Eran Yahav

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/12075841/publications.pdf

Version: 2024-02-01

44 papers	2,378 citations	623734 14 h-index	677142 22 g-index
45 all docs	45 docs citations	45 times ranked	879 citing authors

#	Article	IF	Citations
1	Adversarial examples for models of code. , 2020, 4, 1-30.		58
2	Neural reverse engineering of stripped binaries using augmented control flow graphs. , 2020, 4, 1-28.		33
3	code2vec: learning distributed representations of code. , 2019, 3, 1-29.		606
4	Programming not only by example. , 2018, , .		23
5	A general path-based representation for predicting program properties. ACM SIGPLAN Notices, 2018, 53, 404-419.	0.2	57
6	Effective abstractions for verification under relaxed memory models. Computer Languages, Systems and Structures, 2017, 47, 62-76.	1.4	7
7	Programming with â€œBig Codeâ€: Foundations and Trends in Programming Languages, 2016, 3, 231-284.	1.8	12
8	Statistical similarity of binaries. ACM SIGPLAN Notices, 2016, 51, 266-280.	0.2	43
9	Leveraging a corpus of natural language descriptions for program similarity. , 2016, , .		12
10	Estimating types in binaries using predictive modeling. ACM SIGPLAN Notices, 2016, 51, 313-326.	0.2	2
11	Tracelet-based code search in executables. , 2014, , .		63
12	Code completion with statistical language models. , 2014, , .		272
13	Abstract semantic differencing via speculative correlation. , 2014, , .		25
14	Synthesis of Memory Fences via Refinement Propagation. Lecture Notes in Computer Science, 2014, , 237-252.	1.3	12
15	Code completion with statistical language models. ACM SIGPLAN Notices, 2014, 49, 419-428.	0.2	139
16	Abstraction-guided synthesis of synchronization. International Journal on Software Tools for Technology Transfer, 2013, 15, 413-431.	1.9	6
17	Predicate Abstraction for Relaxed Memory Models. Lecture Notes in Computer Science, 2013, , 84-104.	1.3	28
18	Dynamic synthesis for relaxed memory models. , 2012, , .		42

#	Article	IF	Citations
19	Typestate-based semantic code search over partial programs. , 2012, , .		66
20	Dynamic synthesis for relaxed memory models. ACM SIGPLAN Notices, 2012, 47, 429-440.	0.2	16
21	Automatic fine-grain locking using shape properties. , 2011, , .		23
22	Partial-coherence abstractions for relaxed memory models. , 2011, , .		47
23	Asynchronous assertions. ACM SIGPLAN Notices, 2011, 46, 275-288.	0.2	0
24	Abstraction-guided synthesis of synchronization. , 2010, , .		95
25	PHALANX. ACM SIGPLAN Notices, 2010, 45, 41-50.	0.2	0
26	Chameleon. , 2009, , .		82
27	Chameleon. ACM SIGPLAN Notices, 2009, 44, 408-418.	0.2	24
28	Inferring Synchronization under Limited Observability. Lecture Notes in Computer Science, 2009, , 139-154.	1.3	41
29	Experience with Model Checking Linearizability. Lecture Notes in Computer Science, 2009, , 261-278.	1.3	68
30	Generating precise and concise procedure summaries. ACM SIGPLAN Notices, 2008, 43, 221-234.	0.2	12
31	Deriving linearizable fine-grained concurrent objects. ACM SIGPLAN Notices, 2008, 43, 125-135.	0.2	16
32	The CLOSER. , 2008, , .		26
33	QVM. , 2008, , .		77
34	QVM. ACM SIGPLAN Notices, 2008, 43, 143-162.	0.2	18
35	CGCExplorer. , 2007, , .		23
36	CGCExplorer. ACM SIGPLAN Notices, 2007, 42, 456-467.	0.2	3

#	Article	IF	Citations
37	Correctness-preserving derivation of concurrent garbage collection algorithms. ACM SIGPLAN Notices, 2006, 41, 341-353.	0.2	5
38	Effective typestate verification in the presence of aliasing. , 2006, , .		106
39	Correctness-preserving derivation of concurrent garbage collection algorithms. , 2006, , .		25
40	InterproceduralÂShapeÂAnalysis forÂCutpoint-FreeÂPrograms. Lecture Notes in Computer Science, 2005, , 284-302.	1.3	57
41	On the Utility of Canonical Abstraction. , 2005, , 215-253.		0
42	Verifying safety properties using separation and heterogeneous abstractions. , 2004, , .		40
43	Automatically Verifying Concurrent Queue Algorithms. Electronic Notes in Theoretical Computer Science, 2003, 89, 450-463.	0.9	16
44	Verifying safety properties of concurrent Java programs using 3-valued logic. ACM SIGPLAN Notices, 2001, 36, 27-40.	0.2	30