CURRICULUM VITAE

Daniele Gorla

Data di nascita: 12 Ottobre 1976
Web Page: `http://www.dsi.uniroma1.it/~gorla`
E-mail: `gorla@di.uniroma1.it`

Titoli di studio:

- *Maturità scientifica* con 60/60 presso il liceo "Villa Sora" di Frascati (RM) nel 1995;
- *Diploma in violino* (equiparato a *laurea triennale*, in base a decreto ministeriale del 27/9/02) con 10/10 presso il Conservatorio di Perugia nel Luglio 1999;
- *Laurea (quinquennale) in Informatica* con 110/110 e lode presso l'università di Roma "La Sapienza" nel Dicembre 2000.
- *Dottorato di Ricerca in "Informatica ed Applicazioni"*, ciclo XVII presso l'università di Firenze (supervisors: Prof. Rocco De Nicola e Prof. Rosario Pugliese), Febbraio 2005.

Posizione attuale (da Marzo 2006)**:** ricercatore presso il Dipartimento di Informatica (facoltà di Scienze Matematiche, Fisiche e Naturali) dell'Università di Roma "La Sapienza".

Aree di ricerca: semantica della programmazione concorrente; calcoli di processo con distribuzione, mobilità e/o crittografia; metodi formali per la sicurezza.

Formazione:

- *Microsoft DotNet Crash Course* tenutosi a Cambridge (UK), Marzo 2002;
- *Bertinoro International Summer School for Graduate Studies in Computer Science,* Maggio 2002;
- *Marktoberdorf International Summer School* su "Models, Algebras and Logic of Engineering Software", Agosto 2002;
- *Summer School on Software Security*, tenutasi ad Eugene (Oregon, USA), Giugno 2004.

Attività professionali:

- Contratti:
  - *Contratto di collaborazione coordinata e continuativa* con l'università di Firenze (dipartimento di Sistemi ed Informatica), per svolgere ricerca su "Calcoli per la specifica e la verifica di protocolli crittografici", Aprile – Ottobre 2001;
  - *Contratto* con l'università di Bologna (Dipartimento di Elettronica, Informatica e Sistemistica) per svolgere ricerca su "Utilizzo di teorie logiche per l'analisi di problemi di fault-tolerance in sistemi distribuiti", Luglio – Ottobre 2002;
  - *Assegno di Ricerca* presso il Dip. di Informatica dell'Università di Roma "La Sapienza" da Ottobre 2003 a Febbraio 2006;
- Periodi di ricerca svolti all'estero:
  - *Marie Curie Fellowship* presso COGS (Centre for Research in Cognitive Science) dell'Università del Sussex a Brighton (UK), Luglio – Settembre 2003;
  - *Research associate* all'università del Sussex (Dep. of Informatics), Gennaio – Aprile 2005;
  - *Visiting professor* presso il laboratorio "Preuves, Programmes et Systèmes" (PPS) dell'università "Paris Diderot" (Parigi VII) nel Febbraio 2008;
  - *Visiting Fellow* presso la Faculty of Engineering and Information Technology della University of Technology di Sydney (Australia) Settembre – Ottobre 2010;
- Presentazioni invitate:
  - *Invited speaker* al workshop **SecCo'03**, satellite di ICALP'03 (titolo del talk: "Types for Global Computing Systems");
  - *Invited lecturer* alla scuola internazionale **BASICS'09**, organizzata dall'università di Shanghai nell'Ottobre 2009;

- *Invited tutorial* al workshop **"Qualitative and Quantitative Network Protocol Analysis"**, organizzato a Dagstuhl nel Febbraio 2010;

- Membro di comitati di programma:
  - *PC co-chair* e *organizzatore* del workshop **SecCo'07: Security Issues in Concurrency**, affiliato a CONCUR'07;
  - *membro del PC* e *panelista* del workshop **EXPRESS'07: Expressiveness in Concurrency**, affiliato a CONCUR'07;
  - *membro del PC* del workshop **PLID'08: Programming Language Interference and Dependence**, affiliato a SAS'08;
  - *PC co-chair* e *organizzatore* dei workshop **EXPRESS'08** ed **EXPRESS'09**, affiliati a CONCUR'08 e CONCUR'09;
  - *membro del PC* di **MFPS XXV: the 25th Conference on Mathematical Foundations of Programming Semantics**, 2009;
  - *membro del PC* di **CONCUR 2010: the 21st Conference on Concurrency Theory**, 2010;
- Attività di Editore*:*
  - *Guest editor* per le **Electronic Notes in Theoretical Computer Science,** (atti di SecCo'07 e EXPRESS'08);
  - *Guest editor* per il **Journal of Computer Security** (special issue di SecCo'07)
  - *Guest editor* per il **Journal of Mathematical Structures in Computer Science** (special issue di EXPRESS'08 e di EXPRESS'09);
  - *Guest editor* per gli **Electronic Proceedings in Theoretical Computer Science,** (atti di EXPRESS'09);
- Attività di Revisore:
  - *riviste internazionali:* ACM TOPLAS, Information and Computation (2 articoli), Acta Informatica (2 articoli), Bullettin EATCS, Theoretical Informatics and Applications, Journal of Logic and Algebraic Programming (2 articoli), Theoretical Computer Science, International Journal of Foundations of Computer Science (2 articoli), Journal of Computer Security, Mathematical Structures in Computer Science, Nordic Journal of Computing, Journal of Computer Science and Technology, ACM Computing Surveys;
  - *conferenze e workshop internazionali:* FST&TCS'02, ACM–SAC'03, Foundations of Global Computing Workshop 2003, COORDINATION'04, CONCUR'04, FoSSaCS'05, COORDINATION'05, CSFW'05, CONCUR'05, EXPRESS'05, SECCO'05, ESOP'06, CSFW'06, COORDINATION'06, LICS'06, COORDINATION'07, CONCUR'07, EXPRESS'07 (3 articoli), SECCO'07 (2 articoli), FoSSaCS'08, COORDINATION'08, ICALP'08, ICLP'08, PerCom'09, ESOP'09, CONCUR'09 (2 articoli), ESOP'10, FoSSaCS'10, CIAC'10, IFIP-TCS'10, CONCUR'10 (3 articoli), EXPRESS'10.

Attività istituzionale (presso l'università di Roma "La Sapienza"):

- *Membro della commissione scientifica* del dipartimento di Informatica, triennio 2006-09;
- *Membro della commissione scientifica* del dipartimento di Informatica, triennio 2009-12;
- *Membro della commissione qualità* del corso di laurea specialistica in Informatica, a.a. 2006-07;
- *Membro della commissione qualità* del corso di laurea specialistica in Informatica, a.a. 2007-08;
- *Responsabile ERASMUS per l'attivazione di nuovi contatti*, dal Luglio 2007.

