

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

168829

31
h-index

139680

61
g-index

211
all docs

211
docs citations

211
times ranked

2116
citing authors

#	ARTICLE	IF	CITATIONS
1	Active Learning of Abstract System Models from Traces using Model Checking. , 2022, , .		0
2	Unbounded-Time Safety Verification of Guarded LTI Models with Inputs by Abstract Acceleration. Journal of Automated Reasoning, 2021, 65, 157-203.	1.1	3
3	Exposing previously undetectable faults in deep neural networks. , 2021, , .		8
4	Explanations for Occluded Images. , 2021, , .		8
5	Automated formal synthesis of provably safe digital controllers for continuous plants. Acta Informatica, 2020, 57, 223-244.	0.5	4
6	Learning Concise Models from Long Execution Traces. , 2020, , .		10
7	A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review, 2020, 37, 100270.	10.2	203
8	Explaining Image Classifiers Using Statistical Fault Localization. Lecture Notes in Computer Science, 2020, , 391-406.	1.0	21
9	Structural Test Coverage Criteria for Deep Neural Networks. Transactions on Embedded Computing Systems, 2019, 18, 1-23.	2.1	52
10	DeepConcolic: Testing and Debugging Deep Neural Networks. , 2019, , .		32
11	Structural Test Coverage Criteria for Deep Neural Networks. , 2019, , .		14
12	JBMC: Bounded Model Checking for Java Bytecode. Lecture Notes in Computer Science, 2019, , 219-223.	1.0	14
13	Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the Hamming Distance. , 2019, , .		47
14	Gollum. , 2019, , .		24
15	Effective Verification for Low-Level Software with Competing Interrupts. Transactions on Embedded Computing Systems, 2018, 17, 1-26.	2.1	6
16	Bit-Precise Procedure-Modular Termination Analysis. ACM Transactions on Programming Languages and Systems, 2018, 40, 1-38.	1.7	12
17	DSValidator. , 2018, , .		1
18	Concolic testing for deep neural networks. , 2018, , .		190

#	ARTICLE	IF	CITATIONS
19	Program Synthesis for Program Analysis. ACM Transactions on Programming Languages and Systems, 2018, 40, 1-45.	1.7	6
20	Optimising Spectrum Based Fault Localisation for Single Fault Programs Using Specifications. Lecture Notes in Computer Science, 2018, , 246-263.	1.0	5
21	Efficient verification of multi-property designs (The benefit of wrong assumptions). , 2018, , .		7
22	Verification of tree-based hierarchical read-copy update in the Linux kernel. , 2018, , .		2
23	Model Checking Boot Code from AWS Data Centers. Lecture Notes in Computer Science, 2018, , 467-486.	1.0	16
24	JBMC: A Bounded Model Checking Tool for Verifying Java Bytecode. Lecture Notes in Computer Science, 2018, , 183-190.	1.0	42
25	Counterexample Guided Inductive Synthesis Modulo Theories. Lecture Notes in Computer Science, 2018, , 270-288.	1.0	33
26	Lost in abstraction: Monotonicity in multi-threaded programs. Information and Computation, 2017, 252, 30-47.	0.5	3
27	Independence Abstractions and Models of Concurrency. Lecture Notes in Computer Science, 2017, , 151-168.	1.0	2
28	Don't Sit on the Fence. ACM Transactions on Programming Languages and Systems, 2017, 39, 1-38.	1.7	16
29	Satisfiability checking and symbolic computation. ACM Communications in Computer Algebra, 2017, 50, 145-147.	0.2	2
30	Incremental bounded model checking for embedded software. Formal Aspects of Computing, 2017, 29, 911-931.	1.4	15
31	Automated Formal Synthesis of Digital Controllers for State-Space Physical Plants. Lecture Notes in Computer Science, 2017, , 462-482.	1.0	19
32	Program synthesis: challenges and opportunities. Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences, 2017, 375, 20150403.	1.6	19
33	Sound Numerical Computations in Abstract Acceleration. Lecture Notes in Computer Science, 2017, , 38-60.	1.0	3
34	Formal Techniques for Effective Co-verification of Hardware/Software Co-designs. , 2017, , .		22
35	Abstract Interpretation with Unfoldings. Lecture Notes in Computer Science, 2017, , 197-216.	1.0	7
36	Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs. ACM Transactions on Programming Languages and Systems, 2017, 39, 1-27.	1.7	22

