Andrew W Appel

List of Publications by Year
in descending order

109 papers	4,484 citations	293460 24 h-index	206121 51 g-index
116 all docs	116 docs citations	116 times ranked	1299 citing authors

#	Article	IF	Citations
1	Coqâ€™s vibrant ecosystem for verification engineering (invited talk). , 2022, , .		3
2	Abstraction and subsumption in modular verification of C programs. Formal Methods in System Design, 2021, 58, 322-345.	0.9	4
3	Compositional optimizations for CertiCoq. , 2021, 5, 1-30.		8
4	Deriving efficient program transformations from rewrite rules. , 2021, 5, 1-29.		2
5	Connecting Higher-Order Separation Logic to a First-Order Outside World. Lecture Notes in Computer Science, 2020, , 428-455.	1.0	5
6	Verified sequential Malloc/Free. , 2020, , .		5
7	Closure conversion is safe for space. , 2019, 3, 1-29.		12
8	Abstraction and Subsumption in Modular Verification of C Programs. Lecture Notes in Computer Science, 2019, , 573-590.	1.0	6
9	VST-Floyd: A Separation Logic Tool to Verify Correctness of C Programs. Journal of Automated Reasoning, 2018, 61, 367-422.	1.1	56
10	A verified messaging system. , 2017, 1, 1-28.		17
11	Position paper: the science of deep specification. Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences, 2017, 375, 20160331.	1.6	21
12	Verified Correctness and Security of mbedTLS HMAC-DRBG. , 2017, , .		30
13	Shrink fast correctly!. , 2017, , .		6
14	Bringing Order to the Separation Logic Jungle. Lecture Notes in Computer Science, 2017, , 190-211.	1.0	10
15	Modular Verification for Computer Security. , 2016, , .		5
16	Compositional CompCert. ACM SIGPLAN Notices, 2015, 50, 275-287.	0.2	14
17	Verification of a Cryptographic Primitive. ACM Transactions on Programming Languages and Systems, 2015, 37, 1-31.	1.7	97
18	Compositional CompCert. , 2015, , .		51

#	Article	IF	Citations
19	Verification of a cryptographic primitive: SHA-256 (abstract). , 2015, , .		3
20	Portable Software Fault Isolation. , 2014, , .		11
21	Verified Compilation for Shared-Memory C. Lecture Notes in Computer Science, 2014, , 107-127.	1.0	33
22	Mostly Sound Type System Improves a Foundational Program Verifier. Lecture Notes in Computer Science, 2013, , 17-32.	1.0	3
23	Verified heap theorem prover by paramodulation. , 2012, , .		5
24	A List-Machine Benchmark for Mechanized Metatheory. Journal of Automated Reasoning, 2012, 49, 453-491.	1.1	5
25	A Certificate Infrastructure for Machine-Checked Proofs of Conditional Information Flow. Lecture Notes in Computer Science, 2012, , 369-389.	1.0	19
26	Verified heap theorem prover by paramodulation. ACM SIGPLAN Notices, 2012, 47, 3-14.	0.2	5
27	Local actions for a curry-style operational semantics. , 2011, , .		2
28	Security Seals on Voting Machines. ACM Transactions on Information and System Security, 2011, 14, 1-29.	4.5	14
29	Verified Software Toolchain. Lecture Notes in Computer Science, 2011, , 1-17.	1.0	80
30	VeriSmall: Verified Smallfoot Shape Analysis. Lecture Notes in Computer Science, 2011, , 231-246.	1.0	11
31	Formal Verification of Coalescing Graph-Coloring Register Allocation. Lecture Notes in Computer Science, 2010, , 145-164.	1.0	12
32	A theory of indirection via approximation. ACM SIGPLAN Notices, 2010, 45, 171-184.	0.2	3
33	A theory of indirection via approximation. , 2010, , .		20
34	Semantic foundations for typed assembly languages. ACM Transactions on Programming Languages and Systems, 2010, 32, 1-67.	1.7	29
35	Concurrent Separation Logic for Pipelined Parallelization. Lecture Notes in Computer Science, 2010, , 151-166.	1.0	11
36	A Logical Mix of Approximation and Separation. Lecture Notes in Computer Science, 2010, , 439-454.	1.0	0

