PAUL C. ATTIE

Augusta University
Augusta, Georgia, USA
Tel: +1 706 469 2476
PATTIE@augusta.edu

PERSONAL EXPERIENCE

Personal Information

Born	July 21, 1961, Beirut, Lebanon.
Languages	Fluent in Arabic and English. Some French.

Educational Background

Ph.D., 1995	The University of Texas at Austin (Computer Science) Adviser: Professor E. Allen Emerson.
M.Sc., 1982	University College London, University of London (Computer Science).
B.A., 1981	Oxford University (Engineering Science).

Honors

Nominated twice for the Teaching Excellence Award at the American University of Beirut, 2015–16 and 2017–18.

Most downloaded paper from the *Information and Computation* website in the period October–December 2004. `http://top25.sciencedirect.com/index.php?subject_area_id=7&journal_id=08905401&cat_id=2`

National Science Foundation Faculty Early Career Development (CAREER) Award, June 1997 – June 2001.

Distinction in the national mathematics contest, 1978, United Kingdom. I was the only student in my high school to achieve this award.

Open Exhibition (a form of scholarship) upon admission to New College, Oxford University, 1977.

Work Experience

Aug. 19 – present	Professor of Computer and Cyber Sciences, Augusta University.
Sep. 17 – Jun. 19	Professor of Computer Science, American University of Beirut.
Oct. 07 – Aug. 17	Associate Professor of Computer Science, American University of Beirut.
Sep. 06 – Aug. 07	Freshman Adviser, Faculty of Arts and Sciences, American University of Beirut.
Sep. 06 – Jan. 07	Acting Chairman, Department of Computer Science, American University of Beirut.
Sep. 05 – Sept. 07	Assistant Professor of Computer Science, American University of Beirut.
Jan. 00 – Aug. 05	Assistant Professor of Computer Science, Northeastern University.
Jul. 03 – Aug. 05	Research Affiliate, MIT Computer Science and Artificial Intelligence Laboratory.
Mar. 00 – Jul. 03	Research Affiliate, MIT Laboratory for Computer Science.
Sep. 99 – Mar. 00	Visiting Scientist, MIT Laboratory for Computer Science.
Jan. 95 – Dec. 99	Assistant Professor of Computer Science, Florida International University.
Jan. 94 – Dec. 94	Instructor, Florida International University.
Jan. 90 – Jun. 93	Full-time Member of Technical Staff, Microelectronics and Computer Technology Corporation.
Mar. 86 – Dec. 89	Research Assistant, Microelectronics and Computer Technology Corporation (half-time).
Apr. 84 – Sep. 85	Research Assistant, Department of Computer Sciences, The University of Texas at Austin.
Feb. 83 – Jun. 83	Instructor, American University of Beirut, Department of Mathematics. Taught Math 209 (Digital Computer Programming) and Math 250 (Introduction to Computer Science).
Feb. 83 – Jun. 83	Programmer (half-time), American University of Beirut computer center.

PUBLICATIONS

Refereed Journal Papers

1. P.C. Attie and W.L. Cocke, "Model and Program Repair via Group Actions and Structure Unwinding," *ACM Transactions on Computational Logic (TOCL)*, Vol. 26, April 2025.
2. M. Jaber, Y. Falcone, P.C. Attie, A. Khalil, R. Hallal, A. El-Hokayem, "From Global Choreographies to Verifiable Efficient Distributed Implementations," *Journal of Logical and Algebraic Methods in Programming*, Vol. 115, October 2020.
3. P.C. Attie, S. Bensalem, M. Bozga, M. Jaber, J. Sifakis, and F. A. Zaraket, "Global and Local Deadlock Freedom in BIP," *ACM Transactions on Software Engineering Methodology (TOSEM)*, 26(3):1–48, January 2018.
4. P.C. Attie, K. Dak Al Bab, and M. Sakr, "Model and Program Repair via SAT Solving", *ACM Transactions on Embedded Computing Systems (TECS)*, 17(2):1–25, December 2017.
5. P.C. Attie, "Finite-state concurrent programs can be expressed succinctly in triple normal form," *Information Processing Letters (IPL)*, Vol. 123, July 2017, pp. 8–13.
6. P.C. Attie and N.A. Lynch, "Dynamic Input/Output Automata: a Formal and Compositional Model for Dynamic Systems," *Information and Computation*, Vol. 249, August 2016, pp. 28–75.
7. P.C. Attie, "Synthesis of large dynamic concurrent programs from dynamic specifications," *Formal Methods in System Design*, Vol. 48, June 2016, pp. 94–147.
8. P.C. Attie, E. Baranov, S. Bluidze, M. Jaber, and J. Sifakis, "A General Framework for Architecture Composability," *Formal Aspects of Computing.* Vol. 28, Issue 2, April 2016, pp. 207–231.
9. P.C. Attie, "Finite-state concurrent programs can be expressed in pairwise normal form," *Theoretical Computer Science.* Vol. 619, 14 March 2016, pp. 1—31.
10. P.C. Attie, "On the refinement of liveness properties of distributed systems," *Formal Methods in System Design*, 39(1):1–46, August 2011.
11. P.C. Attie, R. Guerraoui, P. Kouznetsov, N.A. Lynch, and S. Rajsbaum, "The impossibility of boosting distributed service resilience," *Information and Computation*, 209(6):927-950, June 2011.
12. D. Goldin, S. Smolka, P.C. Attie, and E. Sondregger, "Turing machines, transition systems, and interaction," *Information and Computation*, 194(2), November 2004, 101–128. *Most downloaded paper in the period October–December 2004.*
13. P.C. Attie, A. Arora, and E.A. Emerson, "Synthesis of fault-tolerant concurrent programs" *ACM Transactions on Programming Languages and Systems*, 26(1), January 2004, 125–185.
14. P.C. Attie, "Wait-free byzantine consensus," *Information Processing Letters*, 83(4), August 2002, 221–227.
15. P.C. Attie and E.A. Emerson, "Synthesis of concurrent systems for an atomic read/write model of computation," *ACM Transactions on Programming Languages and Systems*, 23(2), March 2001, 187–242.

