Curriculum Vitae

Gianluca Amato
Dipartimento di Economia
Università "G. d'Annunzio" Chieti–Pescara

Telephone: +39 085 4537686
e-mail: gianluca.amato@unich.it

Education

Jul 1996 Master's degree in "Scienze dell'Informazione" (Computer Science), University of Pisa (final marks: 110/110 with honors). Master's Thesis Title: Properties of the lattice of observables in logic programming". Supervisor: Giorgio Levi. Referee: Laura Ricci.

Oct 1997 Licenza (Special Master's degree) in Scienze dell'Informazione (Computer Science), Scuola Normale Superiore di Pisa (final marks: 70/70 with honors).

Apr 2001 Ph.D. degree in Computer Science, University of Pisa. Thesis title: "Sequent Calculi and Indexed Categories as a Foundation for Logic Programming". Advisor: Giorgio Levi. International referees: Fernando Orejas, James Harland.

Academic positions

Dec 2000 – Aug 2002 Post-doc at the Mathematics and Computer Science Department of the University of Udine, Italy, within the project "Verification of Declarative Programs".

Sep 2002 – Oct 2017 Permanent position as Assistant Professor (Ricercatore) in Computer Science at the University of Chieti-Pescara, Italy

Nov 2017 – now Permanent position as Associate Professor in Mathematical Logic at the University of Chieti-Pescara, Italy

Periods of studies and research abroad

Jan – Jun 1998 "Mathematics and Computer Science Department", Wesleyan University, Middletown (CT), USA, collaborating with prof. James Lipton.

Mar – Apr 2001 "Departamento de Sistemas Informáticos y Computación", Universidad Poliécnica de Valencia, Spain.

Nov – Dec 2003 "Mathematics and Computer Science Department", Wesleyan University, Middletown (CT), USA, collaborating with prof. James Lipton.

Jul – Aug 2004 "STIX Laboratoire", École Polytechnique, Paris, collaborating with the research team led by prof. Radhia Cousot.

Nov – Dec 2015 "Mathematics and Computer Science Department", Wesleyan University, Middletown (CT), USA, collaborating with prof. James Lipton.

Teaching and related activities

- Lecturer for several courses at the University of Chieti–Pescara: operating systems, computer architecture, databases, web development, data mining, deep learning, software engineering, history of computer science, mathematical logic, calculus.
- Lecturer and teaching assistant for the "School on univalent Mathematics", organized by "Scuola Matematica Interuniversitaria (SMI)", July 17th - July 23rd 2022, Cortona, Italy.
- Advisor of more than 70 bachelor theses, 15 master theses and 1 PhD thesis in computer science at the University of Chieti–Pescara.

System administration activities

- System administrator for the computer lab "Aula Informatica C" of the University of Chieti–Pescara, a classroom with around 80 desktop PCs with Linux operating system, a centralized authentication system based on FreeIPA and centralized home directories based on NFS.
- System administrator for a cluster (several compute nodes and one shared storage node) based on the oVirt virtualization system. The cluster contans both hardware and software dedicated to research (such as Jupyter Hub) and to didactics (such as Moodle).
- Global administrator for the Microsoft 365 tenant at the University of Chieti–Pescara.

Organization of conferences and schools

- Member of the organizing committee for the following conferences: Programming languages, Implementations, Logics and Programs (PLILP '98), Algebraic and logic programming (ALP '98), Joint Conference on Declarative programming APPIA – GULP – PRODE (AGP '00), Workshop on Functional and (Constraint) Logic Programming (WFLP '02).
- Member of the program committee for the following conferences: NSAD (Numerical & Symbolic Abstract Domains) '14 , NSAD '17.

- Organizer of the following conferences at the University of Chieti–Pescara: Geometry and Computer Science (GNCS '17), MATE – Maieutica Arte Tecnologie Economia (MATE '18), Applied Proof Theory (APT '22)
- Organizer of the following schools at the University of Chieti–Pescara: 6th Ph.D. School/Conference on Mathematical Modeling of Complex Systems (2019), Geometric Deep Learning First Italian School (2022).

Research interests

Abstract Interpretation Theory. Abstract interpretation is a semantic theory introduced by Patrick and Radhia Cousot in 1977 in order to design static analysis of computer programs and, more in general, software systems.

