

# Daniel Kroening

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/5593292/publications.pdf>

Version: 2024-02-01

200  
papers

5,689  
citations

147801

31  
h-index

123424

61  
g-index

211  
all docs

211  
docs citations

211  
times ranked

1844  
citing authors

#	ARTICLE	IF	CITATIONS
1	A Tool for Checking ANSI-C Programs. Lecture Notes in Computer Science, 2004, , 168-176.	1.3	851
2	A Survey of Automated Techniques for Formal Software Verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27, 1165-1178.	2.7	237
3	A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review, 2020, 37, 100270.	15.3	203
4	CBMC – C Bounded Model Checker. Lecture Notes in Computer Science, 2014, , 389-391.	1.3	198
5	Concolic testing for deep neural networks. , 2018, , .		190
6	SATABS: SAT-Based Predicate Abstraction for ANSI-C. Lecture Notes in Computer Science, 2005, , 570-574.	1.3	185
7	Predicate Abstraction of ANSI-C Programs Using SAT. Formal Methods in System Design, 2004, 25, 105-127.	0.8	127
8	Error explanation with distance metrics. International Journal on Software Tools for Technology Transfer, 2006, 8, 229-247.	1.9	119
9	Behavioral consistency of C and verilog programs using bounded model checking. , 2003, , .		114
10	Partial Orders for Efficient Bounded Model Checking of Concurrent Software. Lecture Notes in Computer Science, 2013, , 141-157.	1.3	101
11	Software Verification for Weak Memory via Program Transformation. Lecture Notes in Computer Science, 2013, , 512-532.	1.3	89
12	Dynamic Cutoff Detection in Parameterized Concurrent Programs. Lecture Notes in Computer Science, 2010, , 645-659.	1.3	78
13	Interpolant Strength. Lecture Notes in Computer Science, 2010, , 129-145.	1.3	70
14	Decision Procedures. Texts in Theoretical Computer Science, 2016, , .	0.8	68
15	Completeness and Complexity of Bounded Model Checking. Lecture Notes in Computer Science, 2004, , 85-96.	1.3	67
16	Model checking concurrent linux device drivers. , 2007, , .		64
17	Efficient Computation of Recurrence Diameters. Lecture Notes in Computer Science, 2003, , 298-309.	1.3	63
18	Deciding Bit-Vector Arithmetic with Abstraction. , 2007, , 358-372.		61

#	ARTICLE	IF	CITATIONS
19	Termination Analysis with Compositional Transition Invariants. Lecture Notes in Computer Science, 2010, , 89-103.	1.3	61
20	Software Verification Using k-Induction. Lecture Notes in Computer Science, 2011, , 351-368.	1.3	61
21	Structural Test Coverage Criteria for Deep Neural Networks. Transactions on Embedded Computing Systems, 2019, 18, 1-23.	2.9	52
22	Making the Most of BMC Counterexamples. Electronic Notes in Theoretical Computer Science, 2005, 119, 67-81.	0.9	47
23	Word level predicate abstraction and refinement for verifying RTL verilog. , 2005, , .		47
24	Understanding Counterexamples with explain. Lecture Notes in Computer Science, 2004, , 453-456.	1.3	47
25	Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the Hamming Distance. , 2019, , .		47
26	Mixed abstractions for floating-point arithmetic. , 2009, , .		45
27	Test-case generation for embedded simulink via formal concept analysis. , 2011, , .		43
28	Mutation-Based Test Case Generation for Simulink Models. Lecture Notes in Computer Science, 2010, , 208-227.	1.3	43
29	Hardware verification using ANSI-C programs as a reference. , 2003, , .		42
30	JBMC: A Bounded Model Checking Tool for Verifying Java Bytecode. Lecture Notes in Computer Science, 2018, , 183-190.	1.3	42
31	Cogent: Accurate Theorem Proving for Program Verification. Lecture Notes in Computer Science, 2005, , 296-300.	1.3	40
32	Deciding floating-point logic with abstract conflict driven clause learning. Formal Methods in System Design, 2014, 45, 213-245.	0.8	39
33	Ranking Function Synthesis for Bit-Vector Relations. Lecture Notes in Computer Science, 2010, , 236-250.	1.3	36
34	Safety Verification and Refutation by k-Invariants and k-Induction. Lecture Notes in Computer Science, 2015, , 145-161.	1.3	36
35	Loop Summarization Using Abstract Transformers. Lecture Notes in Computer Science, 2008, , 111-125.	1.3	35
36	Race analysis for systemc using model checking. ACM Transactions on Design Automation of Electronic Systems, 2010, 15, 1-32.	2.6	34