#	ARTICLE	IF	CITATIONS
37	Verifying digital systems with MATLAB. , 2017, , .		6
38	Functional Requirements-Based Automated Testing for Avionics. , 2017, , .		5
39	DSSynth: An automated digital controller synthesis tool for physical plants. , 2017, , .		4
40	Modular Demand-Driven Analysis of Semantic Difference for Program Versions. Lecture Notes in Computer Science, 2017, , 405-427.	1.0	16
41	Sound and Automated Synthesis of Digital Stabilizing Controllers for Continuous Plants. , 2017, , .		10
42	Lifting CDCL to Template-Based Abstract Domains for Program Verification. Lecture Notes in Computer Science, 2017, , 307-326.	1.0	3
43	Static Program Analysis for Identifying Energy Bugs in Graphics-Intensive Mobile Apps. , 2016, , .		6
44	Sound static deadlock analysis for C/Pthreads. , 2016, , .		17
45	Assisted Coverage Closure. Lecture Notes in Computer Science, 2016, , 49-64.	1.0	1
46	Decision Procedures. Texts in Theoretical Computer Science, 2016, , .	0.5	68
47	Preface: Special Issue on Interpolation. Journal of Automated Reasoning, 2016, 57, 1-2.	1.1	1
48	The virtues of conflict. , 2016, , .		0
49	Generating test case chains for reactive systems. International Journal on Software Tools for Technology Transfer, 2016, 18, 319-334.	1.7	6
50	SC^2 : Satisfiability Checking Meets Symbolic Computation. Lecture Notes in Computer Science, 2016, , 28-43.	1.0	17
51	Danger Invariants. Lecture Notes in Computer Science, 2016, , 182-198.	1.0	9
52	Equivalence Checking of a Floating-Point Unit Against a High-Level C Model. Lecture Notes in Computer Science, 2016, , 551-558.	1.0	4
53	Probabilistic Fault Localisation. Lecture Notes in Computer Science, 2016, , 65-81.	1.0	3
54	Automatic Generation of Propagation Complete SAT Encodings. Lecture Notes in Computer Science, 2016, , 536-556.	1.0	7

#	ARTICLE	IF	CITATIONS
55	Formalizing and Checking Thread Refinement for Data-Race-Free Execution Models. Lecture Notes in Computer Science, 2016, , 515-530.	1.0	4
56	2LS for Program Analysis. Lecture Notes in Computer Science, 2016, , 905-907.	1.0	24
57	v2c – A Verilog to C Translator. Lecture Notes in Computer Science, 2016, , 580-586.	1.0	11
58	Unbounded Safety Verification for Hardware Using Software Analyzers. , 2016, , .		2
59	The virtues of conflict. ACM SIGPLAN Notices, 2016, 51, 1-12.	0.2	0
60	Verifying Synchronous Reactive Systems using Lazy Abstraction. , 2015, , .		1
61	Accelerating invariant generation. , 2015, , .		8
62	Effective Verification of Low-Level Software with Nested Interrupts. , 2015, , .		10
63	Using Program Synthesis for Program Analysis. Lecture Notes in Computer Science, 2015, , 483-498.	1.0	12
64	Synthesising Interprocedural Bit-Precise Termination Proofs (T). , 2015, , .		20
65	Hardware Verification Using Software Analyzers. , 2015, , .		30
66	Equivalence Checking Using Trace Partitioning. , 2015, , .		6
67	Under-approximating loops in C programs for fast counterexample detection. Formal Methods in System Design, 2015, 47, 75-92.	0.9	14
68	Faster Linearizability Checking via P-Compositionality. Lecture Notes in Computer Science, 2015, , 50-65.	1.0	10
69	Property-Driven Fence Insertion Using Reorder Bounded Model Checking. Lecture Notes in Computer Science, 2015, , 291-307.	1.0	8
70	Proving Safety with Trace Automata and Bounded Model Checking. Lecture Notes in Computer Science, 2015, , 325-341.	1.0	7
71	Successful Use of Incremental BMC in the Automotive Industry. Lecture Notes in Computer Science, 2015, , 62-77.	1.0	13
72	Learning the Language of Error. Lecture Notes in Computer Science, 2015, , 114-130.	1.0	13