In [40, 1] we prove that completeness of abstract domains w.r.t. semantic operators is a property which is preserved by the operation of lowest upper bound and greatest lower bound in the lattice of abstractions, provided the semantic operators are completely additive. We also show some negative results concerning some properties of domain refinements that, although seem quite natural, do no hold in practice.

In [15, 10] we introduce the concept of *observational completeness*, which is useful when we want abstractions which have the greatest possible precision but only w.r.t. a given class of observations. An application of this concept appears to the field of cellular automata in [17].

In [21] we introduce two techniques called *localized widening* and *localized narrowing*. Both are variations of the standard analysis algorithm, based on an ascending phase (which computes an approximated result of the analysis) followed by a descending phase (which refines the result). With localized widening, we apply widening during ascending phase in a more selective way, which leads to a greater precision in the presence of nested loops. Localized narrowing is a more radical variant of the standard analysis algorithm, in which the two phases are replaced by a continuous alternation of ascending and descending phases, in which partial results are immediately refined locally, based on a hierarchical ordering of the equations. In [28] we integrate localization with a previous work of Seidl, Apinis e Vojdani and we propose a completely different approach for solving data-flow analysis, which replaces widening and narrowing with a single binary operator called warrowing. The new approach is immediately applicable to equation systems with an infinite number of unknowns, a case which is quite common in context-sensitive inter-procedural analyses.

In [36, 32] we study the relation between the collecting semantics and the analyses it is suitable for. We characterize many standard collecting semantics for first order functional programming as the initial object in appropriately chosen families of analysis.

Static analysis of logic programs. In [5] we implement an abstract domain for goal-independent pair-sharing analysis of logic programs. This is the first implementation of a domain obtained by linear refinement. The method of linear refinement may be

used when the concrete domain and the operation we want to refine make a quantale, which is an algebraic model of linear logic. In [6, 11] we consider the domain `Sharing` by Langen, one of the most known domains for the analysis of sharing properties, and we introduce new best-correct abstract operators to improve its precision in goal-dependent analysis. Finally, in [7, 14] we consider the interaction between aliasing and linearity information, and we show for the first time the best-correct abstract operator for `Sharing` $\times$ `Lin` and similar domains. In [22] the unification operator of `Sharing` $\times$ `Lin`, which is critical to achieve precision in the analysis, is proved to be optimal even in the case of substitutions with multiple bindings.

Extensions of logic programming languages. In [2, 3, 41] we propose a framework for reasoning about properties of proofs in sequent calculi. Using this framework we try to reconcile the proof-theoretic approach to logic programming, based on the concept of uniform proof, with the classic approach based on model-theory and fix-point theory. Using abstract interpretation to model relationships among proofs we are able to recover well known results on the analysis of pure logic programs in richer languages such as λ-prolog and LinLog.

While this approach is based on proof-theory, there are languages such as CLP where model-theory plays a fundamental role. In [41, 4, 9] we show a categorical framework which is able to deal with the operational, denotational and declarative semantics of a wide class of logic languages. In this way, it is possible to deal with concepts such as abstract data types, control-flow operators, functional-logic programming and constraints in an uniform way which is consistent with the idea of having three different inter-related semantics: operational, declarative and fix-point semantics. A program is interpreted in an indexed category where the base category contains all possible states which may be encountered during the execution of the program (for example, global constraints or type informations) while fibers encode the corresponding logical properties. We define appropriate concepts of categorical resolution and prove results of soundness and completeness. We are actually working on extending this framework to logic languages based on hereditary Harrop formulas.

Analysis of numerical properties of imperative languages. In [13, 16] we develop techniques which allow to specialize numerical abstract domains for a specific program. This method works by capturing an execution trace of the program, which is then analyzed trough statistical methods such as Principal Component Analysis and Independent Component Analysis. The result is a linear transformation of the space of the possible values of variables in the program. This transformation determines a set of linear constraints which are then fed to a template domain based on parallelotopes.

In [12] we show a prototype, written in the R language, of an analyzer which uses the techniques shown above. This software, called `RANDOM` (R-based ANalyzer for NUMerical DOMains) is available with an open source license a the URL `http://www.sci.unich.it/~amato/random/`.