#	Article	IF	Citations
37	A Fresh Look at Separation Algebras and Share Accounting. Lecture Notes in Computer Science, 2009, , 161-177.	1.0	86
38	Multimodal Separation Logic for Reasoning About Operational Semantics. Electronic Notes in Theoretical Computer Science, 2008, 218, 5-20.	0.9	10
39	Oracle Semantics for Concurrent Separation Logic. , 2008, , 353-367.		55
40	A very modal model of a modern, major, general type system. ACM SIGPLAN Notices, 2007, 42, 109-122.	0.2	21
41	A very modal model of a modern, major, general type system. , 2007, , .		88
42	A List-machine Benchmark for Mechanized Metatheory. Electronic Notes in Theoretical Computer Science, 2007, 174, 95-108.	0.9	4
43	Separation Logic for Small-Step cminor. Lecture Notes in Computer Science, 2007, , 5-21.	1.0	36
44	A Compositional Logic for Control Flow. Lecture Notes in Computer Science, 2005, , 80-94.	1.0	34
45	Polymorphic lemmas and definitions in $lambda$Prolog and Twelf. Theory and Practice of Logic Programming, 2004, 4, 1-39.	1.1	5
46	Real-time concurrent collection on stock multiprocessors. ACM SIGPLAN Notices, 2004, 39, 205-216.	0.2	54
47	Dependent types ensure partial correctness of theorem provers. Journal of Functional Programming, 2004, 14, 3-19.	0.5	15
48	Construction of a Semantic Model for a Typed Assembly Language. Lecture Notes in Computer Science, 2004, , 30-43.	1.0	10
49	A Trustworthy Proof Checker. Journal of Automated Reasoning, 2003, 31, 231-260.	1.1	21
50	Mechanisms for secure modular programming in Java. Software - Practice and Experience, 2003, 33, 461-480.	2.5	17
51	A provably sound TAL for back-end optimization. ACM SIGPLAN Notices, 2003, 38, 208-219.	0.2	5
52	A provably sound TAL for back-end optimization. , 2003, , .		23
53	Foundational proof checkers with small witnesses. , 2003, , .		31
54	Creating and preserving locality of java applications at allocation and garbage collection times. ACM SIGPLAN Notices, 2002, 37, 13-25.	0.2	6

#	Article	IF	Citations
55	An indexed model of recursive types for foundational proof-carrying code. ACM Transactions on Programming Languages and Systems, 2001, 23, 657-683.	1.7	222
56	Type-preserving garbage collectors. ACM SIGPLAN Notices, 2001, 36, 166-178.	0.2	1
57	Optimal spilling for CISC machines with few registers. ACM SIGPLAN Notices, 2001, 36, 243-253.	0.2	14
58	Efficient Substitution in Hoare Logic Expressions. Electronic Notes in Theoretical Computer Science, 2001, 41, 35-49.	0.9	1
59	Type-preserving garbage collectors. , 2001, , .		35
60	Efficient and safe-for-space closure conversion. ACM Transactions on Programming Languages and Systems, 2000, 22, 129-161.	1.7	22
61	Viewpoint: Technological access control interferes with noninfringing scholarship. Communications of the ACM, 2000, 43, 21-23.	3.3	2
62	SAFKASI. ACM Transactions on Software Engineering and Methodology, 2000, 9, 341-378.	4.8	106
63	A semantic model of types and machine instructions for proof-carrying code. , 2000, , .		93
64	Hierarchical modularity. ACM Transactions on Programming Languages and Systems, 1999, 21, 813-847.	1.7	38
65	SSA is functional programming. ACM SIGPLAN Notices, 1998, 33, 17-20.	0.2	122
66	Lambda-splitting. , 1997, , .		10
67	Lambda-splitting. ACM SIGPLAN Notices, 1997, 32, 112-124.	0.2	0
68	Shrinking lambda expressions in linear time. Journal of Functional Programming, 1997, 7, 515-540.	0.5	39
69	Empirical and analytic study of stack versus heap cost for languages with closures. Journal of Functional Programming, 1996, 6, 47-74.	0.5	36
70		0.2	3
71	Iterated register coalescing. ACM Transactions on Programming Languages and Systems, 1996, 18, 300-324.	1.7	197
72	Iterated register coalescing. , 1996, , .		17

