Javier Esparza

List of Publications by Citations

166 papers	3,567 citations	29 h-index	54 g-index
172 ext. papers	3,821 ext. citations	0.9 avg, IF	5.55 L-index

#	Paper	IF	Citations
166	Free Choice Petri Nets **1995**,		406
165	Reachability analysis of pushdown automata: Application to model-checking. *Lecture Notes in Computer Science*, **1997**, 135-150	0.9	270
164	Efficient Algorithms for Model Checking Pushdown Systems. *Lecture Notes in Computer Science*, **2000**, 232-247	0.9	159
163	An Improvement of McMillan's Unfolding Algorithm. *Formal Methods in System Design*, **2002**, 20, 285-310	1.4	142
162	Decidability and complexity of Petri net problems An introduction. *Lecture Notes in Computer Science*, **1998**, 374-428	0.9	125
161	Model checking using net unfoldings. *Science of Computer Programming*, **1994**, 23, 151-195	1.1	123
160	Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, **1997**, 34, 85-107	0.9	106
159	Model checking LTL with regular valuations for pushdown systems. *Information and Computation*, **2003**, 186, 355-376	0.8	81
158	Complexity results for 1-safe nets. *Theoretical Computer Science*, **1995**, 147, 117-136	1.1	77
157	A generic approach to the static analysis of concurrent programs with procedures **2003**,		72
156	A BDD-Based Model Checker for Recursive Programs. *Lecture Notes in Computer Science*, **2001**, 324-336	0.9	69
155	A Note on On-the-Fly Verification Algorithms. *Lecture Notes in Computer Science*, **2005**, 174-190	0.9	68
154	An improvement of McMillan's unfolding algorithm. *Lecture Notes in Computer Science*, **1996**, 87-106	0.9	68
153	Petri Nets, Commutative Context-Free Grammars, and Basic Parallel Processes. *Fundamenta Informaticae*, **1997**, 31, 13-25	1	60
152	Efficient algorithms for pre* and post* on interprocedural parallel flow graphs **2000**,		53
151	An Automata-Theoretic Approach to Interprocedural Data-Flow Analysis. *Lecture Notes in Computer Science*, **1999**, 14-30	0.9	49
150	A Fully Verified Executable LTL Model Checker. *Lecture Notes in Computer Science*, **2013**, 463-478	0.9	46

149	Reachability Analysis of Multithreaded Software with Asynchronous Communication. *Lecture Notes in Computer Science*, **2005**, 348-359	0.9	40
148	An SMT-Based Approach to Coverability Analysis. *Lecture Notes in Computer Science*, **2014**, 603-619	0.9	38
147	Verification of Safety Properties Using Integer Programming: Beyond the State Equation. *Formal Methods in System Design*, **2000**, 16, 159-189	1.4	35
146	A polynomial-time algorithm to decide liveness of bounded free choice nets. *Theoretical Computer Science*, **1992**, 102, 185-205	1.1	35
145	Newtonian program analysis. *Journal of the ACM*, **2010**, 57, 1-47	2	33
144	Parikh□ theorem: A simple and direct automaton construction. *Information Processing Letters*, **2011**, 111, 614-619	0.8	33
143	Symbolic Context-Bounded Analysis of Multithreaded Java Programs. *Lecture Notes in Computer Science*, **2008**, 270-287	0.9	33
142	Proving Termination of Probabilistic Programs Using Patterns. *Lecture Notes in Computer Science*, **2012**, 123-138	0.9	31
141	Deterministic Automata for the (F,G)-Fragment of LTL. *Lecture Notes in Computer Science*, **2012**, 7-22	0.9	31
140	Limit-Deterministic B□hi Automata for Linear Temporal Logic. *Lecture Notes in Computer Science*, **2016**, 312-332	0.9	30
139	On the decidability of model checking for several □calculi and Petri nets **1994**, 115-129		29
138	Decidability Issues for Petri Nets. *BRICS Report Series*, **1994**, 1,		29
137	General refinement and recursion operators for the Petri Box calculus. *Lecture Notes in Computer Science*, **1993**, 130-140	0.9	28
136	Reachability in live and safe free-choice Petri nets is NP-complete. *Theoretical Computer Science*, **1998**, 198, 211-224	1.1	27
135	Constraint-Based Analysis of Broadcast Protocols. *Lecture Notes in Computer Science*, **1999**, 50-66	0.9	26
134	An Unfolding Algorithm for Synchronous Products of Transition Systems. *Lecture Notes in Computer Science*, **1999**, 2-20	0.9	26
133	Reachability in cyclic extended free-choice systems. *Theoretical Computer Science*, **1993**, 114, 93-118	1.1	25
132	Abstraction Refinement with Craig Interpolation and Symbolic Pushdown Systems. *Lecture Notes in Computer Science*, **2006**, 489-503	0.9	25