In [19] we show a variant of the domain of parallelotopes where the constraint matrix is not fixed ahead of time, but varies during the analysis. The implementation of this new domain in `RANDOM` has been presented in [18]. Other theoretical

and experimental results on the domain of parallelotopes appear in [30].

Having `RANDOM` reached a complexity which is not easy to manage in the R language, we are actually working on a new static analyzer named `JANDOM` (JVM-based ANalyzer for NUMerical DOMains), written in the Scala language. The new analyzer has been presented for the first time in [20]. `JANDOM` is still in development, and it is available with open-source license on GitHub at the URL `https://github.com/jandom-devel/Jandom/`.

In [23] we consider the problem of removing a constraint from a polyhedron in double representation (constraints and generators). While this normally requires to recompute from scratch the representation based on generators, our work shows a new algorithm which, most of the times, is able to reuse the generators of the original polyhedron. This new algorithm is useful in the implementation of widening for the domain of polyhedra, but it may have other applications in the field of computational geometry.

In [24] we show a new domain combinator based on Minkowski's sum. Given two abstract domains A and B, an object in the abstract domain A+B is a pair made from an object of A and one of B, whose intended semantics is the Minkowski's sum of the two sets. Differently from other domain combinators, this one is (sometimes) able to improve precision of the analysis w.r.t. the one attainable with A and B separately without writing ad-hoc code. Some experiment using the sum of intervals and parallelotopes are promising.

In [25, 31] we study the operation of narrowing in template-based domains. We show that, in practice, infinite descending chains are quite rare. We also show that, when working with template domain with integer bounds, we may replace narrowing with intersection without compromising termination of the analysis. Based on this result, we show new narrowing operators for template-based abstract domains with rational bounds.

Static analysis of objects. In [26] we deal with sharing analysis, i.e., the problem of inferring all couples of variable which may be potentially bound to data structures with shared components. In this work we show how keeping track of linearity of variables improves the precision of sharing analysis, and we show the formal specification of a combined analysis of aliasing, sharing and linearity. We plan to implement this analysis within the tool `JANDOM`.

Weird numbers. A recent topic of interest is the search for weird numbers (natural numbers which are abundant but not semi-perfect). In trying to pursue the final aim of finding odd weird numbers, in [27] we show some techniques which allow us to find weird numbers with up to seven distinct prime factors. In [33] we develop a completely different approach which allows us to find primitive abundant and weird numbers with an extremely high count of primitive factors (up to 17) and many weird numbers with non square-free odd part.

Deep Learning. In [34, 37] we propose a multiple-komi modification of the AlphaGo Zero/Leela Zero paradigm. The winrate as a function of the komi is modeled with

a two-parameters sigmoid function, so that the neural network must predict just one more variable to assess the winrate for all komi values. A second novel feature is that training is based on self-play games that occasionaly branch – with changed komi – when the position is uneven. With this setting, reinforcement learning is showed to work on 7x7 Go and 9x9 Go, obtaining very strong playing agents.

Software development

Apart some prototypes strictly connected to a specific paper, I am the main developer of the following softwares:

Random (R-based Analyzer for Numerical Domains). It is a static and dynamic analyzer of programs based on the theory of abstract interpretation, described in [12] and [20]. `RANDOM` analyzes programs written in an imperative fragment of R, and implements the abstract domains of intervals, parallelotopes, template parallelotopes and their combinations. `RANDOM` combines static and dynamic analysis, using statistical techniques in the dynamic phase (Principal Component Analysis and Independent Component Analysis) and abstract interpretation in the static phase. `RANDOM` has been written in the R languages, and it is available with an open-source license at the URL `http://www.sci.unich.it/~amato/random/`.

Jandom (JVM-based Analyzer for Numerical Domains). It is a static analyzer of programs based on the theory of abstract interpretation, presented in [20]. `JANDOM` analyzes programs written in a simple imperative language, Java bytecode and linear transition systems. It implements the domains of intervals and parallelotopes, together with domains for the analysis of sharing and linearity of objects. Moreover, it may use the numerical domains provided by the Parma Polyhedra Library. Different techniques for solving data-flow equations have been implementing: localized widening, localized narrowing, delayed widening, and there is preliminary support for inter-procedural analysis. `JANDOM` also provides some domain combinators, such as reduced product and Minkowski's sum. The analyzer is written in Scala, and it is available with an open-source license on GitHub: `https://github.com/jandom-devel/Jandom/`.