#	ARTICLE	IF	CITATIONS
37	Symbolic Model Checking for Asynchronous Boolean Programs. Lecture Notes in Computer Science, 2005, , 75-90.	1.3	34
38	Symbolic Counter Abstraction for Concurrent Software. Lecture Notes in Computer Science, 2009, , 64-78.	1.3	34
39	Automated pipeline design. , 2001, , .		33
40	Counterexample Guided Inductive Synthesis Modulo Theories. Lecture Notes in Computer Science, 2018, , 270-288.	1.3	33
41	DeepConcolic: Testing and Debugging Deep Neural Networks. , 2019, , .		32
42	An Interpolating Sequent Calculus for Quantifier-Free Presburger Arithmetic. Lecture Notes in Computer Science, 2010, , 384-399.	1.3	32
43	Don't Sit on the Fence. Lecture Notes in Computer Science, 2014, , 508-524.	1.3	31
44	Interpolation-Based Software Verification with Wolverine. Lecture Notes in Computer Science, 2011, , 573-578.	1.3	31
45	Numeric Bounds Analysis with Conflict-Driven Learning. Lecture Notes in Computer Science, 2012, , 48-63.	1.3	31
46	Verifying multi-threaded software with impact. , 2013, , .		30
47	Hardware Verification Using Software Analyzers. , 2015, , .		30
48	Loop Summarization and Termination Analysis. Lecture Notes in Computer Science, 2011, , 81-95.	1.3	28
49	Computational challenges in bounded model checking. International Journal on Software Tools for Technology Transfer, 2005, 7, 174-183.	1.9	27
50	An Interpolating Sequent Calculus for Quantifier-Free Presburger Arithmetic. Journal of Automated Reasoning, 2011, 47, 341-367.	1.4	27
51	A Widening Approach to Multithreaded Program Verification. ACM Transactions on Programming Languages and Systems, 2014, 36, 1-29.	2.1	27
52	Symmetry-Aware Predicate Abstraction for Shared-Variable Concurrent Programs. Lecture Notes in Computer Science, 2011, , 356-371.	1.3	27
53	Formal co-validation of low-level hardware/software interfaces. , 2013, , .		26
54	Abstraction-Based Satisfiability Solving of Presburger Arithmetic. Lecture Notes in Computer Science, 2004, , 308-320.	1.3	26

#	ARTICLE	IF	CITATIONS
55	Scout: A Tool for the Analysis of SystemC Models. , 2008, , 467-470.		25
56	Automatic Analysis of Scratch-Pad Memory Code for Heterogeneous Multicore Processors. Lecture Notes in Computer Science, 2010, , 280-295.	1.3	25
57	2LS for Program Analysis. Lecture Notes in Computer Science, 2016, , 905-907.	1.3	24
58	Gollum. , 2019, , .		24
59	Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs. Lecture Notes in Computer Science, 2014, , 263-278.	1.3	23
60	Formal Techniques for Effective Co-verification of Hardware/Software Co-designs. , 2017, , .		22
61	Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs. ACM Transactions on Programming Languages and Systems, 2017, 39, 1-27.	2.1	22
62	Counterexamples with Loops for Predicate Abstraction. Lecture Notes in Computer Science, 2006, , 152-165.	1.3	21
63	Explaining Image Classifiers Using Statistical Fault Localization. Lecture Notes in Computer Science, 2020, , 391-406.	1.3	21
64	Verifying C++ with STL containers via predicate abstraction. , 2007, , .		20
65	Synthesising Interprocedural Bit-Precise Termination Proofs (T). , 2015, , .		20
66	Linear Completeness Thresholds for Bounded Model Checking. Lecture Notes in Computer Science, 2011, , 557-572.	1.3	20
67	Counterexample-guided abstraction refinement for symmetric concurrent programs. Formal Methods in System Design, 2012, 41, 25-44.	0.8	19
68	Automated Formal Synthesis of Digital Controllers for State-Space Physical Plants. Lecture Notes in Computer Science, 2017, , 462-482.	1.3	19
69	Program synthesis: challenges and opportunities. Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences, 2017, 375, 20150403.	3.4	19
70	Efficient Coverability Analysis by Proof Minimization. Lecture Notes in Computer Science, 2012, , 500-515.	1.3	19
71	Ranking function synthesis for bit-vector relations. Formal Methods in System Design, 2013, 43, 93-120.	0.8	18
72	Counterexample-Guided Precondition Inference. Lecture Notes in Computer Science, 2013, , 451-471.	1.3	18