#	ARTICLE	IF	CITATIONS
73	Propositional Reasoning about Safety and Termination of Heap-Manipulating Programs. Lecture Notes in Computer Science, 2015, , 661-684.	1.0	3
74	Unrestricted Termination and Non-termination Arguments for Bit-Vector Programs. Lecture Notes in Computer Science, 2015, , 183-204.	1.0	16
75	Evaluation of Measures for Statistical Fault Localisation and an Optimising Scheme. Lecture Notes in Computer Science, 2015, , 115-129.	1.0	18
76	Unbounded-Time Analysis of Guarded LTI Systems with Inputs by Abstract Acceleration. Lecture Notes in Computer Science, 2015, , 312-331.	1.0	10
77	Safety Verification and Refutation by k -Invariants and k -Induction. Lecture Notes in Computer Science, 2015, , 145-161.	1.0	36
78	From AgentSpeak to C for Safety Considerations in Unmanned Aerial Vehicles. Lecture Notes in Computer Science, 2015, , 69-81.	1.0	2
79	Camera-laser projector stereo system based anti-collision system for robotic wheelchair users with cognitive impairment. , 2014, , .		0
80	A Widening Approach to Multithreaded Program Verification. ACM Transactions on Programming Languages and Systems, 2014, 36, 1-29.	1.7	27
81	Accelerated test execution using GPUs. , 2014, , .		16
82	Deciding floating-point logic with abstract conflict driven clause learning. Formal Methods in System Design, 2014, 45, 213-245.	0.9	39
83	Abstract satisfaction. , 2014, , .		13
84	Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs. Lecture Notes in Computer Science, 2014, , 263-278.	1.0	23
85	Don't Sit on the Fence. Lecture Notes in Computer Science, 2014, , 508-524.	1.0	31
86	Automating Software Analysis at Large Scale. Lecture Notes in Computer Science, 2014, , 30-39.	1.0	5
87	CBMC â€“ C Bounded Model Checker. Lecture Notes in Computer Science, 2014, , 389-391.	1.0	198
88	Lost in Abstraction: Monotonicity in Multi-threaded Programs. Lecture Notes in Computer Science, 2014, , 141-155.	1.0	15
89	Abstract satisfaction. ACM SIGPLAN Notices, 2014, 49, 139-150.	0.2	1
90	Model and Proof Generation for Heap-Manipulating Programs. Lecture Notes in Computer Science, 2014, , 432-452.	1.0	2

#	ARTICLE	IF	CITATIONS
91	Ranking function synthesis for bit-vector relations. Formal Methods in System Design, 2013, 43, 93-120.	0.9	18
92	Loop summarization using state and transition invariants. Formal Methods in System Design, 2013, 42, 221-261.	0.9	17
93	A visual studio plug-in for CProver. , 2013, , .		0
94	Formal co-validation of low-level hardware/software interfaces. , 2013, , .		26
95	Verifying multi-threaded software with impact. , 2013, , .		30
96	Preface to the special issue "SI: Satisfiability Modulo Theories": Formal Methods in System Design, 2013, 42, 1-2.	0.9	0
97	Abstraction of Syntax. Lecture Notes in Computer Science, 2013, , 396-413.	1.0	2
98	Abstract conflict driven learning. , 2013, , .		17
99	Chaining Test Cases for Reactive System Testing. Lecture Notes in Computer Science, 2013, , 133-148.	1.0	11
100	Abstract conflict driven learning. ACM SIGPLAN Notices, 2013, 48, 143-154.	0.2	10
101	An Abstract Interpretation of DPLL(T). Lecture Notes in Computer Science, 2013, , 455-475.	1.0	10
102	Counterexample-Guided Precondition Inference. Lecture Notes in Computer Science, 2013, , 451-471.	1.0	18
103	Software Verification for Weak Memory via Program Transformation. Lecture Notes in Computer Science, 2013, , 512-532.	1.0	89
104	Interpolation-Based Verification of Floating-Point Programs with Abstract CDCL. Lecture Notes in Computer Science, 2013, , 412-432.	1.0	15
105	Under-Approximating Loops in C Programs for Fast Counterexample Detection. Lecture Notes in Computer Science, 2013, , 381-396.	1.0	18
106	Partial Orders for Efficient Bounded Model Checking of "Concurrent" Software. Lecture Notes in Computer Science, 2013, , 141-157.	1.0	101
107	Automated Verification of Concurrent Software. Lecture Notes in Computer Science, 2013, , 19-20.	1.0	3
108	Counterexample-guided abstraction refinement for symmetric concurrent programs. Formal Methods in System Design, 2012, 41, 25-44.	0.9	19