16. P.C. Attie and E.A. Emerson, "Synthesis of concurrent systems with many similar processes," *ACM Transactions on Programming Languages and Systems* 20(1), January 1998, 51–115.

17. P.C. Attie, M. Singh, E.A. Emerson, A. Sheth, and M. Rusinkiewicz, "Scheduling workflows by enforcing intertask dependencies," *Distributed Systems Engineering Journal*, vol. 3, 1996.

18. P.C. Attie and C. Das, "Automating the refinement of specifications for distributed systems via syntactic transformations," *International Journal of Systems Science; Special Issue on Distributed Computing Systems*, 28(11), 1997, 1129–1144.

19. A. Arora, P.C. Attie, M. Evangelist, and M. Gouda, "Convergence of iteration systems," *Distributed Computing*, vol. 7, 1993, 43–53.

20. P.C. Attie, N. Francez, and O. Grumberg, "Fairness and hyperfairness in multiparty interactions," *Distributed Computing*, vol. 6, 1993, 245–254.

21. C. Tomlinson, P.C. Attie, P. Cannata, G. Meredith, A. Sheth, M. Singh, and D. Woelk, "Workflow support in Carnot," *IEEE Data Engineering; Special Issue on Workflow and Extended Transaction Systems*, 16(2), June 1993.

Refereed Conference Papers

1. P.C. Attie and W.L. Cocke, "Model and Program Repair via Group Actions" *Foundations of Software Science and Computation Structures: 26th International Conference, FoSSaCS 2023, April 2023, Paris, France.*

2. P.C. Attie, "Operational Annotations: A New Method for Sequential Program Verification" *The 14th NASA Formal Methods Symposium, May 2022, Pasadena, California.*

3. P.C. Attie, A. Cherri, K. Dak Al Bab, M. Sakr, and J. Saklawi, "Model and Program Repair via SAT Solving" *13th ACM-IEEE International Conference on Formal Methods and Models for System Design, (MEMOCODE), September, 2015, Austin, Texas.*

4. P.C. Attie, E. Baranov, S. Bliudze, M. Jaber, and J. Sifakis, "A General Framework for Architecture Composability" *12th International Conference on Software Engineering and Formal Methods (SEFM), September, 2014, Grenoble, France.*

5. P.C. Attie, S. Bensalem, M. Bozga, M. Jaber, J. Sifakis, and F.A. Zaraket, "An Abstract Framework for Deadlock Prevention in BIP", *2013 IFIP Joint International Conference on Formal Techniques for Distributed Systems (33rd FORTE / 15th FMOODS), June 2013, Florence, Italy.*

6. P.C. Attie, D.H. Lorenz, A. Portnova, and H. Chockler, "Behavioral Compatibility without State Explosion: A Case Study in Designing a Componentized Elevator Control System," *The 9th International Symposium on Component-Based Software Engineering (CBSE 2006)*, June 2006, Vasteras, Sweden.

7. P.C. Attie, R. Guerraoui, P. Kouznetsov, N. Lynch, and S. Rajsbaum, "The Impossibility of Boosting Distributed Service Resilience," *The 25th International Conference on Distributed Computing Systems (ICDCS'05)*, June 2005, Columbus, Ohio.

8. P.C. Attie and H. Chockler, "Efficiently Verifiable Conditions for Deadlock-freedom of Large Concurrent Programs," in *Proc. of Sixth International Conference on Verification, Model Checking and Abstract Interpretation*, Paris, France, January 2005, 465–481.