Didattica svolta (presso l'università di Roma "La Sapienza"):

- *Esercitazioni* del corso di "Architettura degli Elaboratori I" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale E–O), a.a. 2001/02;
- *Esercitazioni* del corso di "Architettura degli Elaboratori I" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale P–Z), a.a. 2001/02;
- *Esercitazioni* del corso di "Architettura degli Elaboratori I" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale E–O), a.a. 2002/03;
- *Tutoraggio* per il corso di "Laboratorio di Programmazione I" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche), a.a. 2002/03;

- *Esercitazioni* del corso di "Logica Matematica" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2003/04;
- *Tutoraggio* per il corso di "Programmazione II" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2004/05;
- *Ciclo di seminari sul pi-calcolo* per il corso di "Teoria della Concorrenza", a.a. 2004/05;
- *Esercitazioni* del corso di "Logica Matematica" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2005/06.
- *Esercitazioni* del corso di "Architettura degli Elaboratori II" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2005/06;
- *Esercitazioni* del corso di "Architettura degli Elaboratori II" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale E–O), a.a. 2005/06;
- *Ciclo di seminari su information flow* per il corso di "Tecniche di Sicurezza basate su linguaggi" (laurea specialistica in Informatica), a.a. 2005/06;
- *Ciclo di seminari sul pi-calcolo* per il corso di "Teoria della Concorrenza" (laurea specialistica in Informatica), a.a. 2005/06;
- *Esercitazioni* del corso di "Logica Matematica" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2006/07;
- *Supplenza* del corso "Teoria della Concorrenza" (laurea specialistica in Informatica), a.a. 2006/07;
- *Supplenza* del corso "Tecniche di Sicurezza basate su linguaggi" (laurea specialistica in Informatica), a.a. 2006/07;
- *Esercitazioni* del corso di "Logica Matematica" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2007/08;
- *Esercitazioni* del corso di "Architettura degli Elaboratori II" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2007/08;
- *Supplenza* del corso "Teoria della Concorrenza" (laurea specialistica in Informatica), a.a. 2007/08;
- *Supplenza* del corso "Tecniche di Sicurezza basate su linguaggi" (laurea specialistica in Informatica), a.a. 2007/08;
- *Affidamento* del corso "Teoria della Concorrenza" (laurea specialistica in Informatica), a.a. 2008/09;
- *Affidamento* del corso "Tecniche di Sicurezza basate su linguaggi" (laurea specialistica in Informatica), a.a. 2008/09;
- *Esercitazioni* del corso di "Logica Matematica" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale A–D), a.a. 2009/10;
- *Affidamento* del corso "Teoria della Concorrenza" (laurea specialistica in Informatica), a.a. 2009/10;
- *Affidamento* del corso "Metodologie di Programmazione" (1° anno, lauree triennali in Informatica e Tecnologie Informatiche, canale P – Z).

Partecipazione alle seguenti conferenze e workshops internazionali:

- *ETAPS'02*, con presentazione di un lavoro a *FoSSaCS;*
- *Types for Global Computing Workshop (Parigi, Gennaio 2003)*, con presentazione di un lavoro;
- *1st Intern. Conf. on Security in Pervasive Computing (SPC'03)*, con presentazione di un lavoro;
- *2nd EATCS Workshop on Foundations of Global Computing (FGC'03)* affiliato ad ICALP'03;
- *1st Intern.Workshop on Security Issues in Coordination Models, Languages, and Systems (SecCo'03)*, affiliato ad ICALP'03, come invited speaker;
- *ICALP'03*, con presentazione di un lavoro;
- *COORDINATION'04;*
- *19th Annual Symposium on Applied Computing (SAC'04)*, con presentazione di un lavoro;
- *17th IEEE Computer Security Foundations Workshop (CSFW'04)*, con presentazione di un lavoro;
- *3rd EATCS and 2nd UK Grand Challenge joint workshop on Foundations of Global Ubiquitous Computing (FGUC'04)*, con presentazione di un lavoro;
- *Dagstuhl Workshop su "Foundations of Global Computing"*, con presentazione di un lavoro;
- *COORDINATION'05*, con presentazione di un lavoro;
- *ICALP'05*, con presentazione di un lavoro;
- *ETAPS'06*, con presentazione di un lavoro a *FoSSaCS;*
- *EXPRESS'06*, con presentazione di un lavoro;

- *Symposium on "Emerging Trends in Concurrency";*
- *SecCo'07*, come organizzatore e co-chair;
- *EXPRESS'07*, come panelista.
- *CONCUR'08*, con presentazione di un lavoro;
- *EXPRESS'08*, come organizzatore e co-chair;
- *TGC'08*, con presentazione di un lavoro;
- *MFPS XXV*, con presentazione di un lavoro;
- *EXPRESS'09*, come organizzatore e co-chair;
- *BASICS'09*, con invited lecture;
- *Dagstuhl Workshop su "Qualitative and Quantitative Network Protocol Analysis"*, con invited tutorial.

Seminari presso Università ed Enti di Ricerca Nazionali ed Internazionali:

- *"Access Control in a Language with Distribution and Code Mobility"*, presso l'Università di Milano, Novembre 2002;
- *"A Distributed Calculus for Role-Based Access Control"*, presso il Dept. of Informatics, Univ. of Sussex (UK), Maggio 2004;
- *"Comparing communication primitives via their relative expressive power*", presso l'Ecole Polytechnique (Parigi), Marzo 2006;
- *"Comparing calculi for mobility via their relative expressive power*", presso Imperial college (Londra), l'Ecole Polytechnique (Parigi) e PPS-ParigiVII, Marzo 2007;
- *"Towards a unified approach to encodability and separation results for process calculi*", presso l'Ecole Polytechnique (Parigi) e PPS-ParigiVII, Febbraio 2008.

Coinvolto nei seguenti progetti di ricerca nazionali ed internazionali:

- *Models, Calculi and Languages for Network Aware Programming (NAPOLI),* programma di ricerca di interesse nazionale del MURST;
- *Network Aware Programming and Interoperability (NAPI)*, finanziato da Microsoft Research;
- *Mobile Calculi based on Domains (MIKADO),* finanziato dalla CEE nell'iniziativa FET-Global Computing (Contract IST-2001-32222);
- *Architectures for Mobility (AGILE),* finanziato dalla CEE nell'iniziativa FET-Global Computing (Contract IST-2001-32747);
- *Software Engineering for Service-Oriented Overlay Computers (SENSORIA),* finanziato dalla CEE, Sixth Framework Programme, Priority 2 – Information Society Technologies (Contract number 016004);
- *Quantified Information Flow for Process Algebras*, finanziato dalla Royal Society per scambi bilaterali tra "La Sapienza" (**responsabile di sede**) e il "Kings College" di Londra.