131	Analyzing probabilistic pushdown automata. *Formal Methods in System Design*, **2013**, 43, 124-163	1.4	23
130	Petri Nets and Regular Processes. *Journal of Computer and System Sciences*, **1999**, 59, 476-503	1	23
129	From LTL to Deterministic Automata: A Safraless Compositional Approach. *Lecture Notes in Computer Science*, **2014**, 192-208	0.9	23
128	Verification of population protocols. *Acta Informatica*, **2017**, 54, 191-215	0.9	22
127	A Generic Approach to the Static Analysis of Concurrent Programs with Procedures. *International Journal of Foundations of Computer Science*, **2003**, 14, 551-582	0.6	22
126	Model-Checking LTL with Regular Valuations for Pushdown Systems. *Lecture Notes in Computer Science*, **2001**, 316-339	0.9	22
125	Model checking using net unfoldings. *Lecture Notes in Computer Science*, **1993**, 613-628	0.9	22
124	Parameterized Verification of Asynchronous Shared-Memory Systems. *Lecture Notes in Computer Science*, **2013**, 124-140	0.9	21
123	Implementing LTL model checking with net unfoldings. *Lecture Notes in Computer Science*, **2001**, 37-56	0.9	21
122	Complexity results for 1-safe nets. *Lecture Notes in Computer Science*, **1993**, 326-337	0.9	21
121	From LTL and Limit-Deterministic B□hi Automata to Deterministic Parity Automata. *Lecture Notes in Computer Science*, **2017**, 426-442	0.9	20
120	Petri nets, commutative context-free grammars, and basic parallel processes. *Lecture Notes in Computer Science*, **1995**, 221-232	0.9	20
119	Deciding finiteness of Petri nets up to bisimulation. *Lecture Notes in Computer Science*, **1996**, 478-489	0.9	20
118	On the convergence of Newton's method for monotone systems of polynomial equations **2007**,		19
117	jMoped: A Java Bytecode Checker Based on Moped. *Lecture Notes in Computer Science*, **2005**, 541-545	0.9	19
116	On Fixed Point Equations over Commutative Semirings **2007**, 296-307		19
115	On the model checking problem for branching time logics and basic parallel processes. *Lecture Notes in Computer Science*, **1995**, 353-366	0.9	19
114	An efficient automata approach to some problems on context-free grammars. *Information Processing Letters*, **2000**, 74, 221-227	0.8	18

113	Model Checking Probabilistic Pushdown Automata. *Logical Methods in Computer Science*, **2006**, 2,		18
112	From LTL to deterministic automata. *Formal Methods in System Design*, **2016**, 49, 219-271	1.4	17
111	Computing the Least Fixed Point of Positive Polynomial Systems. *SIAM Journal on Computing*, **2010**, 39, 2282-2335	1.1	17
110	The Model-Checking Kit. *Lecture Notes in Computer Science*, **2003**, 463-472	0.9	17
109	A New Unfolding Approach to LTL Model Checking. *Lecture Notes in Computer Science*, **2000**, 475-486	0.9	17
108	Traps characterize home states in free choice systems. *Theoretical Computer Science*, **1992**, 101, 161-176	1.1	16
107	A generic approach to the static analysis of concurrent programs with procedures. *ACM SIGPLAN Notices*, **2003**, 38, 62-73	0.2	16
106	One Theorem to Rule Them All **2018**,		15
105	Efficient Algorithms for Alternating Pushdown Systems with an Application to the Computation of Certificate Chains. *Lecture Notes in Computer Science*, **2006**, 141-153	0.9	15
104	Complexity of pattern-based verification for multithreaded programs **2011**,		14
103	Trapping mutual exclusion in the box calculus. *Theoretical Computer Science*, **1996**, 153, 95-128	1.1	14
102	Rabinizer: Small Deterministic Automata for LTL(F,G). *Lecture Notes in Computer Science*, **2012**, 72-76	0.9	14
101	Pattern-Based Verification for Multithreaded Programs. *ACM Transactions on Programming Languages and Systems*, **2014**, 36, 1-29	1.6	13
100	A Perfect Model for Bounded Verification **2012**,		13
99	Reduction Rules for Colored Workflow Nets. *Lecture Notes in Computer Science*, **2016**, 342-358	0.9	13
98	Parameterized Verification of Asynchronous Shared-Memory Systems. *Journal of the ACM*, **2016**, 63, 1-48	2	12
97	Synthesis rules for Petri nets, and how they lead to new results **1990**, 182-198		12
96	Operational Semantics for the Petri Box Calculus. *Lecture Notes in Computer Science*, **1994**, 210-225	0.9	12