#	ARTICLE	IF	CITATIONS
109	Computing Mutation Coverage in Interpolation-Based Model Checking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31, 765-778.	1.9	4
110	satabs: A Bit-Precise Verifier for C Programs. Lecture Notes in Computer Science, 2012, , 552-555.	1.0	8
111	Wolverine: Battling Bugs with Interpolants. Lecture Notes in Computer Science, 2012, , 556-558.	1.0	3
112	Numeric Bounds Analysis with Conflict-Driven Learning. Lecture Notes in Computer Science, 2012, , 48-63.	1.0	31
113	Efficient Coverability Analysis by Proof Minimization. Lecture Notes in Computer Science, 2012, , 500-515.	1.0	19
114	Satisfiability Solvers Are Static Analysers. Lecture Notes in Computer Science, 2012, , 317-333.	1.0	17
115	Proving Reachability Using FShell. Lecture Notes in Computer Science, 2012, , 538-541.	1.0	3
116	SCRATCH. ACM SIGPLAN Notices, 2011, 46, 311-312.	0.2	1
117	Automatic analysis of DMA races using model checking and k-induction. Formal Methods in System Design, 2011, 39, 83-113.	0.9	16
118	An Interpolating Sequent Calculus for Quantifier-Free Presburger Arithmetic. Journal of Automated Reasoning, 2011, 47, 341-367.	1.1	27
119	Test-case generation for embedded simulink via formal concept analysis. , 2011, , .		43
120	SCRATCH. , 2011, , .		6
121	Strengthening Induction-Based Race Checking with Lightweight Static Analysis. Lecture Notes in Computer Science, 2011, , 169-183.	1.0	6
122	Beyond Quantifier-Free Interpolation in Extensions of Presburger Arithmetic. Lecture Notes in Computer Science, 2011, , 88-102.	1.0	17
123	An Interpolating Decision Procedure for Transitive Relations with Uninterpreted Functions. Lecture Notes in Computer Science, 2011, , 150-168.	1.0	8
124	Loop Summarization and Termination Analysis. Lecture Notes in Computer Science, 2011, , 81-95.	1.0	28
125	Symmetry-Aware Predicate Abstraction for Shared-Variable Concurrent Programs. Lecture Notes in Computer Science, 2011, , 356-371.	1.0	27
126	Linear Completeness Thresholds for Bounded Model Checking. Lecture Notes in Computer Science, 2011, , 557-572.	1.0	20