Pubblicazioni su atti di conferenze e workshop internazionali:

[C1] *"On Compositional Reasoning in the Spi–Calculus"*, assieme a Michele Boreale. Atti di **FoSSaCS 2002** (5th International Conference on Foundations of Software Science and Computation Structures), LNCS 2303, pagg. 67 – 81. Springer, 2002.

[C2] *"Resource Access and Mobility Control with Dynamic Privileges Acquisition"*, assieme a Rosario Pugliese. Atti di **ICALP 2003** (30th International Colloquium on Automata, Languages and Programming), LNCS 2719, pagg. 119 – 132. Springer, 2003.

[C3] *"Enforcing Security Policies via Types"*, assieme a Rosario Pugliese. Atti di **SPC 2003** (1st International Conference on Security in Pervasive Computing), LNCS 2802, pagg. 88 – 103. Springer, 2003.

[C4] *"Controlling Data Movement in Global Computing Applications"*, assieme a Rosario Pugliese. Atti di **SAC 2004** (19° Annual ACM-SIGAPP Symposium on Applied Computing), pagg. 1462 – 1467. ACM Press, 2004.

[C5] *"A Distributed Calculus for Role-Based Access Control"*, assieme a Chiara Braghin e Vladimiro Sassone. Atti di **CSFW 2004** (17° Computer Security Foundations Workshop), pagg. 48 – 60. IEEE Computer Society, 2004.

[C6] *"On the expressive power of KLAIM-based Calculi"*, assieme a Rocco De Nicola e Rosario Pugliese. Atti di **EXPRESS 2004** (11th Workshop on Expressiveness in Concurrency), ENTCS 128(2):117 – 130. Elsevier, 2004.

[C7] *"Security Policies as Membranes in Systems for Global Computing"*, assieme a Matthew Hennessy e Vladimiro Sassone. Atti di **FGUC 2004** (3rd EATCS Workshop on Fondations of Global Ubiquitous Computing), ENTCS 138(1):23 – 42. Elsevier, 2004.

[C8] *"Global Computing in a Dynamic Network of Tuple Spaces"*, assieme a Rocco De Nicola e Rosario Pugliese. Atti di **COORDINATION 2005** (7th Intern. Conf. on Coordination Models and Languages), LNCS 3454, pagg. 157 – 172. Springer, 2005.

[C9] *"Pattern Matching over a Dynamic Network of Tuple Spaces"*, assieme a Rocco De Nicola e Rosario Pugliese. Atti di **FMOODS 2005** (7th IFIP Intern. Conf. on Formal Methods for Open Object-based Distributed Systems), LNCS 3535, pagg. 1 – 14. Springer, 2005.

[C10] *"Basic Observables for a Calculus for Global Computing"*, assieme a Rocco De Nicola e Rosario Pugliese. Atti di **ICALP 2005** (32nd International Colloquium on Automata, Languages and Programming), LNCS 3580, pagg. 1226 – 1238. Springer, 2005.

[C11] *"On the Relative Expressive Power of Asynchronous Communication Primitives"*. Atti di **FoSSaCS 2006** (9th International Conference on Foundations of Software Science and Computation Structures), LNCS 3921, pagg. 47 – 62. Springer, 2006.

[C12] *"Inferring Dynamic Credentials for Role-based Trust Management"*, assieme a M. Hennessy e V. Sassone. Atti di **PPDP 2006** (8th ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming), pagg. 213 – 224. ACM Press, 2006.

[C13] *"Synchrony vs Asynchrony in Communication Primitives"*. Atti di **EXPRESS 2006** (13th Workshop on Expressiveness in Concurrency), ENTCS 175(3): 87 – 108. Elsevier, 2007.

[C14] *"From Flow Logic to Static Type Systems for Coordination Languages"*, assieme a R.De Nicola, R.R.Hansen, F.Nielson, H.Riis Nielson, C.W. Probst e R.Pugliese. Atti di **COORDINATION 2008** (10th Intern. Conf. on Coordination Models and Languages), LNCS 5052, pagg. 100 – 116. Springer, 2008.

[C15] *"Network Applications of Graph Bisimulation"*, assieme a Pietro Cenciarelli ed Emilio Tuosto. Atti di **IGCT 2008** (4th Intern. Conf. On Graph Transformation), LNCS 5214, pagg. 131 – 146. Springer, 2008.

[C16] *"Towards a Unified Approach to Encodability and Separation Results for Process Calculi"*. Atti di **CONCUR 2008** (19th Intern. Conf. on Concurrency Theory), LNCS 5201, pagg. 492 – 507. Springer, 2008.

[C17] *"On the Relative Expressive Power of Ambient-based Calculi"*. Atti di **TGC 2008** (4th Intern. Symp. on Trustworthy Global Computing), LNCS 5474, pagg. 141 – 156. Springer, 2009.

[C18] *"On the Relative Expressive Power of Calculi for Mobility"*. Atti di **MFPS XXV** (25th Intern. Conf. on Mathematical Foundations of Programming Semantics), ENTCS 249, pagg. 269 – 286. Elsevier, 2009.

[C19] *"Depletable Channels: Dynamics and Behaviour"*, assieme a Pietro Cenciarelli ed Ivano Salvo. Atti di **FCT 2009** (17th Intern. Symp. on Fundamentals of Computation Theory), LNCS 5699, pagg. 50 – 61. Springer, 2009.

[C20] *"Concurrent Pattern Calculus"*, assieme a Thomas Given-Wilson e Barry Jay. Atti di **IFIP-TCS 2010** (6th Intern. IFIP Conference on Theoretical Computer Science), IFIP AICT 323, pagg. 244 – 258. IFIP, 2010.

[C21] *"A semiring-based trace semantics for processes with applications to information leakage analysis"*, assieme a Michele Boreale e David Clark. Atti di **IFIP-TCS 2010** (6th Intern. IFIP Conference on Theoretical Computer Science), IFIP AICT 323, pagg. 340 – 354. IFIP, 2010.

Pubblicazioni su libri e raccolte:

[B1] *"The KLAIM Project: Theory and Practice"*, assieme a L.Bettini, V.Bono, R.De Nicola, G.Ferrari, M.Loreti, E.Moggi, R.Pugliese, E.Tuosto, B.Venneri. **Global Computing: Programming Environments, Languages, Security and Analysis of Systems**, LNCS 2874, pagg. 88 – 150. Springer, 2003. Parzialmente basato su [C2, C3].