#	Article	IF	Citations
73	A Debugger for Standard ML. Journal of Functional Programming, 1995, 5, 155-200.	0.5	33
74	A type-based compiler for standard ML. , 1995, , .		61
75	A type-based compiler for standard ML. ACM SIGPLAN Notices, 1995, 30, 116-129.	0.2	7
76	Cache performance of fast-allocating programs. , 1995, , .		18
77	Separate compilation for Standard ML. , 1994, , .		20
78	Axiomatic bootstrapping. ACM Transactions on Programming Languages and Systems, 1994, 16, 1699-1718.	1.7	8
79	Space-efficient closure representations. , 1994, , .		64
80	Loop headers in ?-calculus or CPS. Higher-Order and Symbolic Computation, 1994, 7, 337-343.	1.2	9
81	Unrolling lists. ACM SIGPLAN Lisp Pointers, 1994, VII, 185-195.	0.1	6
82	Space-efficient closure representations. ACM SIGPLAN Lisp Pointers, 1994, VII, 150-161.	0.1	6
83	Separate compilation for Standard ML. ACM SIGPLAN Notices, 1994, 29, 13-23.	0.2	1
84	Smartest recompilation. , 1993, , .		41
85	Special Issue on ML. Journal of Functional Programming, 1993, 3, 389-389.	0.5	1
86	A critique of Standard ML. Journal of Functional Programming, 1993, 3, 391-429.	0.5	15
87	Special Issue on ML. Journal of Functional Programming, 1992, 2, i-i.	0.5	0
88	Callee-save registers in continuation-passing style. Higher-Order and Symbolic Computation, 1992, 5, 191-221.	1.2	18
89	Is POPL mathematics or science?. ACM SIGPLAN Notices, 1992, 27, 87-89.	0.2	2
90	Debuggable concurrency extensions for standard ML. ACM SIGPLAN Notices, 1991, 26, 120-131.	0.2	5

#	Article	IF	Citations
91	Virtual memory primitives for user programs. Computer Architecture News, 1991, 19, 96-107.	2.5	1
92	Virtual memory primitives for user programs. Operating Systems Review (ACM), 1991, 25, 96-107.	1.5	0
93	Virtual memory primitives for user programs. , 1991, , .		160
94	Virtual memory primitives for user programs. ACM SIGPLAN Notices, 1991, 26, 96-107.	0.2	20
95	Standard ML of New Jersey. Lecture Notes in Computer Science, 1991, , 1-13.	1.0	111
96	An advisor for flexible working sets. Performance Evaluation Review, 1990, 18, 153-162.	0.4	0
97	A runtime system. Higher-Order and Symbolic Computation, 1990, 3, 343-380.	1.2	45
98	An advisor for flexible working sets. , 1990, , .		29
99	Debugging standard ML without reverse engineering. , 1990, , .		39
100	Simple generational garbage collection and fast allocation. Software - Practice and Experience, 1989, 19, 171-183.	2.5	234
101	Allocation without locking. Software - Practice and Experience, 1989, 19, 703-705.	2.5	6
102	Runtime tags aren't necessary. Higher-Order and Symbolic Computation, 1989, 2, 153-162.	1.2	55
103	Vectorized garbage collection. Journal of Supercomputing, 1989, 3, 151-160.	2.4	8
104	The world's fastest Scrabble program. Communications of the ACM, 1988, 31, 572-578.	3.3	47
105	Generalizations of the sethi-ullman algorithm for register allocation. Software - Practice and Experience, 1987, 17, 417-421.	2.5	14
106	Garbage collection can be faster than stack allocation. Information Processing Letters, 1987, 25, 275-279.	0.4	109
107	Semantics-directed code generation. , 1985, , .		13
108	An Efficient Program for Many-Body Simulation. SIAM Journal on Scientific and Statistical Computing, 1985, 6, 85-103.	1.5	452

#	Article	IF	Citations
109	Hoare logic. , 0, , 10-15.		6