95	Checking system properties via integer programming. *Lecture Notes in Computer Science*, **1996**, 250-264	0.9	12
94	Approximative Methods for Monotone Systems of Min-Max-Polynomial Equations. *Lecture Notes in Computer Science*, **2008**, 698-710	0.9	11
93	Newton's Method for ω-Continuous Semirings. *Lecture Notes in Computer Science*, **2008**, 14-26	0.9	11
92	Rewriting Models of Boolean Programs. *Lecture Notes in Computer Science*, **2006**, 136-150	0.9	11
91	Net Reductions for LTL Model-Checking. *Lecture Notes in Computer Science*, **2001**, 310-324	0.9	11
90	Complexity of pattern-based verification for multithreaded programs. *ACM SIGPLAN Notices*, **2011**, 46, 499-510	0.2	10
89	Verifying Probabilistic Procedural Programs. *Lecture Notes in Computer Science*, **2004**, 16-31	0.9	10
88	Existence of home states in Petri nets is decidable. *Information Processing Letters*, **2016**, 116, 423-427	0.8	10
87	An Extension of Newton's Method to ω-Continuous Semirings. *Lecture Notes in Computer Science*, **2007**, 157-168	0.9	9
86	A Brief History of Strahler Numbers. *Lecture Notes in Computer Science*, **2014**, 1-13	0.9	9
85	Model Checking Parameterized Asynchronous Shared-Memory Systems. *Lecture Notes in Computer Science*, **2015**, 67-84	0.9	9
84	Peregrine: A Tool for the Analysis of Population Protocols. *Lecture Notes in Computer Science*, **2018**, 604-611	0.9	9
83	On Negotiation as Concurrency Primitive. *Lecture Notes in Computer Science*, **2013**, 440-454	0.9	9
82	Compositional synthesis of live and bounded free choice Petri nets. *Lecture Notes in Computer Science*, **1991**, 172-187	0.9	9
81	Reachability Analysis of Synchronized PA Systems. *Electronic Notes in Theoretical Computer Science*, **2005**, 138, 153-178	0.7	8
80	Top-down synthesis of live and bounded free choice nets **1990**, 118-139		8
79	jMoped: A Test Environment for Java Programs **2007**, 164-167		8
78	On Negotiation as Concurrency Primitive II: Deterministic Cyclic Negotiations. *Lecture Notes in Computer Science*, **2014**, 258-273	0.9	8

77	Model checking LTL using constraint programming. *Lecture Notes in Computer Science*, **1997**, 1-20	0.9	8
76	Towards Efficient Verification of Population Protocols **2017**,		7
75	Polynomial analysis algorithms for free choice Probabilistic Workflow Nets. *Performance Evaluation*, **2017**, 117, 104-129	1.2	7
74	A solution to the covering problem for 1-bounded conflict-free Petri nets using Linear Programming. *Information Processing Letters*, **1992**, 41, 313-319	0.8	7
73	Parameterized Analysis of Immediate Observation Petri Nets. *Lecture Notes in Computer Science*, **2019**, 365-385	0.9	7
72	Separability in Conflict-Free Petri Nets **2006**, 1-18		7
71	Automatic Error Correction of Java Programs. *Lecture Notes in Computer Science*, **2010**, 67-81	0.9	7
70	Model Checking Procedural Programs **2018**, 541-572		7
69	Verifying Single and Multi-mutator Garbage Collectors with Owicki-Gries in Isabelle/HOL. *Lecture Notes in Computer Science*, **2000**, 619-628	0.9	7
68	Grammars as Processes. *Lecture Notes in Computer Science*, **2002**, 277-297	0.9	7
67	Learning Workflow Petri Nets. *Lecture Notes in Computer Science*, **2010**, 206-225	0.9	6
66	Checking Qualitative Liveness Properties of Replicated Systems with Stochastic Scheduling. *Lecture Notes in Computer Science*, **2020**, 372-397	0.9	6
65	Verification of Graph Transformation Systems with Context-Free Specifications. *Lecture Notes in Computer Science*, **2010**, 107-122	0.9	6
64	An effective tableau system for the linear time Ecalculus. *Lecture Notes in Computer Science*, **1996**, 98-109	0.9	6
63	Model checking of persistent Petri nets **1991**, 35-52		5
62	An Efficient Normalisation Procedure for Linear Temporal Logic and Very Weak Alternating Automata **2020**,		5
61	Structural Invariants for the Verification of Systems with Parameterized Architectures. *Lecture Notes in Computer Science*, **2020**, 228-246	0.9	5
60	FPsolve: A Generic Solver for Fixpoint Equations over Semirings. *Lecture Notes in Computer Science*, **2014**, 1-15	0.9	5