9. P.C. Attie, "On the implementation complexity of specifications of concurrent programs," in *Proc. of 17th International Symposium on Distributed Computing (DISC)*, Sorrento, Italy, October 2003. Springer Lecture Notes in Computer Science vol. 2848, 151–165.

10. P.C. Attie, A. Lahanas, and V. Tsaoussidis, "Beyond AIMD: explicit fair-share calculation," in *Proc. of Eighth IEEE Symposium On Computers And Communications (ISCC'2003)*, Antalya, Turkey, July 2003.

11. P.C. Attie and N.A. Lynch, "Dynamic input/output automata: a formal model for dynamic systems," in *Proc. of CONCUR'01, the International Conference on Concurrency Theory*, Aalborg, Denmark, August 2001, Springer Lecture Notes in Computer Science vol. 2154.

12. P.C. Attie, "Synthesis of large concurrent programs via pairwise composition," in *Proc. of CONCUR'99: 10th International Conference on Concurrency Theory*, Eindhoven, Holland, August 1999. Springer Lecture Notes in Computer Science vol. 1664.

13. P.C. Attie, "Liveness-preserving simulation relations," in *Proc. of 18th Annual ACM Symposium on the Principles of Distributed Computing (PODC)*, Atlanta, Georgia, May 1999, 63–72.

14. A. Arora, P.C. Attie, and E.A. Emerson, "Synthesis of fault-tolerant concurrent programs (extended abstract)," in *Proc. of 17th Annual ACM Symposium on the Principles of Distributed Computing (PODC)*, Puerto Vallarta, Mexico, June 1998, 173–182.

15. P.C. Attie and E.A. Emerson, "Synthesis of concurrent systems for an atomic read / atomic write model of computation (extended abstract)," in *Proc. of 15th Annual ACM Symposium on the Principles of Distributed Computing (PODC)*, Philadelphia, Pennsylvania, May 1996, 111–120.

16. S. Chen, Y. Deng, P.C. Attie, and W. Sun, "Optimal deadlock detection in distributed systems based on locally constructed wait-for-graphs," in *Proc. of 16th International Conference on Distributed Computing Systems (ICDCS)*, Hong Kong, May 1996.

17. Y. Deng, W. Du, P.C. Attie, and M. Evangelist, "A formalism for architectural modeling of concurrent real-time systems," in *Proc. of 8th International Conference on Software Engineering and Knowledge Engineering*, Lake Tahoe, Nevada, June 1996, 408–417.

18. M.P. Singh, G. Meredith, C. Tomlinson, and P.C. Attie, "An event algebra for specifying and scheduling workflows," in *Proc. of 4th International Conference on Database Systems for Advanced Applications (DASFAA)*, Singapore, April 1995.

19. P.C. Attie, M.P. Singh, A.P. Sheth, and M. Rusinkiewicz, "Specifying and enforcing intertask dependencies," in *Proc. of 19th VLDB Conference*, Dublin, Ireland, August 1993, 134–145.

20. D. Woelk, P.C. Attie, P. Cannata, G. Meredith, A.P. Sheth, M.P. Singh, and C. Tomlinson, "Task scheduling using intertask dependencies in Carnot," in *Proc. of ACM SIGMOD International Conference on Management of Data (Industrial Session)*, Washington, D.C., May 1993, 491–494.

21. P.C. Attie, N. Francez, and O. Grumberg, "Fairness and hyperfairness in multiparty interactions (extended abstract)," in *Proc. of 17th Annual ACM Symposium on the Principles of Programming Languages (POPL)*, San Francisco, California, January 1990, 292–305.

22. P.C. Attie and E.A. Emerson, "Synthesis of concurrent systems with many similar sequential processes (extended abstract)," in *Proc. of 16th Annual ACM Symposium on the Principles of Programming Languages (POPL)*, Austin Texas, January 1989, 191–201.

23. P.C. Attie, I.R. Forman, and E. Levy, "On fairness as an abstraction for the design of distributed systems," in *Proc. of 10th International Conference on Distributed Computing Systems (ICDCS)*, Paris, France, May 1990, 150–157.

24. A. Arora, P.C. Attie, M. Evangelist, and M. Gouda, "Convergence of iteration systems (extended abstract)," in *Proc. of CONCUR'90: Theories of Concurrency*, Amsterdam, Holland, August 1990. Springer Lecture Notes in Computer Science vol. 458, 70–82.

25. P.C. Attie, G. Bruns, M. Evangelist, C. Richter, and V. Shen, "Vanna: A visual environment for the design of distributed systems," in *Proc. of Tri-ADA Conference*, Pittsburgh, Pennsylvania, October 1989, 546–553.