#	ARTICLE	IF	CITATIONS
73	Under-Approximating Loops in C Programs for Fast Counterexample Detection. Lecture Notes in Computer Science, 2013, , 381-396.	1.3	18
74	Evaluation of Measures for Statistical Fault Localisation and an Optimising Scheme. Lecture Notes in Computer Science, 2015, , 115-129.	1.3	18
75	Verification of SpecC using predicate abstraction. Formal Methods in System Design, 2006, 30, 5-28.	0.8	17
76	An abstraction-based decision procedure for bit-vector arithmetic. International Journal on Software Tools for Technology Transfer, 2009, 11, 95-104.	1.9	17
77	Loop summarization using state and transition invariants. Formal Methods in System Design, 2013, 42, 221-261.	0.8	17
78	Abstract conflict driven learning. , 2013, , .		17
79	Sound static deadlock analysis for C/Pthreads. , 2016, , .		17
80	$\text{SC}^2$ : Satisfiability Checking Meets Symbolic Computation. Lecture Notes in Computer Science, 2016, , 28-43.	1.3	17
81	Beyond Quantifier-Free Interpolation in Extensions of Presburger Arithmetic. Lecture Notes in Computer Science, 2011, , 88-102.	1.3	17
82	Satisfiability Solvers Are Static Analysers. Lecture Notes in Computer Science, 2012, , 317-333.	1.3	17
83	Automatic analysis of DMA races using model checking and k-induction. Formal Methods in System Design, 2011, 39, 83-113.	0.8	16
84	Accelerated test execution using GPUs. , 2014, , .		16
85	Don't Sit on the Fence. ACM Transactions on Programming Languages and Systems, 2017, 39, 1-38.	2.1	16
86	Modular Demand-Driven Analysis of Semantic Difference for Program Versions. Lecture Notes in Computer Science, 2017, , 405-427.	1.3	16
87	Model Checking Boot Code from AWS Data Centers. Lecture Notes in Computer Science, 2018, , 467-486.	1.3	16
88	Making Software Verification Tools Really Work. Lecture Notes in Computer Science, 2011, , 28-42.	1.3	16
89	Unrestricted Termination and Non-termination Arguments for Bit-Vector Programs. Lecture Notes in Computer Science, 2015, , 183-204.	1.3	16
90	Word-Level Predicate-Abstraction and Refinement Techniques for Verifying RTL Verilog. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27, 366-379.	2.7	15