#	ARTICLE	IF	CITATIONS
127	Interpolation-Based Software Verification with Wolverine. Lecture Notes in Computer Science, 2011, , 573-578.	1.0	31
128	Software Verification Using k-Induction. Lecture Notes in Computer Science, 2011, , 351-368.	1.0	61
129	Making Software Verification Tools Really Work. Lecture Notes in Computer Science, 2011, , 28-42.	1.0	16
130	Soundness of Data Flow Analyses for Weak Memory Models. Lecture Notes in Computer Science, 2011, , 272-288.	1.0	12
131	Tightening Test Coverage Metrics: A Case Study in Equivalence Checking Using k-Induction. Lecture Notes in Computer Science, 2011, , 297-315.	1.0	2
132	Verification and falsification of programs with loops using predicate abstraction. Formal Aspects of Computing, 2010, 22, 105-128.	1.4	14
133	Verified software: theories, tools and experiments. International Journal on Software Tools for Technology Transfer, 2010, 12, 405-408.	1.7	0
134	Context-aware counter abstraction. Formal Methods in System Design, 2010, 36, 223-245.	0.9	9
135	Periodic Orbits and Equilibria in Glass Models for Gene Regulatory Networks. IEEE Transactions on Information Theory, 2010, 56, 805-820.	1.5	10
136	Race analysis for systems using model checking. ACM Transactions on Design Automation of Electronic Systems, 2010, 15, 1-32.	1.9	34
137	Interpolant Strength. Lecture Notes in Computer Science, 2010, , 129-145.	1.0	70
138	Boom: Taking Boolean Program Model Checking One Step Further. Lecture Notes in Computer Science, 2010, , 145-149.	1.0	8
139	Ranking Function Synthesis for Bit-Vector Relations. Lecture Notes in Computer Science, 2010, , 236-250.	1.0	36
140	Automatic Analysis of Scratch-Pad Memory Code for Heterogeneous Multicore Processors. Lecture Notes in Computer Science, 2010, , 280-295.	1.0	25
141	An Interpolating Sequent Calculus for Quantifier-Free Presburger Arithmetic. Lecture Notes in Computer Science, 2010, , 384-399.	1.0	32
142	Dynamic Cutoff Detection in Parameterized Concurrent Programs. Lecture Notes in Computer Science, 2010, , 645-659.	1.0	78
143	Termination Analysis with Compositional Transition Invariants. Lecture Notes in Computer Science, 2010, , 89-103.	1.0	61
144	Interpolating Quantifier-Free Presburger Arithmetic. Lecture Notes in Computer Science, 2010, , 489-503.	1.0	11

#	ARTICLE	IF	CITATIONS
145	Mutation-Based Test Case Generation for Simulink Models. Lecture Notes in Computer Science, 2010, , 208-227.	1.0	43
146	Coverage in interpolation-based model checking. , 2010, , .		9
147	Strengthening properties using abstraction refinement. , 2009, , .		5
148	Fixed points for multi-cycle path detection. , 2009, , .		4
149	An abstraction-based decision procedure for bit-vector arithmetic. International Journal on Software Tools for Technology Transfer, 2009, 11, 95-104.	1.7	17
150	A framework for Satisfiability Modulo Theories. Formal Aspects of Computing, 2009, 21, 485-494.	1.4	4
151	Mixed abstractions for floating-point arithmetic. , 2009, , .		45
152	Loopfrog: A Static Analyzer for ANSI-C Programs. , 2009, , .		5
153	Symbolic Counter Abstraction for Concurrent Software. Lecture Notes in Computer Science, 2009, , 64-78.	1.0	34
154	Speeding Up Simulation of SystemC Using Model Checking. Lecture Notes in Computer Science, 2009, , 1-16.	1.0	3
155	Finding Lean Induced Cycles in Binary Hypercubes. Lecture Notes in Computer Science, 2009, , 18-31.	1.0	1
156	Computing Binary Combinatorial Gray Codes Via Exhaustive Search With SAT Solvers. IEEE Transactions on Information Theory, 2008, 54, 1819-1823.	1.5	14
157	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.	1.9	15
158	A Survey of Automated Techniques for Formal Software Verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27, 1165-1178.	1.9	237
159	Embedded software verification: Challenges and solutions. , 2008, , .		0
160	An efficient SAT encoding of circuit codes. , 2008, , .		6
161	Race analysis for SystemC using model checking. , 2008, , .		15
162	Towards a Classification of Hamiltonian Cycles in the 6-Cube. Journal of Satisfiability, Boolean Modeling and Computation, 2008, 4, 57-74.	1.2	5