MANUSCRIPTS AND OTHER PUBLICATIONS

Theses

"Formal methods for the synthesis of concurrent programs from temporal logic specifications," Ph.D. dissertation, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas, USA, May 1995.

"Addition of an error detecting layer to the network independent file transfer protocol," M.Sc. thesis, Department of Computer Science, University College London, University of London, London, UK, September 1982.

Workshop papers

1. P.C. Attie and H. Chockler, "Automatic Verification of Fault-Tolerant Register Emulations". *Electr. Notes Theor. Comput. Sci. 149(1): 49-60 (2006).*

2. P.C. Attie and D. Lorenz, "Correctness of Model-based Component Composition without State Explosion," in *Proc. of ECOOP 2003 Workshop on Correctness of Model-based Software Composition*, Darmstadt, Germany, July 2003.

3. T. Araragi, P.C. Attie, I. Keidar, K. Kogure, V. Luchangco, N.A. Lynch, and K. Mano, "On formal modeling of agent computations," in *Proc. of Formal Approaches to Agent-Based Systems (First International Workshop, FAABS 2000, Greenbelt, MD, USA, April 2000)*, Springer Lecture Notes in Artificial Intelligence, vol. 1871, pp. 48-62, 2000.

Brief Articles and Reviews

P.C. Attie, "Efficient formal methods for the synthesis of concurrent programs," *ACM SIGSOFT Software Engineering Notes*, 25(1), January 2000.

A. Arora and P.C. Attie, "Honoree Dijkstra says students should continually challenge," report on *Symposium on Frontiers in Computing—A tribute to Edsger W. Dijkstra, IEEE Computer*, December 1990, 100–101.

RESEARCH GRANTS AND CONTRACTS AWARDED

Sep. 05 – Dec. 07	Principal Investigator, National Science Foundation, Science of Design Program: "Design Locality: A Concept for Controlling the Design Complexity of Large Software Systems." (subcontract) This subcontract is based on the award below, and was made to accommodate my move to AUB.	$62,000
Dec. 04 – Dec. 07	Co-Principal Investigator, National Science Foundation, Science of Design Program: "Design Locality: A Concept for Controlling the Design Complexity of Large Software Systems."	$450,000
Sept. 04 – May. 05	Principal Investigator, Subcontract from VeroModo, Inc., DARPA/Air Force, STTR, Phase I, Topic No. AF04-T023: "A Framework for Modeling and Analyzing Complex Distributed Systems."	$30,000
Jun. 02 – Jun. 04	Principal Investigator, National Science Foundation, Software Engineering and Languages program: "Constructing Large Complex Systems via Tractable Pairwise Composition of Software Components."	$150,000
Jun. 97 – Jun. 01	Principal Investigator, National Science Foundation CAREER program: "Tractable methods for the synthesis of concurrent programs."	$200,000
Jun. 96 – Aug. 97	Principal Investigator, Rome Laboratory, U.S. Air Force: "A theoretical framework for specification refinement via transformations."	$19,000
Apr. 96 – Mar. 99	Co-Principal Investigator, Air Force Office of Scientific Research, U.S. Air Force: "A formal approach for the design of distributed real-time systems."	$348,000
Sep. 93 – Dec. 95	Co-Principal Investigator, Rome Laboratory, U.S. Air Force: "Intermediate architectural representations for the Knowledge-Based Software Assistant."	$400,000
Total		$1,597,000

My share of these funds was approximately $898,000.

TEACHING

At Augusta University, School of Computer and Cyber Sciences, Fall 2019 – present

Software Engineering, CSCI 4711 (Spring 19–20, undergraduate)

Review of propositional and first order logic. Hoare logic. Procedural abstraction and data abstraction. Data models. Implementation of these ideas in Java. Testing methodologies: black-box, white-box, regression, integration, and unit testing. Test adequacy criteria: statement-, branch-, and path-coverage.

Principles of Computer Programming II, CSCI 1302 (Fall 23–24, Spring 23–24; undergraduate)

Assignments, sequence, selection, iteration, and method calls. Exception handling. One-dimesional and multi-dimensional arrays. Reference types and value types. Functions that use ref, out, named, optional, and variable-length parameters. Inheritance, polymorphism, and interfaces. Input and output from/to text files stored in the filesystem. Recursion and write recursive functions.

Operating systems, CSCI 3271 (Fall 19–20, Spring 19–20, Fall 20–21, Spring 20–21, Fall 21–22, Spring 21–22, Fall 22–23, Spring 22–23, undergraduate)

Processes, concurrency, and deadlock. The operating system kernel. The memory hierarchy. Memory management and virtual memory. Input/output. Device drivers and interrupts. File systems. RAID.