#	ARTICLE	IF	CITATIONS
91	Race analysis for SystemC using model checking. , 2008, , .		15
92	Incremental bounded model checking for embedded software. Formal Aspects of Computing, 2017, 29, 911-931.	1.8	15
93	Interpolation-Based Verification of Floating-Point Programs with Abstract CDCL. Lecture Notes in Computer Science, 2013, , 412-432.	1.3	15
94	Lost in Abstraction: Monotonicity in Multi-threaded Programs. Lecture Notes in Computer Science, 2014, , 141-155.	1.3	15
95	Computing Binary Combinatorial Gray Codes Via Exhaustive Search With SAT Solvers. IEEE Transactions on Information Theory, 2008, 54, 1819-1823.	2.4	14
96	Verification and falsification of programs with loops using predicate abstraction. Formal Aspects of Computing, 2010, 22, 105-128.	1.8	14
97	Under-approximating loops in C programs for fast counterexample detection. Formal Methods in System Design, 2015, 47, 75-92.	0.8	14
98	Structural Test Coverage Criteria for Deep Neural Networks. , 2019, , .		14
99	JBMC: Bounded Model Checking for Java Bytecode. Lecture Notes in Computer Science, 2019, , 219-223.	1.3	14
100	Abstract satisfaction. , 2014, , .		13
101	Successful Use of Incremental BMC in the Automotive Industry. Lecture Notes in Computer Science, 2015, , 62-77.	1.3	13
102	Learning the Language of Error. Lecture Notes in Computer Science, 2015, , 114-130.	1.3	13
103	Approximation Refinement for Interpolation-Based Model Checking. , 2008, , 68-82.		13
104	Computing Over-Approximations with Bounded Model Checking. Electronic Notes in Theoretical Computer Science, 2006, 144, 79-92.	0.9	12
105	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		12
106	Using Program Synthesis for Program Analysis. Lecture Notes in Computer Science, 2015, , 483-498.	1.3	12
107	Bit-Precise Procedure-Modular Termination Analysis. ACM Transactions on Programming Languages and Systems, 2018, 40, 1-38.	2.1	12
108	Soundness of Data Flow Analyses for Weak Memory Models. Lecture Notes in Computer Science, 2011, , 272-288.	1.3	12

#	ARTICLE	IF	CITATIONS
109	Chaining Test Cases for Reactive System Testing. Lecture Notes in Computer Science, 2013, , 133-148.	1.3	11
110	Interpolating Quantifier-Free Presburger Arithmetic. Lecture Notes in Computer Science, 2010, , 489-503.	1.3	11
111	v2c " A Verilog to C Translator. Lecture Notes in Computer Science, 2016, , 580-586.	1.3	11
112	Periodic Orbits and Equilibria in Glass Models for Gene Regulatory Networks. IEEE Transactions on Information Theory, 2010, 56, 805-820.	2.4	10
113	Abstract conflict driven learning. ACM SIGPLAN Notices, 2013, 48, 143-154.	0.2	10
114	Effective Verification of Low-Level Software with Nested Interrupts. , 2015, , .		10
115	Learning Concise Models from Long Execution Traces. , 2020, , .		10
116	Faster Linearizability Checking via P-Compositionality. Lecture Notes in Computer Science, 2015, , 50-65.	1.3	10
117	An Abstract Interpretation of DPLL(T). Lecture Notes in Computer Science, 2013, , 455-475.	1.3	10
118	Unbounded-Time Analysis of Guarded LTI Systems with Inputs by Abstract Acceleration. Lecture Notes in Computer Science, 2015, , 312-331.	1.3	10
119	Sound and Automated Synthesis of Digital Stabilizing Controllers for Continuous Plants. , 2017, , .		10
120	Context-aware counter abstraction. Formal Methods in System Design, 2010, 36, 223-245.	0.8	9
121	Danger Invariants. Lecture Notes in Computer Science, 2016, , 182-198.	1.3	9
122	Coverage in interpolation-based model checking. , 2010, , .		9
123	A SAT-based algorithm for reparameterization in symbolic simulation. , 2004, , .		8
124	Over-Approximating Boolean Programs with Unbounded Thread Creation. , 2006, , .		8
125	Verification of Boolean programs with unbounded thread creation. Theoretical Computer Science, 2007, 388, 227-242.	0.9	8
126	Accelerating invariant generation. , 2015, , .		8