ScalaFix. ScalaFix is the analysis engine of Jandom. It is a library written in Scala which implements several algorithms for solving fixpoint equations on partially ordered sets. The software is available with an open-source license on GitHub: `https://github.com/jandom-devel/ScalaFix/`.

SAI. SAI (and the related SAI-server) is a multiple-komi variant of LeelaZero, an open-source reimplementation of AlphaGo Zero. SAI implements many novel ideas, with the aim of obtaining an artificial intelligence able to play Go at a super-human level, not only in standard matches but also in the presence of handicap stones or non-standard komi. The main ideas implemented in SAI have been discussed in [34, 39, 37, 38]. The software is available on GitHub: `https://github.com/sai-dev`.

Universal Algebra in UniMath. This is a module implementing universal algebra in the UniMath theorem prover. The code is available on GitHub at the URL `https://github.com/amato-gianluca/UniMath` and is now largely part of the ufficial UniMath distribution. A preliminary version as been presented in [35].

Technical knowledge

Operating Systems. Very good knowledge of DOS, Windows and Unix, especially Linux on Intel architectures, both at the user and at the system administrator level.

Networks. Expert in the design and administration of computer network, even in mixed Windows–Unix environments. Very good knowledge of the Ethernet and TCP/IP protocol and of the most important protocols at application level.

Assembly Languages. Very good knowledge of the assembler for Intel 80x86 microprocessors. Good knowledge of the hardware architecture of a PC, and of the assembler for Zilog Z80 and MIPS microprocessors.

High Level Languages . Very good knowledge of the main programming paradigms (functional, imperative, logic, object oriented). Very good knowledge of the languages C, C++, Java, Scala, Pascal, R, PHP, Python. Good knowledge of Scheme Lisp, Prolog and Javascript. Basic knowledge of Basic and Ruby.

Typesetting languages. Good knowledge of HTML, CSS, TeX and LaTeX.

Database. Very good knowledge of the principles of relational databases, the language SQL, the DBMS MySQL, expecially in the context of web-based applications realized with the scripting language PHP or the Play framework. Good knowledge of MongoDB.

Publications

[1] G. Amato and G. Levi. "Properties of the lattice of observables in logic programming". In: *1997 Joint Conf. on Declarative Programming, APPIA-GULP-PRODE'97, Grado, Italy, June 16–19, 1997*. Ed. by M. Falaschi, M. Navarro, and A. Policriti. Grado, Italy, June 1997, pp. 175–187.

[2] G. Amato and G. Levi. "Abstract Interpretation Based Semantics of Sequent Calculi". In: *Static Analysis, 7th International Symposium, SAS 2000, Santa Barbara, CA, USA, June 29 – July 1, 2000, Proceedings*. Ed. by J. Palsberg. Vol. 1824. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2000, pp. 38–57. DOI: 10.1007/b87738.

[3] G. Amato. "Correct Answers for First Order Logic". In: *Declarative Programming — Selected Papers from AGP'00*. Ed. by A. Dovier, M. C. Meo, and A. Omicini. Vol. 48. Electronic Notes in Theoretical Computer Science. Elsevier, June 2001, pp. 45–64. DOI: 10.1016/S1571-0661(04)00149-5.

[4] G. Amato and J. Lipton. "Indexed Categories and Bottom-Up Semantics of Logic Programs". In: *Logic for Programming, Artificial Intelligence, and Reasoning, 8th International Conference, LPAR 2001 Havana, Cuba, December 3–7, 2001 Proceedings*. Ed. by R. Nieuwenhuis and A. Voronkov. Vol. 2250. Lecture Notes in Artificial Intelligence. Springer, 2001, pp. 438–454. DOI: 10.1007/3-540-45653-8_30.

[5] G. Amato and F. Spoto. "Abstract Compilation for Sharing Analysis". In: *Functional and Logic Programming, 5th International Symposium, FLOPS 2001 Tokyo, Japan, March 7–9, 2001 Proceedings*. Ed. by H. Kuchen and K. Ueda. Vol. 2024. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, Mar. 2001, pp. 311–325. DOI: 10.1007/3-540-44716-4_20.