Software Engineering, CSCI 7130 (Spring 20–21, Spring 22-23, Fall 24–25; graduate)

Review of propositional and first order logic. Hoare logic. Procedural abstraction and data abstraction. Implementation of these ideas in Java. Testing methodologies: black-box, white-box, regression, integration, and unit testing. Test adequacy criteria: statement-, branch-, and path-coverage. SMT and SAT solvers. Semantics of Concurrency. Temporal logic and model checking. Model repair. Principles of code documentation.

Network and Distributed Algorithm, CSCI 7350 (Spring 22–23; graduate)

Models of distributed computing systems. Reasoning about distributed algorithms. Safety and liveness. Fundamental problems and algorithms in the synchronous and asynchronous models, and in message passing and shared memory. Leader election, network search, and spanning trees in the asynchronous case. Synchronizers, logical time, replicated state machines. Stable property detection. Mutual exclusion, dining philosophers, general resource allocation.

Theory of computation, CSCI 7500 (Fall 23–24, graduate)

Mathematical Logic, Recursive function theory, Computability theory. Proof theory: Deductive systems, models, satisfiability, validity. Soundness, consistency, and completeness. Primitive recursive and general recursive functions. Recursive and recursively enumerable sets. Set theory. Axioms of ZF set theory. Countable and uncountable sets. Diagonalization. The axiom of choice. Complexity theory. The complexity classes: P, NP, coNP, and

polynomial-time reductions (review). Completeness and complete problems. Further complexity classes, including PSPACE, EXPTIME, and their complete problems. The polynomial hierarchy

At American University of Beirut, Department of Computer Science, Fall 2005 – Spring 2019

Special Topics in Theoretical Computer Science, CMPS 396T (Spring 13–14, Fall 15–16, graduate) Mathematical Logic. Set theory. Theory of computation and recursive function theory. Theory of computational complexity.

Advanced Software Engineering, CMPS 363 (Spring 11/12, Spring 12/13, Spring 13–14, graduate) Advanced methods for writing specifications. SAT and SMT solvers. Behavioral subtyping. Modular decomposition. Semantic models of interfaces. State machines. Models of concurrency. Analysis of concurrent systems using temporal logic proof and model checking. Correct-by-construction methodologies. Design patterns.

Software Graduation Project, CMPS 299 (Fall 13–14, Spring 13–14) Students specify, design, and construct a large and complex software system.

Discrete Mathematics, CMPS 211 (Fall 14–15, Summer 15–16) Elementary propositional and predicate logic, mathematical proof, induction, recursion, elementary set theory, counting, growth of functions, algorithms.

Software Engineering, CMPS 253 (Spring 05–06 Spring 06–07, Spring 07–08, Fall 07–08, Spring 08–09, Fall 10–11, Spring 10–11, Spring 11–12, Spring 12–13, undergraduate, 2 sections, except 1 section in Spring 12–13) The software life cycle. Software process models. Requirements analysis and engineering. Data models. Specification and analysis. Abstraction: procedure abstraction and data abstraction. Information hiding. Design: decomposition of a large program into modules. Testing.

Analysis of Algorithms, CMPS 356 (Fall 12–13 Fall 14–15, Fall 15–16, graduate) Algorithm design techniques: dynamic, greedy, amortized. Use of Hoare logic to reason about tree and graph algorithms. Linear programming. FFT. Cryptography. String matching. Computational geometry.

Advanced Algorithms and Data Structures, CMPS 256 (Summer 08–09, Fall 09–10, Fall 12–13 (2 sections), Summer 13–14, undergraduate). Efficiency of algorithms. Running time: worst-, average-, and best-case. Asymptotic notation: O, Ω, and Θ. Recurrence relations. Divide-and-conquer. Sorting and searching. Greedy algorithms. Amortized analysis. Graph search.

Distributed Computing, CMPS 396N (Fall 09–10, Fall 10–11, graduate). Models of distributed computation: shared memory and message passing, synchronous and asynchronous. Fault tolerance in distributed systems. Consistency, synchronization, and agreement in distributed systems. The consensus problem. Design abstractions such as logical time and component-based building blocks. Canonical problems, such as leader election, breadth-first search of a network, resource allocation, and mutual exclusion.

Introduction to Programming, CMPS 200 (Spring 07–08, Fall 08–09, Spring 08–09, Fall 13–14, Spring 15–16, undergraduate). Introduction to programming using the Java programming language. Assignment, if and while statements. Basics of object-oriented programming. Recursion.

The Logic of Programming, CMPS 283 (Fall 06–07, undergraduate) Review of propositional and predicate logic. Hoare logic, preconditions and postconditions, program specification using pre- and post-conditions. Partial and total correctness. Proofs of program correctness, loop invariants, variant functions. Formal derivation of programs from specifications. Brief introduction to temporal logic and concurrency.