#	ARTICLE	IF	CITATIONS
127	Exposing previously undetectable faults in deep neural networks. , 2021, , .		8
128	Approximating Predicate Images for Bit-Vector Logic. Lecture Notes in Computer Science, 2006, , 242-256.	1.3	8
129	Property-Driven Fence Insertion Using Reorder Bounded Model Checking. Lecture Notes in Computer Science, 2015, , 291-307.	1.3	8
130	SAT-Based Summarization for Boolean Programs. , 2007, , 131-148.		8
131	Boom: Taking Boolean Program Model Checking One Step Further. Lecture Notes in Computer Science, 2010, , 145-149.	1.3	8
132	An Interpolating Decision Procedure for Transitive Relations with Uninterpreted Functions. Lecture Notes in Computer Science, 2011, , 150-168.	1.3	8
133	satabs: A Bit-Precise Verifier for C Programs. Lecture Notes in Computer Science, 2012, , 552-555.	1.3	8
134	Explanations for Occluded Images. , 2021, , .		8
135	Abstract Interpretation with Unfoldings. Lecture Notes in Computer Science, 2017, , 197-216.	1.3	7
136	Efficient verification of multi-property designs (The benefit of wrong assumptions). , 2018, , .		7
137	Proving Safety with Trace Automata and Bounded Model Checking. Lecture Notes in Computer Science, 2015, , 325-341.	1.3	7
138	Automatic Generation of Propagation Complete SAT Encodings. Lecture Notes in Computer Science, 2016, , 536-556.	1.3	7
139	An efficient SAT encoding of circuit codes. , 2008, , .		6
140	SCRATCH. , 2011, , .		6
141	Equivalence Checking Using Trace Partitioning. , 2015, , .		6
142	Static Program Analysis for Identifying Energy Bugs in Graphics-Intensive Mobile Apps. , 2016, , .		6
143	Generating test case chains for reactive systems. International Journal on Software Tools for Technology Transfer, 2016, 18, 319-334.	1.9	6
144	Verifying digital systems with MATLAB. , 2017, , .		6

#	ARTICLE	IF	CITATIONS
145	Effective Verification for Low-Level Software with Competing Interrupts. Transactions on Embedded Computing Systems, 2018, 17, 1-26.	2.9	6
146	Program Synthesis for Program Analysis. ACM Transactions on Programming Languages and Systems, 2018, 40, 1-45.	2.1	6
147	VCEGAR: Verilog CounterExample Guided Abstraction Refinement. , 2007, , 583-586.		6
148	Strengthening Induction-Based Race Checking with Lightweight Static Analysis. Lecture Notes in Computer Science, 2011, , 169-183.	1.3	6
149	Image Computation and Predicate Refinement for RTL Verilog using Word Level Proofs. , 2007, , .		5
150	Towards a Classification of Hamiltonian Cycles in the 6-Cube. Journal of Satisfiability, Boolean Modeling and Computation, 2008, 4, 57-74.	1.2	5
151	Strengthening properties using abstraction refinement. , 2009, , .		5
152	Loopfrog: A Static Analyzer for ANSI-C Programs. , 2009, , .		5
153	Functional Requirements-Based Automated Testing for Avionics. , 2017, , .		5
154	Optimising Spectrum Based Fault Localisation for Single Fault Programs Using Specifications. Lecture Notes in Computer Science, 2018, , 246-263.	1.3	5
155	Automating Software Analysis at Large Scale. Lecture Notes in Computer Science, 2014, , 30-39.	1.3	5
156	An Algebraic Algorithm for the Identification of Glass Networks with Periodic Orbits Along Cyclic Attractors. Lecture Notes in Computer Science, 2007, , 140-154.	1.3	5
157	Fixed points for multi-cycle path detection. , 2009, , .		4
158	A framework for Satisfiability Modulo Theories. Formal Aspects of Computing, 2009, 21, 485-494.	1.8	4
159	Computing Mutation Coverage in Interpolation-Based Model Checking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31, 765-778.	2.7	4
160	DSSynth: An automated digital controller synthesis tool for physical plants. , 2017, , .		4
161	Automated formal synthesis of provably safe digital controllers for continuous plants. Acta Informatica, 2020, 57, 223-244.	0.5	4
162	Equivalence Checking of a Floating-Point Unit Against a High-Level C Model. Lecture Notes in Computer Science, 2016, , 551-558.	1.3	4