[6] G. Amato and F. Scozzari. "Optimality in Goal-Dependent Analysis of Sharing". In: *Proceedings of the Joint Conference on Declarative Programming (AGP'02)*. Ed. by J. J. Moreno-Navarro and J. Mariño-Carballo. Madrid: Universidad Politécnica de Madrid, 2002, pp. 189–205.

[7] G. Amato and F. Scozzari. "A general framework for variable aliasing: Towards optimal operators for sharing properties". In: *Logic Based Program Synthesis and Transformation 12th International Workshop, LOPSTR 2002, Madrid, Spain, September 17–20, 2002. Revised Selected Papers*. Ed. by M. Leuschel. Vol. 2664. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2003, pp. 52–70. DOI: 10.1007/3-540-45013-0_6.

[8] G. Amato, M. Coppola, et al. "Modeling Web Applications by the Multiple Levels of Integrity Policy". In: *Proceedings of the International Workshop on Automated Specification and Verification of Web Sites (WWV 2005)*. Ed. by M. Alpuente, S. Escobar, and M. Falaschi. Vol. 157. Electronic Notes in Theoretical Computer Science 2. Elsevier, May 2006, pp. 167–185. DOI: 10.1016/j.entcs.2005.12.053.

[9] G. Amato, J. Lipton, and R. McGrail. "On the algebraic structure of declarative programming languages". In: *Theoretical Computer Science* 410.46 (2009), pp. 4626–4671. DOI: 10.1016/j.tcs.2009.07.038.

[10] G. Amato and F. Scozzari. "Observational Completeness on Abstract Interpretation". In: *Logic, Language, Information and Computation, 16th International Workshop, WoLLIC 2009, Tokyo, Japan, June 21–24, 2009*. Ed. by H. Ono, M. Kanazawa, and R. de Queiroz. Vol. 5514. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2009, pp. 99–112. DOI: 10.1007/978-3-642-02261-6_9.

[11] G. Amato and F. Scozzari. "Optimality in goal-dependent analysis of Sharing". In: *Theory and Practice of Logic Programming* 9.5 (Sept. 2009), pp. 617–689. DOI: 10.1017/S1471068409990111.

[12] G. Amato, M. Parton, and F. Scozzari. "A Tool Which Mines Partial Execution Traces to Improve Static Analysis". In: *First International Conference, RV 2010, St. Julians, Malta, November 1-4, 2010. Proceedings*. Ed. by H. Barringer and *et al.* Vol. 6418. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2010, pp. 475–479. DOI: 10.1007/978-3-642-16612-9.

[13] G. Amato, M. Parton, and F. Scozzari. "Deriving Numerical Abstract Domains via Principal Component Analysis". In: *17th International Symposium, SAS 2010, Perpignan, France, September 14-16, 2010, Proceedings*. Ed. by R. Cousot and M. Martel. Vol. 6337. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2010, pp. 134–150. DOI: 10.1007/978-3-642-15769-1.

[14] G. Amato and F. Scozzari. "On the interaction between sharing and linearity". In: *Theory and Practice of Logic Programming* 10.1 (Jan. 2010), pp. 49–112. DOI: 10.1017/S1471068409990160.

[15] G. Amato and F. Scozzari. "Observational Completeness on Abstract Interpretation". In: *Fundamenta Informaticae* 106.2–4 (2011), pp. 149–173. DOI: 10.3233/FI-2011-381.

[16] G. Amato, M. Parton, and F. Scozzari. "Discovering invariants via Simple Component Analysis". In: *Journal of Symbolic Computation* 47.12 (2012), pp. 1533–1560. DOI: 10.1016/j.jsc.2011.12.052.

[17] G. Amato and F. Scozzari. "Analysis and Verification of Navigation Strategies by Abstract Interpretation of Cellular Automata". In: *Motion in Games*. Ed. by M. Kallmann and K. Bekris. Vol. 7660. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2012, pp. 378–381. DOI: 10.1007/978-3-642-34710-8_37.

[18] G. Amato and F. Scozzari. "Random: R-Based Analyzer for Numerical Domains". In: *Logic for Programming, Artificial Intelligence, and Reasoning 18th International Conference, LPAR-18, Mérida, Venezuela, March 11-15, 2012. Proceedings*. Ed. by N. Bjørner and A. Voronkov. Vol. 7180. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2012, pp. 375–382. DOI: 10.1007/978-3-642-28717-6_29.

