

# Dominique Devriese

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/4245212/publications.pdf>

Version: 2024-02-01

65  
papers

823  
citations

1307594

7  
h-index

1199594

12  
g-index

67  
all docs

67  
docs citations

67  
times ranked

270  
citing authors

#	ARTICLE	IF	CITATIONS
1	Plausible sealing for gradual parametricity. , 2022, 6, 1-28.		2
2	Purity of an ST monad: full abstraction by semantically typed back-translation. , 2022, 6, 1-27.		5
3	Fully abstract from static to gradual. , 2021, 5, 1-30.		8
4	Linear capabilities for fully abstract compilation of separation-logic-verified code. Journal of Functional Programming, 2021, 31, .	0.8	4
5	StkTokens<i>: Enforcing well-bracketed control flow and stack encapsulation using linear capabilities</i>. Journal of Functional Programming, 2021, 31, .	0.8	3
6	On the semantic expressiveness of recursive types. , 2021, 5, 1-29.		5
7	CapablePtrs: Securely Compiling Partial Programs Using the Pointers-as-Capabilities Principle. , 2021, , .		7
8	Efficient and provable local capability revocation using uninitialized capabilities. , 2021, 5, 1-30.		14
9	Borrowed Capabilities: Flexibly Enforcing Revocation on a Capability Architecture. , 2021, , .		0
10	Leibniz equality is isomorphic to Martin-L�f identity, parametrically. Journal of Functional Programming, 2020, 30, .	0.8	1
11	Reasoning about a Machine with Local Capabilities. ACM Transactions on Programming Languages and Systems, 2020, 42, 1-53.	2.1	7
12	Gavial: Programming the web with multi-tier FRP. The Art Science and Engineering of Programming, 2020, 4, .	0.5	6
13	A Categorical Approach to Secure Compilation. Lecture Notes in Computer Science, 2020, , 155-179.	1.3	2
14	How to do proofs: practically proving properties about effectful programs' results (functional) Tj ETQq0 0 0 rgBT /Overlock 10 Tf 50 222		
15	Modular effects in Haskell through effect polymorphism and explicit dictionary applications: a new approach and the �VeriFast verifier as a case study. , 2019, , .		3
16	Linear capabilities for fully abstract compilation of separation-logic-verified code. , 2019, 3, 1-29.		10
17	Temporal Safety for Stack Allocated Memory on Capability Machines. , 2019, , .		5
18	StkTokens: enforcing well-bracketed control flow and stack encapsulation using linear capabilities. , 2019, 3, 1-28.		26

#	ARTICLE	IF	CITATIONS
19	Parametricity versus the universal type. , 2018, 2, 1-23.		12
20	Degrees of Relatedness. , 2018, , .		9
21	Coherent explicit dictionary application for Haskell. , 2018, , .		6
22	Proof-relevant unification: Dependent pattern matching with only the axioms of your type theory. Journal of Functional Programming, 2018, 28, .	0.8	7
23	Scalagna 0.1: towards multi-tier programming with Scala and Scala.js. , 2018, , .		4
24	Elmsvuur: A Multi-tier Version of Elm and its Time-Traveling Debugger. Lecture Notes in Computer Science, 2018, , 79-97.	1.3	0
25	Reasoning About a Machine with Local Capabilities. Lecture Notes in Computer Science, 2018, , 475-501.	1.3	12
26	Coherent explicit dictionary application for Haskell. ACM SIGPLAN Notices, 2018, 53, 81-93.	0.2	0
27	Lifting proof-relevant unification to higher dimensions. , 2017, , .		4
28	Expressive and strongly type-safe code generation. , 2017, , .		1
29	Efficient Functional Reactive Programming Through Incremental Behaviors. Lecture Notes in Computer Science, 2017, , 321-338.	1.3	2
30	FRP IoT modules as a Scala DSL. , 2017, , .		8
31	Experience Report. , 2017, , .		4
32	Parametric quantifiers for dependent type theory. , 2017, 1, 1-29.		17
33	Security Guarantees for the Execution Infrastructure of Software Applications. , 2016, , .		3
34	On Modular and Fully-Abstract Compilation. , 2016, , .		11
35	Eliminating dependent pattern matching without K. Journal of Functional Programming, 2016, 26, .	0.8	5
36	Reasoning about Object Capabilities with Logical Relations and Effect Parametricity. , 2016, , .		22

#	ARTICLE	IF	CITATIONS
37	Fully-abstract compilation by approximate back-translation. , 2016, , .		26
38	Fully-abstract compilation by approximate back-translation. ACM SIGPLAN Notices, 2016, 51, 164-177.	0.2	3
39	Unifiers as equivalences: proof-relevant unification of dependently typed data. , 2016, , .		8
40	Generating safe boundary APIs between typed EDSLs and their environments. ACM SIGPLAN Notices, 2016, 51, 31-34.	0.2	0
41	Generating safe boundary APIs between typed EDSLs and their environments. , 2015, , .		0
42	Secure multi-execution of web scripts: Theory and practice. Journal of Computer Security, 2014, 22, 469-509.	0.8	13
43	Pattern matching without K. , 2014, , .		15
44	Stateful Declassification Policies for Event-Driven Programs. , 2014, , .		29
45	Multi-Tier Functional Reactive Programming for the Web. , 2014, , .		23
46	Overlapping and Order-Independent Patterns. Lecture Notes in Computer Science, 2014, , 87-106.	1.3	16
47	Partial Type Signatures for Haskell. Lecture Notes in Computer Science, 2014, , 17-32.	1.3	4
48	Pattern matching without K. ACM SIGPLAN Notices, 2014, 49, 257-268.	0.2	6
49	Information Flow Control for Web Scripts. Lecture Notes in Computer Science, 2014, , 124-145.	1.3	0
50	Typed syntactic meta-programming. ACM SIGPLAN Notices, 2013, 48, 73-86.	0.2	3
51	Monadic abstract interpreters. , 2013, , .		19
52	Typed syntactic meta-programming. , 2013, , .		8
53	Fixing idioms. , 2013, , .		5
54	Monadic abstract interpreters. ACM SIGPLAN Notices, 2013, 48, 399-410.	0.2	6

#	ARTICLE	IF	CITATIONS
55	Finally tagless observable recursion for an abstract grammar model. Journal of Functional Programming, 2012, 22, 757-796.	0.8	10
56	PESAP: A Privacy Enhanced Social Application Platform. , 2012, , .		4
57	FlowFox. , 2012, , .		84
58	Secure Multi-Execution through Static Program Transformation. Lecture Notes in Computer Science, 2012, , 186-202.	1.3	18
59	Reactive non-interference for a browser model. , 2011, , .		35
60	On the bright side of type classes. ACM SIGPLAN Notices, 2011, 46, 143-155.	0.2	19
61	Information flow enforcement in monadic libraries. , 2011, , .		23
62	On the bright side of type classes. , 2011, , .		33
63	Explicitly Recursive Grammar Combinators. Lecture Notes in Computer Science, 2011, , 84-98.	1.3	5
64	Noninterference through Secure Multi-execution. , 2010, , .		167
65	Security and Privacy of Online Social Network Applications. , 0, , 206-221.		0