Pubblicazioni su riviste internazionali:

[J1] *"Process Calculi and the Verification of Security Protocols"*, assieme a Michele Boreale. Parzialmente basato su [C1]. **International Journal of Telecommunication and Information Technology**, special issue su Cryptographic Protocol Verification, pagg. 28 – 40. National Institute of Telecommunications – Warsaw, 2002.

[J2] *"Security Policies as Membranes in Systems for Global Computing"*, assieme a Matthew Hennessy e Vladimiro Sassone. Versione estesa di [C7]. **Logical Methods in Computer Science**, 1(3:2), 1 – 23, 2005. Precedentemente apparso come Research Rep. 02/04, Dept. Informatics, Un. Sussex at Brighton (UK)

[J3] *"Role-Based Access Control for a Distributed Calculus"*, assieme a Chiara Braghin e Vladimiro Sassone. Versione estesa di [C5]. **Journal of Computer Security**, 14(2):113 – 155, IOS Press, 2006. Precedentemente apparso come Tech.Rep. 08/2004, Dip. Informatica, Univ. "La Sapienza".

[J4] *"On the expressive power of the KLAIM language"*, assieme a Rocco De Nicola e Rosario Pugliese. Versione estesa di [C6]. **Theoretical Computer Science**, 356(3):387 – 421, Elsevier, 2006. Precedentemente apparso come Tech.Rep. 09/2004, Dip. Informatica, Univ. "La Sapienza".

[J5] *"Confining Data and Processes in Global Computing Applications"*, assieme a Rocco De Nicola e Rosario Pugliese. Versione estesa di [C4]. **Science of Computer Programming**, 63(1):57 – 87, Elsevier, 2006.

[J6] *"Global Computing in a Dynamic Network of Tuple Spaces"*, assieme a Rocco De Nicola e Rosario Pugliese. Versione estesa di [C8]. **Science of Computer Programming**, 64(2):187 – 204, Elsevier Science, 2007. Precedentemente apparso come Tech.Rep. 05/2005, Dip. Informatica, Univ. "La Sapienza".

[J7] *"Basic Observables for a Calculus for Global Computing"*, assieme a Rocco De Nicola e Rosario Pugliese. Versione estesa di [C10]. **Information and Computation**, 205(10):1491 – 1525, Elsevier 2007. Precedentemente apparso come Tech.Rep. 07/2004, Dip. Informatica, Univ. "La Sapienza".

[J8] *"Comparing Communication Primitives via their Relative Expressive Power"*. Versione estesa e rivista di [C11] e [C13]. **Information and Computation**, 206(8):931 – 952, Elsevier 2008.

[J9] *"Dynamic management of capabilities in a network aware coordination language"*, assieme a Rosario Pugliese. Versione estesa ed ampliata di [C2]. **Journal of Logic and Algebraic Programming**, 78:665 – 689, Elsevier 2009. Precedentemente apparso come Tech.Rep. 06/2004, Dip. Informatica, Univ. "La Sapienza".

[J10] *"Tree-functors, Determinacy and Bisimulations"*, assieme a Rocco De Nicola e Anna Labella. **Mathematical Structures in Computer Science**, 20(3):319 – 358, CUP 2010. Precedentemente apparso come Tech. Rep. 02/2006, Dip. Informatica, Univ. "La Sapienza" col titolo "Characterising Bisimulations Functorially".

[J11] *"From Flow Logic to Static Type Systems for Coordination Languages"*, assieme a R.De Nicola, R.R.Hansen, F.Nielson, H.Riis Nielson, C.W. Probst e R.Pugliese. Versione estesa di [C14]. **Science of Computer Programming**, 75(6): 376 – 397. Elsevier, 2010.

[J12] *"Towards a Unified Approach to Encodability and Separation Results for Process Calculi"*. Versione estesa e rivista di [C16]. **Information and Computation**, 208(9):1031 – 1053. Elsevier 2010.

[J13] *"A Taxonomy of Calculi for Distribution and Mobility"*. Versione estesa e rivista di [C17] e [C18]. Apparirà su **Distributed Computing**, Springer 2010. Precedentemente apparso come Tech. Rep. 09/2006, Dip. Informatica, Univ. "La Sapienza" col titolo "Comparing Calculi for Mobility via their Relative Expressive Power".

Pubblicazioni come Editore:

[E1] *"Security Issues in Concurrency (SecCo'07): Proceedings"*, assieme a Catuscia Palamidessi. **Electronic Notes in Theoretical Computer Science**, volume 194(1). Elsevier, 2008.

[E2] *"Expressiveness Issues in Concurrency (EXPRESS'08): Proceedings"*, assieme a Thomas Hildebrandt. **Electronic Notes in Theoretical Computer Science**, volume 242(1). Elsevier, 2009.

[E3] *"Expressiveness Issues in Concurrency (EXPRESS'09): Proceedings"*, assieme a Sybille Froeschle. **Electronic Proceedings in Theoretical Computer Science**, volume 8, 2009.

[E4] *"Security Issues in Concurrency (SecCo'07): Special issue"*, assieme a Catuscia Palamidessi. **Journal of Computer Security**, volume 18(2). IOS Press, 2010.

[E5] *"Expressiveness Issues in Concurrency (EXPRESS'08): Special issue"*, assieme a Thomas Hildebrandt. **Mathematical Structures in Computer Science**, volume 20(1). CUP, 2010.

[E6] *"Expressiveness Issues in Concurrency (EXPRESS'09): Special issue"*, assieme a Sybille Froeschle. **Mathematical Structures in Computer Science**, in preparazione.

Libri di Testo:

[L1] *"Introduzione alla logica e al linguaggio matematico"*, assieme a Giorgio T. Bagni e Anna Labella. McGraw-Hill, 2009. ISBN 978-88-386-6505-9.

Rapporti di Ricerca:

[R1] *"Enforcing Security Policies via Types"*, assieme a Rosario Pugliese. Tech.Rep. 05/2004, Dip. Informatica, Univ. "La Sapienza". Versione estesa di [C3]

[R2] *"Inferring Dynamic Credentials for Role-based Trust Management"*, assieme a Matthew Hennessy e Vladimiro Sassone. Tech. Rep. 04/2006, Dip. di Informatica, Univ. di Roma "La Sapienza" (Italy). Versione estesa di [C12].

[T1] **Tesi di Laurea: "***Grammatiche di sincronizzazione per generare l'esposizione di una fuga"*. Dip. Informatica, Univ. di Roma "La Sapienza", Dicembre 2000.