59	Minimizing Test Suites with Unfoldings of Multithreaded Programs. *Transactions on Embedded Computing Systems*, **2017**, 16, 1-24	1.8	4
58	**2015**,		4
57	Learning Workflow Petri Nets. *Fundamenta Informaticae*, **2011**, 113, 205-228	1	4
56	Zeros of the Hankel function of real order out of the principal Riemann sheet. *Journal of Computational and Applied Mathematics*, **1991**, 37, 89-99	2.4	4
55	Reachability in reversible Free Choice systems **1991**, 384-397		4
54	Locality-Based Abstractions. *Lecture Notes in Computer Science*, **2005**, 118-134	0.9	4
53	Complexity of Verification and Synthesis of Threshold Automata. *Lecture Notes in Computer Science* , **2020**, 144-160	0.9	4
52	SDSIrep: A Reputation System Based on SDSI **2008**, 501-516		4
51	Probabilistic Abstractions with Arbitrary Domains. *Lecture Notes in Computer Science*, **2011**, 334-350	0.9	4
50	Model checking parameterized asynchronous shared-memory systems. *Formal Methods in System Design*, **2017**, 50, 140-167	1.4	3
49	Distributed Markov Chains. *Lecture Notes in Computer Science*, **2015**, 117-134	0.9	3
48	Static analysis of deterministic negotiations **2017**,		3
47	A negative result on depth-first net unfoldings. *International Journal on Software Tools for Technology Transfer*, **2008**, 10, 161-166	1.3	3
46	Verification using PEP. *Lecture Notes in Computer Science*, **1996**, 591-594	0.9	3
45	A Unified Translation of Linear Temporal Logic to □Automata. *Journal of the ACM*, **2020**, 67, 1-61	2	3
44	Polynomial Analysis Algorithms for Free Choice Probabilistic Workflow Nets. *Lecture Notes in Computer Science*, **2016**, 89-104	0.9	3
43	A False History of True Concurrency: From Petri to Tools. *Lecture Notes in Computer Science*, **2010**, 180-186	0.9	3
42	Solving Fixed-Point Equations by Derivation Tree Analysis. *Lecture Notes in Computer Science*, **2011**, 19-35	0.9	3