#	ARTICLE	IF	CITATIONS
163	Approximation Refinement for Interpolation-Based Model Checking. , 2008, , 68-82.		13
164	Scoot: A Tool for the Analysis of SystemC Models. , 2008, , 467-470.		25
165	Loop Summarization Using Abstract Transformers. Lecture Notes in Computer Science, 2008, , 111-125.	1.0	35
166	Decision Procedures for the Grand Challenge. Lecture Notes in Computer Science, 2008, , 428-437.	1.0	0
167	Equality Logic and Uninterpreted Functions. , 2008, , 59-80.		3
168	Image Computation and Predicate Refinement for RTL Verilog using Word Level Proofs. , 2007, , .		5
169	Verifying C++ with STL containers via predicate abstraction. , 2007, , .		20
170	Model checking concurrent linux device drivers. , 2007, , .		64
171	Formal verification at higher levels of abstraction. IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, 2007, , .	0.0	3
172	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		12
173	Verification of Boolean programs with unbounded thread creation. Theoretical Computer Science, 2007, 388, 227-242.	0.5	8
174	Deciding Bit-Vector Arithmetic with Abstraction. , 2007, , 358-372.		61
175	VCEGAR: Verilog CounterExample Guided Abstraction Refinement. , 2007, , 583-586.		6
176	SAT-Based Summarization for Boolean Programs. , 2007, , 131-148.		8
177	An Algebraic Algorithm for the Identification of Glass Networks with Periodic Orbits Along Cyclic Attractors. Lecture Notes in Computer Science, 2007, , 140-154.	1.0	5
178	Lifting Propositional Interpolants to the Word-Level. , 2007, , .		1
179	Over-Approximating Boolean Programs with Unbounded Thread Creation. , 2006, , .		8
180	Computing Over-Approximations with Bounded Model Checking. Electronic Notes in Theoretical Computer Science, 2006, 144, 79-92.	0.9	12

#	ARTICLE	IF	CITATIONS
181	Verification of SpecC using predicate abstraction. Formal Methods in System Design, 2006, 30, 5-28.	0.9	17
182	Error explanation with distance metrics. International Journal on Software Tools for Technology Transfer, 2006, 8, 229-247.	1.7	119
183	Approximating Predicate Images for Bit-Vector Logic. Lecture Notes in Computer Science, 2006, , 242-256.	1.0	8
184	Counterexamples with Loops for Predicate Abstraction. Lecture Notes in Computer Science, 2006, , 152-165.	1.0	21
185	Making the Most of BMC Counterexamples. Electronic Notes in Theoretical Computer Science, 2005, 119, 67-81.	0.9	47
186	Computational challenges in bounded model checking. International Journal on Software Tools for Technology Transfer, 2005, 7, 174-183.	1.7	27
187	Cogent: Accurate Theorem Proving for Program Verification. Lecture Notes in Computer Science, 2005, , 296-300.	1.0	40
188	SATABS: SAT-Based Predicate Abstraction for ANSI-C. Lecture Notes in Computer Science, 2005, , 570-574.	1.0	185
189	Word level predicate abstraction and refinement for verifying RTL verilog. , 2005, , .		47
190	Symbolic Model Checking for Asynchronous Boolean Programs. Lecture Notes in Computer Science, 2005, , 75-90.	1.0	34
191	A SAT-based algorithm for reparameterization in symbolic simulation. , 2004, , .		8
192	A Tool for Checking ANSI-C Programs. Lecture Notes in Computer Science, 2004, , 168-176.	1.0	851
193	Predicate Abstraction of ANSI-C Programs Using SAT. Formal Methods in System Design, 2004, 25, 105-127.	0.9	127
194	Completeness and Complexity of Bounded Model Checking. Lecture Notes in Computer Science, 2004, , 85-96.	1.0	67
195	Abstraction-Based Satisfiability Solving of Presburger Arithmetic. Lecture Notes in Computer Science, 2004, , 308-320.	1.0	26
196	Understanding Counterexamples with explain. Lecture Notes in Computer Science, 2004, , 453-456.	1.0	47
197	Hardware verification using ANSI-C programs as a reference. , 2003, , .		42
198	Behavioral consistency of C and verilog programs using bounded model checking. , 2003, , .		114

#	ARTICLE	IF	CITATIONS
199	Efficient Computation of Recurrence Diameters. Lecture Notes in Computer Science, 2003, , 298-309.	1.0	63
200	Automated pipeline design. , 2001, , .		33