[T2] **Tesi di Dottorato: "***Semantic Approaches to Global Computing Systems"*. PhD thesis XVII-04-I, Dip. di Sistemi ed Informatica, Univ. di Firenze. Dicembre 2004. **Menzione del Consiglio Italiano dell'EATCS** come seconda migliore tesi di dottorato del biennio 2004/05 di argomento "Programming Languages and Semantics".

Calcolo dell'*h*-index:

- Effettuato mediante *publish-or-perish*: **11**
- Effettuato manualmente tramite *google-scholar*, escludendo le auto-citazioni: **10**
    - [B1] citato da 77
    - [C2] citato da 29
    - [C7] citato da 25
    - [C5] citato da 22
    - [J12] citato da 20
    - [J7] citato da 18
    - [C11] citato da 13
    - [C1] citato da 13
    - [J5] citato da 10
    - [C3] citato da 10

Calcolo del *g*-index:

- Effettuato mediante *publish-or-perish*: **18**
- Effettuato manualmente tramite *google-scholar*, escludendo le auto-citazioni: **16**
  Infatti, oltre ai 10 articoli citati per il calcolo dell'*h*-index, si hanno i seguenti articoli:
    - [C8] citato da 10
    - [C4] citato da 7
    - [J5] citato da 5
    - [C9] citato da 5
    - [J8] citato da 5
    - [C13] citato da 4

Io sottoscritto dichiaro che tutto quanto è contenuto nel presente curriculum corrisponde a verità, ai sensi degli artt. 46 e 47 del D.P.R. 445/2000

Roma, 3 Agosto 2010

Presentazione della ricerca svolta

La ricerca svolta finora si inquadra nello studio di formalismi per programmi concorrenti. In base ai diversi aspetti considerati, si possono identificare tre linee di ricerca principali: sviluppo ed utilizzo di equivalenze comportamentali per descrivere e verificare il comportamento di programmi concorrenti; approcci basati sull'analisi statica di programmi per specificarne e garantirne proprietà di sicurezza; confronto di diversi formalismi, in base al loro potere espressivo. Tali studi sono stati condotti su linguaggi minimali (anche noti come *calcoli di processo*) con caratteristiche evolute, tra cui primitive crittografiche, sofisticate primitive di comunicazione, distribuzione e mobilità del codice.

Di seguito si riportano commenti dettagliati sulle tre linee di ricerca; [T2] raccoglie buona parte della ricerca svolta fino all'inizio del 2005 (anche se apparsa in letteratura negli anni successivi) raggruppando assieme lavori dei tre filoni di ricerca [C2,C3,C4,C6,C8,C9,C10,C14,B1,J4,J5,J6,J7,J9,J11,R1]. Il criterio di selezione è stato il particolare linguaggio su cui le varie teorie sono state sviluppate; in particolare, è stato usato il linguaggio KLAIM [DFP98], acronimo di "Kernel Language for Agents Interaction and Mobility". Le caratteristiche fondamentali di tale linguaggio sono: processi distribuiti su una rete di nodi, possibilità di programmare il movimento di codice tra i vari nodi della rete, comunicazione sia locale che remota basata su spazi delle tuple e pattern matching [Gel85].

1. Studio ed utilizzo di equivalenze comportamentali

Una caratteristica fondamentale di un calcolo di processi è la possibilità di definire equivalenze comportamentali che permettano di ragionare sulle funzionalità di un dato programma senza considerarne i dettagli implementativi. Tali equivalenze possono essere usate, ad esempio, per mostrare che un'implementazione corrisponde alla relativa specifica o possono essere alla base di un'ottimizzazione del codice. Tra le varie equivalenze definite per sistemi concorrenti, di particolare importanza è la *bisimulazione* [Park81]: essa mette in relazione processi in grado di esibire lo stesso comportamento osservabile e di evolvere in processi ancora bisimili. Ovviamente, formalismi diversi portano a nozioni di bisimulazione diverse; nella ricerca svolta abbiamo considerato alcuni di questi formalismi, le relative bisimulazioni e l'uso che se ne può fare per provare proprietà di programmi concorrenti.

In [C10,J7] definiamo nozioni di bisimulazione per calcoli derivati da KLAIM in grado di modellare fallimenti e problematiche di connettività della rete. La teoria è utilizzata in [C8,J6,J7,T2] per dimostrare la correttezza di alcuni protocolli tipici di sistemi distribuiti e fault-tolerant: 'I filosofi a cena', 'il k-set agreement' e due protocolli per il message delivering.

Abbiamo inoltre considerato lo Spi-calcolo [AG98], una variante crittografica del π-calcolo [MPW92] in cui i dati scambiati in una comunicazione possono essere cifrati. In questo scenario, abbiamo sviluppato in [C1] un'assiomatizzazione corretta (e completa, nel caso di processi finiti) per la bisimulazione. Oltre all'interesse teorico, tale risultato è di utilità pratica poiché fornisce un metodo molto semplice di provare proprietà di protocolli crittografici. Ciò è mostrato in [J1], dove leggi equazionali derivate dall'assiomatizzazione di [C1] sono usate per dimostrare proprietà di segretezza ed autenticità del protocollo KERBEROS [KN93].

Un'altra applicazione della bisimulazione è in [C15], dove abbiamo definito tale relazione per un modello della concorrenza basato sulla riscrittura di ipergrafi. Un ipergrafo viene visto come un'astrazione di un sistema concorrente (in cui i nodi rappresentano risorse e gli iperarchi i processi che usano tali risorse) e un passo di riscrittura corrisponde all'esecuzione simultanea di azioni da parte dei processi in esecuzione nel sistema. Anche in questo caso la teoria viene poi applicata a casi concreti: ad esempio, mostriamo che la bisimulazione coincide con l'equivalenza funzionale su grafi che rappresentano componenti con input e output (come un circuito) ed uguaglia grafi con stesso flusso massimo, per grafi che rappresentano reti di flusso, o con stessa fault-tolerance, per grafi con nodi che possono fallire in modo permanente.

Sulla stessa linea di ricerca, in [C19] sviluppiamo un modello per reti ad hoc in cui i nodi della rete hanno una carica che diminuisce nel corso delle computazioni a seguito del passaggio di informazioni. Anche in questo caso, vogliamo utilizzare equivalenze comportamentali per mettere in relazione reti diverse ma con lo

stesso comportamento osservabile. L'equivalenza considerata in questo lavoro è l'equivalenza a tracce, secondo cui due reti sono equivalenti se sono in grado di esibire le stesse sequenze di azioni (chiamate, appunto, tracce). Il risultato tecnico principale di questo lavoro è il mostrare che tale equivalenza in questo modello può essere caratterizzata mediante due semplici misure: il massimo flusso di informazione che una rete può trasmettere e il minimo flusso di informazione in grado di bloccare la rete, dove una rete è bloccata se non può più trasmettere informazione a causa della perdita di energia di alcuni suoi nodi. Inoltre, il lavoro mostra che calcolare quest'ultima misura è un problema NP-completo e quindi incoraggia lo studio di altre equivalenze (tra cui la bisimulazione) con una caratterizzazione computazionalmente più efficiente.