41	The complexity of verifying population protocols. *Distributed Computing*, **2021**, 34, 133-177	1.2	3
40	Lower Bounds on the State Complexity of Population Protocols **2021**,		3
39	A Logical Viewpoint on Process-Algebraic Quotients. *Lecture Notes in Computer Science*, **1999**, 499-514	0.9	3
38	Proof-Checking Protocols Using Bisimulations. *Lecture Notes in Computer Science*, **1999**, 525-540	0.9	3
37	Shortest paths in reachability graphs. *Lecture Notes in Computer Science*, **1993**, 224-241	0.9	3
36	Space-efficient scheduling of stochastically generated tasks. *Information and Computation*, **2012**, 210, 87-110	0.8	2
35	A strongly polynomial algorithm for criticality of branching processes and consistency of stochastic context-free grammars. *Information Processing Letters*, **2013**, 113, 381-385	0.8	2
34	Derivation tree analysis for accelerated fixed-point computation. *Theoretical Computer Science*, **2011**, 412, 3226-3241	1.1	2
33	Derivation Tree Analysis for Accelerated Fixed-Point Computation. *Lecture Notes in Computer Science*,301-313	0.9	2
32	Complexity Results for 1-safe Nets. *DAIMI Report Series*, **1993**, 22,	0	2
31	Simple Representative Instantiations for Multicast Protocols. *Lecture Notes in Computer Science*, **2003**, 128-143	0.9	2
30	Computing the Expected Execution Time of Probabilistic Workflow Nets. *Lecture Notes in Computer Science*, **2019**, 154-171	0.9	2
29	Peregrine 2.0: Explaining Correctness of Population Protocols Through Stage Graphs. *Lecture Notes in Computer Science*, **2020**, 550-556	0.9	2
28	Negotiations and Petri Nets. *Lecture Notes in Computer Science*, **2016**, 203-225	0.9	2
27	Black Ninjas in the Dark **2018**,		2
26	Verification of Systems with an Infinite State Space. *Lecture Notes in Computer Science*, **2001**, 183-186	0.9	2
25	Advances in Parameterized Verification of Population Protocols. *Lecture Notes in Computer Science*, **2017**, 7-14	0.9	1
24	Unfolding Based Minimal Test Suites for Testing Multithreaded Programs **2015**,		1

23	Reactive and Proactive Diagnosis of Distributed Systems Using Net Unfoldings **2012**,		1
22	Solving Monotone Polynomial Equations. *International Federation for Information Processing*, **2008**, 285-298		1
21	Computing the Concurrency Threshold of Sound Free-Choice Workflow Nets. *Lecture Notes in Computer Science*, **2018**, 3-19	0.9	1
20	A False History of True Concurrency: From Petri to Tools. *Lecture Notes in Computer Science*, **2010**, 1-2	0.9	1
19	Negotiation Programs. *Lecture Notes in Computer Science*, **2015**, 157-178	0.9	1
18	Space-Efficient Scheduling of Stochastically Generated Tasks. *Lecture Notes in Computer Science*, **2010**, 539-550	0.9	1
17	Towards efficient verification of population protocols. *Formal Methods in System Design*, **2021**, 57, 305-342	1.4	1
16	Negotiation as concurrency primitive. *Acta Informatica*, **2019**, 56, 93-159	0.9	1
15	Computing Parameterized Invariants of Parameterized Petri Nets. *Lecture Notes in Computer Science*, **2021**, 141-163	0.9	1
14	On least fixed points of systems of positive polynomials. *ACM Communications in Computer Algebra*, **2010**, 43, 81-83	0.2	
13	Population Protocols: Beyond Runtime Analysis. *Lecture Notes in Computer Science*, **2021**, 28-51	0.9	
12	An Algebraic Approach to the Static Analysis of Concurrent Software. *Lecture Notes in Computer Science*, **2002**, 3-3	0.9	
11	Monotonic Set-Extended Prefix Rewriting and Verification of Recursive Ping-Pong Protocols. *Lecture Notes in Computer Science*, **2006**, 415-429	0.9	
10	The asynchronous committee meeting problem. *Lecture Notes in Computer Science*, **1994**, 276-287	0.9	
9	FPSOLVE: A Generic Solver for Fixpoint Equations Over Semirings. *International Journal of Foundations of Computer Science*, **2015**, 26, 805-825	0.6	
8	Stochastic Process Creation. *Lecture Notes in Computer Science*, **2009**, 24-33	0.9	
7	Analysis of Systems with Stochastic Process Creation. *Lecture Notes in Computer Science*, **2010**, 1-1	0.9	
6	Deterministic Negotiations: Concurrency for Free. *Lecture Notes in Computer Science*, **2014**, 23-31	0.9	

List of Publications

5	Message-Passing Algorithms for the Verification of Distributed Protocols. *Lecture Notes in Computer Science*, **2014**, 222-241	0.9
4	Finding Cut-Offs in Leaderless Rendez-Vous Protocols is Easy. *Lecture Notes in Computer Science*, **2021**, 42-61	0.9
3	Back to the Future: A Fresh Look at Linear Temporal Logic. *Lecture Notes in Computer Science*, **2021**, 3-13	0.9
2	Abduction of trap invariants in parameterized systems. *Electronic Proceedings in Theoretical Computer Science, EPTCS*,346, 1-17	
1	Separators in Continuous Petri Nets. *Lecture Notes in Computer Science*, **2022**, 81-100	0.9