Computer Aided Software Engineering—Design and Modeling Tools, CMPS 396G (Fall 06–07, graduate)

Modern methods of requirements elicitation, engineering and analysis. Tool support for requirements engineering. Modeling, prototyping and analysis. The relation of requirements engineering to the rest of the software life cycle. Problem decomposition, problem frames. Formal methods and approaches, temporal logic model checking.

Advanced Operating Systems, CMPS 372 (Spring 05–06, graduate)

Processes, concurrency, and deadlock. Primitives and models for concurrency, including semaphores, shared memory, message passing. Distributed algorithms and systems. Theoretical models, reasoning about safety and liveness. The operating system kernel. The memory hierarchy. Memory management and virtual memory. Input/output. Device drivers and interrupts. File systems. RAID. Fault-tolerance. Software layering and abstraction as a design principle for operating system software.

Computer Architecture, CMPS 255 (Fall 05–06, undergraduate)

Program execution. Translation of high level programs to assembly language. Representation of Numbers and Computer Arithmetic Digital Logic Design. CPU Performance. CPU Datapath and Control. Pipelining. The Memory Hierarchy.

Advanced Computer Architecture, CMPS 371 (Fall 05–06, graduate)

Fundamentals of computer design. Program execution. Translation of high level programs to assembly language. Technology trends. CPU performance. Defining, measuring, evaluating, and reporting performance. Quantitative principles of computer design. Amdahl's law. Instruction set principles and examples. Classifying instruction set architectures. CPU Datapath Layout and Control. Pipelining. The Memory Hierarchy.

At Northeastern University, College of Computer Science, Winter 2000 – Spring 2005

For courses taught at Northeastern University, I indicate the level (undergraduate or graduate) and the number of times that I have taught it. Northeastern switched from a quarter calendar to a semester calendar starting in Fall 2003, so I also indicate if the course is a quarter or semester course.

Analysis of Software Artifacts (graduate, semester, taught once).

Requirements and Specifications. Specification methods for transformational and reactive programs. Design and Architecture. Design abstraction and refinement. Data refinement

and control refinement. Design notations. Different kinds of analyses: Testing, Verification, Software Model checking, Abstract Interpretation. Tools for performing analysis, e.g., the PVS theorem prover, the IOA toolkit, the PROMELA notation and SPIN model checker, the SMV model checker.

Formal Methods and Software Analysis (graduate, semester, taught once). Axiomatic semantics of sequential programs. Hoare logic, invariants, specifications as precondition / post-condition pairs. Testing methodologies: black-box, white-box, regression, integration, and unit testing. Modeling of concurrent and distributed systems. Shared memory and shared event models. Specification techniques for large distributed systems. Temporal logic model checking. The state explosion problem. Computer-aided derivation and synthesis of software. Abstraction and refinement.

Software testing and verification (graduate, quarter, taught twice). Software engineering issues: modularity, interfaces, and implementations. Abstraction, encapsulation, and refinement. Data abstraction and control abstraction. Implementation of these ideas in Java. Testing methodologies: black-box, white-box, regression, integration, and unit testing. Test adequacy criteria: statement-, branch-, and path-coverage. Overview of methods based on logic: Hoare logic, Temporal logic model checking.

Distributed Computing (graduate, quarter, taught once). Models of distributed computation: shared memory and message passing, synchronous and asynchronous. Fault tolerance in distributed systems. Consistency, synchronization, and agreement in distributed systems. The consensus problem. Design abstractions such as logical time and component-based building blocks. Canonical problems, such as leader election, breadth-first search of a network, resource allocation, and mutual exclusion.

Agent-based computing (graduate, quarter, taught once). Applications of agents: enterprise applications, internet and information access, personal assistants. Architectures for agent systems. Frameworks, languages, and infrastructure. Mobile agents. World wide web issues

Algorithms (undergraduate, quarter, taught once). Efficiency of algorithms. Running time: worst-, average-, and best-case. Asymptotic notation: O, Ω, and Θ. Recurrence relations. Divide-and-conquer. Sorting and searching. Greedy algorithms. Amortized analysis. Graph search.

Algorithms and Data (undergraduate, semester, taught twice). Efficiency of algorithms. Running time: worst-, average-, and best-case. Asymptotic notation: O, Ω, and Θ. Recurrence relations. Divide-and-conquer. Sorting and searching. Greedy algorithms. Amortized analysis. Graph search. Minimum spanning trees. Huffman coding.

Analysis of Algorithms (graduate, quarter, taught three times). Correctness and efficiency of algorithms. Running time: worst-, average-, and best-case. Asymptotic notation: O, Ω and Θ. Recurrence relations. Divide-and-conquer. Sorting, selection, and searching. Dynamic programming. Greedy algorithms. Amortized analysis. Graph search. Minimum spanning trees. NP-completeness.