Per meglio valutare e comprendere le nozioni di equivalenza usate, abbiamo anche indagato tali equivalenze in astratto, studiandole cioè non per un particolare linguaggio di riferimento ma piuttosto passando ad un modello denotazionale della programmazione concorrente basato su alberi, in grado di modellare sia formalismi di tipo *interleaving* che *truly concurrent*. In tale modello, abbiamo studiato [J10] le proprietà categoriali che caratterizzano tre fondamentali equivalenze in teoria della concorrenza: le versioni strong, branching e weak della bisimulazione. I risultati ottenuti mostrano come tali equivalenze possano essere caratterizzate in maniera molto naturale tramite semplici proprietà di un funtore arricchito, fornendone pertanto un'interpretazione più astratta.

2. Metodi formali per la sicurezza di programmi concorrenti (Language-based security)

Un aspetto cruciale di tutti i formalismi e linguaggi concorrenti è la scelta e il modellamento di una o più forme di interazione tra processi. Collegata a quest'aspetto è la necessità di specificare e garantire opportune proprietà di sicurezza dei sistemi che si andranno a programmare. In letteratura è presente una grandissima varietà di tali proprietà, alcune delle quali sono state affrontate nella ricerca presentata.

Una proprietà tipica è la *segretezza dei dati*. A tal scopo, in [C4] presentiamo un linguaggio in cui il programmatore etichetta i dati sensibili specificando quali nodi della rete (cioè, quali utenti) possono accedere al dato. Si sviluppa poi un sistema di tipi in grado di garantire che questo vincolo verrà rispettato in ogni esecuzione di un sistema ben tipato. Il lavoro è adattato in [J5] a tre calcoli distribuiti: KLAIM [DFP98], il π-calcolo distribuito [HR02] e l'Ambient calcolo [CG00]. Ciò mostra come scelte di linguaggio anche molto diverse tra loro possano comunque supportare uno stesso approccio di tipi, anche se i dettagli tecnici dei vari sistemi sono tra loro differenti e condizionati dalle scelte di linguaggio.

Una problematica ortogonale alla sicurezza dei dati in un sistema distribuito è il *controllo delle attività dei processi* che esso contiene. A tal scopo, in [C2,C3,C7,C14,B1,J2,J9,J11,R1] definiamo dei sistemi di tipo volti a controllare le esecuzioni dei processi su di un nodo: ogni nodo ha associato un tipo (che qui viene visto come la politica del nodo) che descrive le operazioni lecite eseguibili da processi in esecuzione su di esso; l'approccio adottato è il ben noto meccanismo basato su capabilities [SSF99]. In un certo senso, il tipo di un nodo è la sua interfaccia, in quanto è un limite superiore alle azioni che possono essere compiute dai processi che esso ospita. In [C14,B1,J11] presentiamo un sistema di tipo basilare, che viene poi arricchito negli altri lavori con caratteristiche più sofisticate; in [C14,J11] mostriamo inoltre che l'approccio presentato coincide con un'altra tecnica di analisi statica nota in letteratura con il nome di *flow logic* [NNH05]. In [C3,R1] i diritti sono accordati in base all'origine di un processo entrante e si introduce una classificazione più fine delle operazioni, tenendo in considerazione non solo il tipo delle operazioni, ma anche il loro argomento: ad esempio, operazioni che prevedono la lettura di un dato segreto non vengono trattate come le operazioni di lettura di un dato pubblico. In [C2,J9] sviluppiamo il modello base in un'altra direzione, ortogonale alla precedente: il tipo di un nodo può variare nel corso di una computazione a seguito delle interazioni avvenute tra quel nodo ed il resto della rete. Così, alcuni privilegi possono venir acquisiti, persi, trasmessi o revocati dal nodo. Questo scenario descrive bene la realtà di processi in esecuzione su reti WAN e modella in maniera semplice ed elegante applicazioni di commercio elettronico. Ovviamente, la semantica deve essere definita in modo da assicurare che non sia possibile montare attacchi contro la sicurezza del sistema, come ad esempio forgiare capabilities, revocare capabilities in maniera indiscriminata, etc. Infine, in [C7,J2] si analizzano diverse tipologie di politiche per un nodo e le relative proprietà di sicurezza che sono in grado di garantire. In particolare, variando di poco la definizione di tipo (come insieme di azioni, multinsieme di azioni o automa a stati finiti i cui input sono azioni), si possono facilmente esprimere

politiche sempre più sofisticate. Chiaramente, verificare che un processo rispetti la politica specificata da un tipo di elevato potere espressivo è molto costoso in termini computazionali; a tal scopo, usiamo nozioni di trust e relazioni di sottotipo per rendere il type-checking di un processo il più efficiente possibile.

In sistemi distribuiti su scala geografica, l'utilizzo del *trust* tra gli utenti è un modo semplice, robusto e scalabile per assicurare proprietà di sicurezza. L'idea di fondo è di concedere autorizzazioni in base al possesso di certificati che, opportunamente composti, forniscano un'evidenza inequivocabile della legittimità dell'autorizzazione. Tecniche d'analisi statiche (quali sistemi di tipi o logiche) sono state usate con successo per specificare ed utilizzare ambienti in cui il trust è l'elemento fondante [LMW03]. In [C12,R2] abbiamo esteso tali meccanismi per la gestione di certificati in modo da vincolarne la validità in base al momento e al contesto in cui essi sono utilizzati. Entrambe queste caratteristiche sono già da tempo usate nei sistemi commerciali basati sul trust; il loro studio formale ha però rivelato complicazioni tecniche non banali ed ha evidenziato possibili evoluzioni dei sistemi usati in pratica.

Tutti i lavori discussi finora sono basati sul meccanismo di controllo degli accessi definito in letteratura *modello discrezionale*. In tale modello, l'amministratore di un sistema assegna agli utenti dei diritti in base alla loro identità (rappresentata, nel nostro approccio, dall'indirizzo del nodo della rete associato all'utente). In [SCFY96] è stato introdotto un meccanismo alternativo per assegnare diritti ad utenti: il *role-based access control (RBAC)*. In tale modello, i ruoli si frappongono nell'assegnazione dei privilegi agli utenti: i privilegi vengono assegnati ai ruoli (questa assegnazione è solitamente mantenuta invariata nel corso della vita di un sistema) e i ruoli agli utenti (questa assegnazione, invece, varia nel corso della vita di un sistema). Visto che nel corso della computazione gli utenti svolgono diversi ruoli, le azioni che essi intendono eseguire devono essere abilitate dai privilegi associati ai ruoli svolti al momento dell'esecuzione. In [C5,J3] abbiamo studiato l'impatto di tale meccanismo di gestione delle politiche in una versione distribuita del π-calcolo [MPW92] definendo un sistema di tipi in grado di trattare proprietà di sicurezza ed una equivalenza comportamentale per ragionare sulle funzionalità dei sistemi considerati. Come mostrato dagli esempi, la teoria facilita la definizione concreta dei sistemi, una volta fissate le politiche da rispettare.