#	ARTICLE	IF	CITATIONS
163	Formalizing and Checking Thread Refinement for Data-Race-Free Execution Models. Lecture Notes in Computer Science, 2016, , 515-530.	1.3	4
164	Formal verification at higher levels of abstraction. IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, 2007, , .	0.0	3
165	Lost in abstraction: Monotonicity in multi-threaded programs. Information and Computation, 2017, 252, 30-47.	0.7	3
166	Sound Numerical Computations in Abstract Acceleration. Lecture Notes in Computer Science, 2017, , 38-60.	1.3	3
167	Unbounded-Time Safety Verification of Guarded LTI Models with Inputs by Abstract Acceleration. Journal of Automated Reasoning, 2021, 65, 157-203.	1.4	3
168	Probabilistic Fault Localisation. Lecture Notes in Computer Science, 2016, , 65-81.	1.3	3
169	Wolverine: Battling Bugs with Interpolants. Lecture Notes in Computer Science, 2012, , 556-558.	1.3	3
170	Automated Verification of Concurrent Software. Lecture Notes in Computer Science, 2013, , 19-20.	1.3	3
171	Propositional Reasoning about Safety and Termination of Heap-Manipulating Programs. Lecture Notes in Computer Science, 2015, , 661-684.	1.3	3
172	Speeding Up Simulation of SystemC Using Model Checking. Lecture Notes in Computer Science, 2009, , 1-16.	1.3	3
173	Proving Reachability Using FShell. Lecture Notes in Computer Science, 2012, , 538-541.	1.3	3
174	Lifting CDCL to Template-Based Abstract Domains for Program Verification. Lecture Notes in Computer Science, 2017, , 307-326.	1.3	3
175	Equality Logic and Uninterpreted Functions. , 2008, , 59-80.		3
176	Abstraction of Syntax. Lecture Notes in Computer Science, 2013, , 396-413.	1.3	2
177	Independence Abstractions and Models of Concurrency. Lecture Notes in Computer Science, 2017, , 151-168.	1.3	2
178	Satisfiability checking and symbolic computation. ACM Communications in Computer Algebra, 2017, 50, 145-147.	0.4	2
179	Verification of tree-based hierarchical read-copy update in the Linux kernel. , 2018, , .		2
180	Tightening Test Coverage Metrics: A Case Study in Equivalence Checking Using k-Induction. Lecture Notes in Computer Science, 2011, , 297-315.	1.3	2

#	ARTICLE	IF	CITATIONS
181	Model and Proof Generation for Heap-Manipulating Programs. Lecture Notes in Computer Science, 2014, , 432-452.	1.3	2
182	From AgentSpeak to C for Safety Considerations in Unmanned Aerial Vehicles. Lecture Notes in Computer Science, 2015, , 69-81.	1.3	2
183	Unbounded Safety Verification for Hardware Using Software Analyzers. , 2016, , .		2
184	SCRATCH. ACM SIGPLAN Notices, 2011, 46, 311-312.	0.2	1
185	Verifying Synchronous Reactive Systems using Lazy Abstraction. , 2015, , .		1
186	Assisted Coverage Closure. Lecture Notes in Computer Science, 2016, , 49-64.	1.3	1
187	Preface: Special Issue on Interpolation. Journal of Automated Reasoning, 2016, 57, 1-2.	1.4	1
188	DSValidator. , 2018, , .		1
189	Finding Lean Induced Cycles in Binary Hypercubes. Lecture Notes in Computer Science, 2009, , 18-31.	1.3	1
190	Abstract satisfaction. ACM SIGPLAN Notices, 2014, 49, 139-150.	0.2	1
191	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		1
192	Embedded software verification: Challenges and solutions. , 2008, , .		0
193	Verified software: theories, tools and experiments. International Journal on Software Tools for Technology Transfer, 2010, 12, 405-408.	1.9	0
194	A visual studio plug-in for CProver. , 2013, , .		0
195	Preface to the special issue "SI: Satisfiability Modulo Theories": Formal Methods in System Design, 2013, 42, 1-2.	0.8	0
196	Camera-laser projector stereo system based anti-collision system for robotic wheelchair users with cognitive impairment. , 2014, , .		0
197	The virtues of conflict. , 2016, , .		0
198	Decision Procedures for the Grand Challenge. Lecture Notes in Computer Science, 2008, , 428-437.	1.3	0

#	ARTICLE	IF	CITATIONS
199	The virtues of conflict. ACM SIGPLAN Notices, 2016, 51, 1-12.	0.2	0
200	Active Learning of Abstract System Models from Traces using Model Checking. , 2022, , .		0