Analysis of Algorithms (graduate, semester, taught once). Correctness and efficiency of algorithms. Running time: worst-, average-, and best-case. Asymptotic notation: O, Ω and Θ.

Recurrence relations. Divide-and-conquer. Sorting, selection, and searching. Dynamic programming. Greedy algorithms. Amortized analysis. Graph search. Minimum spanning trees. Shortest paths. NP-completeness.

Automata theory (graduate, quarter, taught once). Finite automata, regular languages and regular expressions. Pushdown automata and context-free grammars. The Chomsky hierarchy. Turing machines.

Theory of Computation (graduate, quarter, taught twice). Turing machines. Undecidability. The halting problem. Reducibility. Complexity theory: time and space complexity. The classes P, NP, PSPACE, LOGSPACE, NLOGSPACE, EXPTIME. The canonical complete problems for NP, PSPACE, and NLOGSPACE. Savitch's theorem. Recursion theory: the equivalence of Turing machines and primitive recursive functions. The recursion theorem. Brief overview of first order logic and Kolmogorov complexity.

At Florida International University, School of Computer Science, Winter 1994 – Fall 1999

For courses taught at FIU, I indicate the level (undergraduate or graduate) and the number of times that I have taught it.

Logic for Computer Science (undergraduate, taught 6 times). Propositional and first order logic. The application of logic to the design of software. Preconditions and postconditions. Loop invariants. Hoare logic. I wrote an 80 page monograph for this course.

Distributed Processing (graduate, taught once). Computer network protocols. Direct link networks: ethernet and token-ring. Packet switched networks and routing. Internetworking and the Internet Protocol (IP). End-to-end protocols and the Transmission Control Protocol (TCP). Congestion control. High-speed networking. Models of distributed computing.

Theory of Computation II (graduate, taught once). Turing machines. Unsolvable problems: the halting problem. Reducibility. First order logic. Proof methods: natural deduction, resolution. Verification of programs.

Semantics of Programming Languages (graduate, taught twice). Semantics of sequential programs: axiomatic, denotational and operational semantics. Concurrent programming languages. Semantics of concurrency: nondeterministic interleaving models. Automatic verification: temporal logic model checking.

Thesis Supervision

At American University of Beirut, Department of Computer Science

Rabeeh Abou Ismail, "Specification Construction using SMT Solvers", in progress.

Mouhammad Sakr, "Model Repair via SAT Solving." Defended and graduated September 2014.

Rasha Abdallah, "Tractable Verification of Large Asynchronous Concurrent Programs." Defended and graduated June 2011.

Emile Shartouny, "A Modeling and Specification Verification Tool for Concurrent Programs." Defended and graduated June 2011.

At Florida International University, School of Computer Science

Bradley Rex, graduated with M.Sc., June 1999. Thesis Title: "Inference of K-process program behavior from 2-process programs."

Champak Das, graduated with M.Sc., June 1996. Thesis title: "Automating the refinement of specifications for distributed systems via syntactic transformations."

Ph.D. Committees

Mouhammad Sakr, Saarland University, 2021. Thesis title: Parameterized Verification and Repair of Concurrent Systems.

Peter Drabik, University of Pisa, 2011. Thesis title: Modular Verification of Biological Systems.

Lujun Jia, graduated Fall 2005, Northeastern University, College of Computer Science. Thesis title: Communication Structures for Ad Hoc Networks.

Chi Zhang, graduated May 2003, Northeastern University, College of Computer Science. Thesis title: Receiver-oriented and Measurement-based Transmission Control in Heterogeneous (Wired/Wireless) Networks.

Adrian Llahanas, graduated May 2003, Northeastern University, College of Computer Science. Thesis title: On the Efficiency and Fairness of AIMD-based Congestion Avoidance Algorithms

Student Advising

Number of students advised in Spring 2006 semester (American University of Beirut): 48

Number of students advised in Fall 2006 semester (American University of Beirut): 100, approximately

SERVICE

Service To The University

At Augusta University, School of Computer and Cyber Sciences, Fall 2019 – present

Chairman of School of Computer and Cyber Sciences promotion and tenure committee, 2024–25

Chairman of Augusta University faculty senate information technology resources committee, 2021–25

Member of the Augusta University faculty senate, 2021–present

School of Computer and Cyber Sciences Ph.D. Committee 2019–21

School of Computer and Cyber Sciences Hiring committee 2019–22

Chairman of School of Computer and Cyber Sciences promotion and tenure committee, 2020–21

School of Computer and Cyber Sciences Promotion and tenure committee, 2021–22

Search committee for Dean of the Pamplin College of Arts, Humanities, and Social Sciences

At American University of Beirut, Department of Computer Science, Fall 2005 – present