In tutti i lavori descritti, la sicurezza viene vista come una proprietà *qualitativa* (o binaria) del sistema in esame: un sistema o è sicuro o non lo è. Di recente, molto interesse della comunità si è rivolto allo studio di proprietà *quantitative* di sistemi concorrenti, in particolare collegate alla sicurezza. Un esempio tipico è la formulazione quantificata della non-interferenza [GM82], secondo cui non è più interessante sapere se un programma rivela informazioni segrete o meno, ma piuttosto quanta informazione segreta viene svelata, espressa ad esempio in termini di numero di bit. Un caso tipico è il meccanismo di verifica di una password: a seguito di un tentativo, l'attaccante ottiene qualche informazione sulla password provata (o sa che è quella giusta, oppure sa di sicuro che la password è un'altra). In quest'ottica si inquadra [C21], dove studiamo un meccanismo matematico basato sull'approccio coalgebrico [Rut03] per calcolare la correlazione tra azioni segrete e azioni pubbliche di una semplice algebra di processi non-deterministici. In questo modo abbiamo formulato in maniera alternativa la nozione di *non-deducibility on strategies* di [WJ90], arricchendola con un algoritmo per il calcolo effettivo della matrice di correlazione fornito da [Rut03].

3. Confronto tra diversi formalismi: potere espressivo

Una realtà evidente nell'ambito dei linguaggi di programmazione è l'enorme varietà di proposte presenti in letteratura e usate in pratica. I calcoli di processi, e soprattutto i formalismi con codice mobile sviluppati negli ultimi anni, sono un tipico esempio di questa realtà: molti linguaggi differiscono tra loro per piccoli dettagli e spesso non è ben chiara la differenza tra i vari formalismi. Si è quindi in più modi cercato un criterio per confrontare tra loro diversi linguaggi; a tutt'oggi, non si è ancora sviluppata un'unica metodologia per affrontare questo problema.

Un primo approccio è quello basato sul *potere espressivo assoluto* di un linguaggio. L'idea è di presentare un problema che bipartizioni i linguaggi in base alla possibilità di risolvere il problema in questione. La bipartizione dei linguaggi può essere calcolata formalmente o informalmente. Nel primo caso, si deve fornire una soluzione al problema, per i linguaggi in grado di fornirla, e dimostrare che nessuna soluzione è possibile negli altri; l'ultimo compito è solitamente molto complesso (si vedano, ad esempio, [Her91,Pal03]). Il

metodo informale, invece, è quello di mostrare applicazioni che si modellano bene in un linguaggio e male in altri. Sebbene questo approccio non abbia una solida base teorica, è molto usato in pratica anche perché ha il vantaggio di essere molto vicino al programmatore, mostrandogli le applicazioni tipiche in cui è da preferire un linguaggio rispetto ad un altro. Abbiamo seguito questo approccio in [C9], dove abbiamo mostrato il potere espressivo di diverse varianti, sempre di facile implementazione distribuita, del meccanismo standard di pattern matching del linguaggio LINDA [Gel85].

L'approccio basato sul potere espressivo assoluto di un linguaggio ha due svantaggi: anzitutto, bipartiziona i linguaggi in esame, appiattendo quindi su due soli livelli una possibile gerarchia multilivello tra linguaggi; inoltre, il suo utilizzo è molto limitato per via della notevole difficoltà a dimostrare risultati di impossibilità. Un metodo più flessibile e più maneggevole è quello basato sul *potere espressivo relativo* di due o più linguaggi: tale approccio consiste nel codificare un linguaggio in un altro, o nel dimostrare l'impossibilità di tale codifica. Questo approccio è un'evoluzione del precedente, nel senso che mostra rigorosamente come i costrutti di un linguaggio siano implementabili in un altro o, viceversa, quali caratteristiche di un linguaggio non possano essere implementate. Tuttavia, è possibile sviluppare prove di impossibilità in maniera più semplice, basandosi su proprietà sintattiche e/o semantiche dei linguaggi in questione. Chiaramente, per sviluppare tali risultati, è cruciale fissare le proprietà che la funzione di codifica deve rispettare. Intuitivamente, la codifica deve tradurre termini del linguaggio di partenza senza modificarne il comportamento; di nuovo, ciò può essere formalizzato in molti modi e non c'è un accordo comune su quale sia il modo migliore di formalizzare tale richiesta intuitiva. In quest'ottica si inquadrano [C6,C11,C13,C16, C17,C18,C20,J4,J8,J12,J13].

In [C6,J4] studiamo codifiche *fully abstract* (cioè, che traducono processi equivalenti in processi equivalenti, e viceversa) tra diversi dialetti del linguaggio KLAIM [DFP98]; i vari dialetti semplificano progressivamente il linguaggio di partenza fino a renderlo un calcolo di estrema semplicità. Mentre la nozione di full abstraction ha molto senso in un ambito denotazionale [Plo77], nell'ambito dello studio del potere espressivo relativo ha delle limitazioni. Infatti, valutare la qualità di una codifica basandosi esclusivamente sulla full abstraction può portare a risultati sorprendenti: scegliendo opportunamente le equivalenze da considerare, si possono sempre ottenere codifiche fully abstract o, viceversa, si possono scartare codifiche chiaramente buone. Queste osservazioni evidenziano il fatto che la full abstraction non può essere l'unico criterio per studiare la qualità di una codifica. Ciò che caratterizza i lavori che ora andremo a discutere è la proposta di una serie di criteri più adatto della full abstraction per valutare la qualità di una codifica.

