolic Computation</i> , 2006 , 19, 415-463		2
19	Information Flow Analysis for Java Bytecode. <i>Lecture Notes in Computer Science</i> , 2005 , 346-362	0.9	39
18	Pair-Sharing Analysis of Object-Oriented Programs. <i>Lecture Notes in Computer Science</i> , 2005 , 320-335	0.9	29
17	Information Flow Is Linear Refinement of Constancy. <i>Lecture Notes in Computer Science</i> , 2005 , 351-365	0.9	1

16	Detecting Non-cyclicity by Abstract Compilation into Boolean Functions. <i>Lecture Notes in Computer Science</i> , 2005 , 95-110	0.9	16
15	Class analyses as abstract interpretations of trace semantics. <i>ACM Transactions on Programming Languages and Systems</i> , 2003 , 25, 578-630	1.6	29
14	Pair-independence and freeness analysis through linear refinement. <i>Information and Computation</i> , 2003 , 182, 14-52	0.8	4
13	Logic programs as compact denotations. <i>Computer Languages, Systems and Structures</i> , 2003 , 29, 45-73		1
12	Logic Programs as Compact Denotations. <i>Lecture Notes in Computer Science</i> , 2003 , 339-356	0.9	2
11	A Refinement of the Escape Property. <i>Lecture Notes in Computer Science</i> , 2002 , 154-166	0.9	1
10	A Foundation of Escape Analysis*. <i>Lecture Notes in Computer Science</i> , 2002 , 380-395	0.9	3
9	Abstract Compilation for Sharing Analysis. <i>Lecture Notes in Computer Science</i> , 2001 , 311-325	0.9	1
8	Class Analysis of Object-Oriented Programs through Abstract Interpretation. <i>Lecture Notes in Computer Science</i> , 2001 , 261-275	0.9	7
7	Watchpoint Semantics: A Tool for Compositional and Focussed Static Analyses. <i>Lecture Notes in Computer Science</i> , 2001 , 127-145	0.9	4
6	Operational and goal-independent denotational semantics for Prolog with cut. <i>The Journal of Logic Programming</i> , 2000 , 42, 1-46		10
5	Analysis of Downward Closed Properties of Logic Programs. <i>Lecture Notes in Computer Science</i> , 2000 , 181-196	0.9	
4	Non pair-sharing and freeness analysis through linear refinement. <i>ACM SIGPLAN Notices</i> , 1999 , 34, 52-61	0.2	4
3	Freeness Analysis Through Linear Refinement. <i>Lecture Notes in Computer Science</i> , 1999 , 85-100	0.9	1
2	An experiment in domain refinement: Type domains and type representations for logic programs. <i>Lecture Notes in Computer Science</i> , 1998 , 152-169	0.9	7
1	Abstract Interpretation of Prolog Programs. <i>Lecture Notes in Computer Science</i> , 1998 , 455-470	0.9	4