Committee to write the CMPS department self-study report, Academic year 13/14

Acting Chairman of Computer Science, entire Fall 2006 semester

Freshman Adviser, Academic year 06/07

Coordinator, Graduate Committee, Academic year 05/06

At Northeastern University, College of Computer Science, Winter 2000 – Spring 2004

Member, Faculty Hiring Committee, Fall 2000 – Spring 2002

Member, Graduate Committee, Fall 2000 – Spring 2004

Member, PhD admissions committee, Fall 2000 – Spring 2002

Member, MSc admissions committee, Fall 2002 – present

Member, subcommittee to redesign the MSc program, in preparation for the transition to a semester calendar, 2001

Member, 3 area subcommittees to redesign the PhD program, in preparation for the transition to a semester calendar, 2001

At Florida International University, School of Computer Science, Winter 1994 – Fall 1999

Member, Equipment Committee, Fall 1997 – Spring 1999

Member, Faculty Hiring Committee, Fall 1995 – Spring 1997

Service To The Profession

Co-Editor

Proceedings of the Fourth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), New York University, New York, January 2003.

Special section on verification, model checking, and abstract interpretation, *International Journal on Software Tools for Technology Transfer*. Selection of best papers from VMCAI 2003.

Program Committee Member

14th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE'16, Indian Institute of Technology, Kanpur, India, 2016

Formal Approaches to Parallel and Distributed Systems (4PAD) special session at the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP 2017).

35th IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems (FORTE), Grenoble, France, 2015

34th IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems (FORTE), Berlin, Germany, 2014

14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), Ad-Hoc and Sensor Networks Track, Toronto, Canada, 2012

The 25th IEEE International Conference on Distributed Computing Systems (ICDCS), Columbus, Ohio 2005

Symposium on Network Computing and Applications, Cambridge, Massachusetts, July 2005

18th International Symposium on Distributed Computing (DISC), Amsterdam, Netherlands, October 2004

Fourth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), New York University, New York, 2003

The 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), Providence, Rhode Island, 2003

The 3rd International Conference on Internet Computing, Las Vegas, 2002

11th Intl. Conference on Software Engineering and Knowledge Engineering (SEKE), Germany, 1999

18th International Conference on Distributed Computing Systems (ICDCS), Amsterdam, 1998

10th Intl. Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, 1998

2nd Intl. Conference on Cooperative Information Systems, Kiawah Island, South Carolina, 1997

National Science Foundation Proposal Review Panels

I have served on three National Science Foundation proposal review panels.

Referee (Journals)

Journal of the ACM

Information and Computation

ACM Transactions on Software Engineering and Methodology

ACM Transactions in Computational Logic

Journal of Computer and System Sciences

Formal Aspects of Computing

Logical Methods in Computer Science

Information Processing Letters

Fundamenta Informaticae

IEEE Transactions on Parallel and Distributed Systems

IEEE Transactions on Networking

IEEE Transactions on Software Engineering

IEEE Transactions on Dependable and Secure Computing

Distributed Computing

International Journal on Software Tools for Technology Transfer

Computer Networks and ISDN Journal

Formal Methods in Systems Design

The Computer Journal

Journal of Aerospace Computing, Information, and Communication

International Journal of Software Engineering and Knowledge Engineering

International Journal of Systems Science

International Journal on Cooperative Information Systems

Referee (Conferences)

ACM Symposium on Principles of Distributed Computing, 2022

Symposium on Dependable Software Engineering, 2015

Concurrency Theory, 16'th International Conference, 2005

19th International Symposium on Distributed Computing, 2005

The Dependable Computing and Communications Symposium 2005

The International Conference on Dependable Systems and Networks 2005

European Symposium on Programming 2004

Aspect Oriented Software Development 2004

International Parallel and Distributed Processing Symposium 2002, 2004

DIALM-POMC Joint Workshop on Foundations of Mobile Computing, 2003

IEEE Symposium on Computers and Communications 2003

ACM Symposium on Parallelism in Algorithms and Architectures 2002

ACM Symposium on Principles of Distributed Computing, 1989, 2000, 2001

International Conference on Distributed Computing Systems 2000

International Parallel Processing Symposium 1998

IEEE International Conference on Network Protocols 1997, 1998

International Conference on Computer Aided Verification 1990, 1994, 1995, 1996, 2000

International Conference on Cooperative Information Systems, 1996

International Conference on Software Engineering and Knowledge Engineering, 1996

International Conference on Automated Deduction, 1990

Reviewer of Proposals

National Science Foundation, Division of Computing and Communication Foundations, Formal and Mathematical Foundations cluster, 2004

National Science Foundation, Centers of Research Excellence in Science and Engineering 2003

National Science Foundation, Japan and Korea Program , 1998

National Science Foundation, Software Engineering and Languages Program (1997, 1998)

Registration chair

4th Intl. Conference on Parallel and Distributed Information Systems, Miami Beach, 1996