L'idea è quella di accettare come "valide" solo le funzioni di codifica che rispettano certi criteri fissati. Una volta fissati i criteri, si ottiene un pre-ordine sui linguaggi esaminati tale che $L_1 \leq L_2$ se e soltanto se L_1 può essere tradotto in L_2 tramite una codifica "valida". Ovviamente, cambiando i criteri anche i pre-ordini ottenuti in generale cambieranno. Pertanto, in [C16,J12] abbiamo definito un insieme di criteri e abbiamo cercato di supportarli. La proposta formulata consiste nel chiedere che la codifica sia composizionale (cioè, la codifica di un termine composto sia ottenuta componendo le codifiche dei sottotermini), che non dipenda dai nomi che compaiono nei termini da tradurre (cioè, dipenda solo dalla struttura sintattica dei termini), che preserva e rifletta la computazioni, che non introduca divergenza (cioè, non traduca un termine terminante in un termine con una computazione infinita) e che rispetti una nozione di terminazione con successo. Per mostrare la validità della nostra proposta, abbiamo mostrato che: *(1)* tutti i più noti risultati di codificabilità rispettano i criteri proposti; *(2)* tali criteri non sono banali, nel senso che in letteratura esistono codifiche (anche proposte da autori di prim'ordine, ad esempio [CG00]) che non li soddisfano; *(3)* tutti i risultati di non-codificabilità apparsi finora valgono adottando tali criteri e, anzi, la loro prova risulta molto più semplice; ed infine *(4)* nuovi risultati di impossibilità, alcuni dei quali solo congetturati finora, possono essere formalmente provati usando tali criteri [C11,C13,C17,C18,C20,J8,J13].

Come già detto, il vantaggio dell'approccio basato sul potere espressivo relativo è principalmente quello di costruire gerarchie di linguaggi basate sulla possibilità/impossibilità di codificare un linguaggio in un altro. Un primo esempio è [C11,C13,J8], dove studiamo il potere espressivo di diverse primitive di comunicazione in una famiglia di calcoli basati sul π-calcolo [MPW92]. Le varie primitive sono ottenute combinando in tutti i modi possibili quattro caratteristiche tipiche della comunicazione nei calcoli di processo: sincronia (comunicazioni sincrone e asincrone), arità (spedizione di un singolo messaggio o di più messaggi alla volta), mezzo di comunicazione (canali o spazi delle tuple) e assenza/presenza di pattern matching, nello stile

di [Gel85]. Un secondo esempio è [C17,C18,J13], dove confrontiamo diversi calcoli con mobilità e distribuzione, tra cui il π-calcolo [MPW92], una sua versione distribuita [HR02], Ambient [CG00] e sue varianti (tra cui alcune varianti di Boxed Ambient [BCC04] e di Safe Ambient [LS03]).

Infine, in [C20] abbiamo definito una variante concorrente del pattern calculus [Jay09], un'estensione del lambda-calcolo con strutture dati e pattern matching. L'eleganza e il potere espressivo del nuovo calcolo è stato dimostrato sia tramite esempi che con un confronto rigoroso con altri calcoli di processo, sempre basato sul metodo proposto in [C16,J12]: il π-calcolo [MPW92], lo Spi-calcolo [AG98], LINDA [Gel85] e Fusion [PV98].

Bibliografia essenziale:

[AG98] M. Abadi, A.D. Gordon. *A Calculus for Cryptographic Protocols: the Spi-calculus*. **Information and Computation**, 148(1):1-70, 1999.

[BCC04] M. Bugliesi, G. Castagna, S. Crafa. *Access control for mobile agents: the calculus of Boxed Ambients*. **Trans. on Programming. Languages and Systems**, 26(1):57–124. ACM 2004.

[CG00] L. Cardelli, A.D. Gordon. *Mobile Ambients*. **Theoretical Computer Science**, 240(1):177-213, 2000.

[DFP98] R. De Nicola, G. Ferrari, R. Pugliese. *KLAIM: a Kernel Language for Agents Interaction and Mobility*. **IEEE Transactions on Software Engineering**, 24(5):315-330, 1998.

[Gel85] D. Gelernter. *Generative Communication in LINDA*. **ACM Transactions on Programming Languages and Systems**, 7(1):80-112, 1985.

[GM82] J. Goguen, J. Meseguer. *Security policies and security models*. Atti del **Symposium on Security and Privacy**, pag. 11 – 20. IEEE Computer Society, 1982.

[Her91] M. Herlihy. *Wait-Free Synchronization*. **Transactions on Programming Languages and Systems**, 13(1):124–149. ACM Press, 1991.

[HR02] M. Hennessy, J. Riely. *Resource Access Control in Systems of Mobile Agents*. **Information and Computation**, 173:82-120. 2002.

[Jay09] B. Jay. *Pattern Calculus: Computing with Functions and Data Structures*. Springer, 2009.

[KN93] J. Kohl, B. Neuman. *The Kerberos Network Authentication Service (version 5)*. Internet Request for Comment RFC-1510, 1993.

[LMW03] N. Li, J.C. Mitchell, W.H. Winsborough. *Beyond Proof-of-compliance: Security Analysis in Trust Management*. **Journal of the ACM**, 52(3):474--514, 2005.

[LS03] F. Levi, D. Sangiorgi. *Mobile safe ambients*. **Transactions on Programming Languages and Systems**, 25(1):1–69. ACM Press, 2003.

[MPW92] R. Milner, J. Parrow, D. Walker. *A Calculus of Mobile Processes (part I and II)*. **Information and Computation**, 100(1):1-77, 1992.

[NNH05] F. Nielson, H. Riis Nielson, C. Hankin. *Principles of Program Analysis*. Springer Verlag, Berlin, Germany, second edition, 2005.

[Pal03] C. Palamidessi. *Comparing the expressive power of the synchronous and the asynchronous π-calculi*. **Mathematical Structures in Computer Science**, 13(5):685–719, 2003.

[Park81] D. Park. *Concurrency and automata on infinite sequences*. Proc. of **Theoretical Computer Science**, volume 104 of LNCS, pages 167-183. Springer, 1981.

[Plo77] G.D. Plotkin. *LCF considered as a programming language*. **Theoretical Computer Science**, 5:223–255, 1977.

[PV98] J. Parrow, B. Victor. *The fusion calculus: Expressiveness and symmetry in mobile processes*. Atti di **LICS**, pagg. 176 – 185. IEEE Computer Society, 1998.

[Rut03] J.J.M.M. Rutten. *Behavioural differential equations: a coinductive calculus of streams, automata, and power series*. **Theoretical Computer Science**, 308(1-3):1–53, 2003.

[SCFY96] R. Sandhu, E. Coyne, H. Feinstein, C. Youmann. *Role-Based Access Control Models*. **IEEE Computer**, 29(2):38-47, 1996.

[SSF99] J. S. Shapiro, J. M. Smith, D. J. Farber. *EROS: a fast capability system*. Atti del **Symposium on Operating Systems Principles**, pagg. 170–185, 1999.

[WJ90] J.T. Wittbold, D.M. Johnson. *Information flow in nondeterministic systems*. Atti del **Symposium on Security and Privacy**, pages 144-161. IEEE Computer Society, 1990.