[19] G. Amato and F. Scozzari. "The abstract domain of parallelotopes". In: *Proceedings of the Fourth International Workshop on Numerical and Symbolic Abstract Domains, NSAD 2012*. Ed. by J. Midtgaard and M. Might. Vol. 287. Electronic Notes in Theoretical Computer Science. Elsevier, Nov. 2012, pp. 17–28. DOI: 10.1016/j.entcs.2012.09.003.

[20] G. Amato, S. Di Nardo Di Maio, and F. Scozzari. "Numerical static analysis with Soot". In: *Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program analysis*. SOAP '13. Seattle, USA: ACM, 2013, pp. 25–30. DOI: 10.1145/2487568.2487571.

[21] G. Amato and F. Scozzari. "Localizing widening and narrowing". In: *Static Analysis. 20th International Symposium, SAS 2013, Seattle, WA, USA, June 20-22, 2013, Proceedings*. Ed. by F. Logozzo and M. Fähndrich. Vol. 7935. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2013, pp. 25–42. DOI: 10.1007/978-3-642-38856-9_4.

[22] G. Amato and F. Scozzari. "Optimal multibinding unification for sharing and linearity analysis". In: *Theory and Practice of Logic Programming* 14 (03 2014), pp. 379–400. DOI: 10.1017/S1471068413000070.

[23] G. Amato, F. Scozzari, and E. Zaffanella. "Efficient Constraint/Generator Removal from Double Description of Polyhedra". In: *Fifth International Workshop on Numerical and Symbolic Abstract Domains (NSAD)*. Ed. by A. Simon and A. Venet. Vol. 307. Electronic Notes in Theoretical Computer Science. Elsevier, 2014, pp. 3–15. DOI: 10.1016/j.entcs.2014.08.002.

[24] G. Amato, S. Di Nardo Di Maio, and F. Scozzari. "Sum of Abstract Domains". In: *NASA Formal Methods, 7th International Symposium, NFM 2015, Pasadena, CA, USA, April 27-29, 2015, Proceedings*. Ed. by G. H. Klaus Havelund and R. Joshi. Vol. 9058. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2015, pp. 35–49. DOI: 10.1007/978-3-319-17524-9_4.

[25] G. Amato, S. Di Nardo Di Maio, et al. "Narrowing operators on template abstract domains". In: *FM 2015: Formal Methods, 20th International Symposium, Oslo, Norway, June 24-26, 2015, Proceedings*. Ed. by N. Bjøner and F. de Boer. Vol. 9109. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2015, pp. 57–72. DOI: 10.1007/978-3-319-19249-9_5.

[26] G. Amato, M. C. Meo, and F. Scozzari. "Exploiting linearity in sharing analysis of object-oriented programs". In: *Proceedings of ICTCS 2015, the 16th Italian Conference on Theoretical Computer Science*. Ed. by P. Crescenzi and M. Loreti. Vol. 322. Electronic Notes in Theoretical Computer Science. Elsevier, Apr. 2015, pp. 3–18. DOI: 10.1016/j.entcs.2016.03.002.

[27] G. Amato, M. Haslet, et al. "Primitive weird numbers having more than three distinct prime factors". In: *Rivista di Matematica della Università di Parma* 7.1 (2016), pp. 153–163.

[28] G. Amato, F. Scozzari, et al. "Efficiently intertwining widening and narrowing". In: *Science of Computer Programming* 120 (May 2016), pp. 1–24. DOI: 10.1016/j.scico.2015.12.005.

[29] G. Amato and M. Rubino. "Experimental Evaluation of Numerical Domains for Inferring Ranges". In: *Electronic Notes in Theoretical Computer Science* (2017), pp. 3–16. DOI: 10.1016/j.entcs.2018.03.002.

[30] G. Amato, M. Rubino, and F. Scozzari. "Inferring Linear Invariants with Parallelotopes". In: *Science of Computer Programming* 148 (2017), pp. 161–188. DOI: 10.1016/j.scico.2017.05.011.

[31] G. Amato, S. Di Nardo Di Maio, et al. "Descending chains and narrowing on template abstract domains". In: *Acta Informatica* 55.6 (2018), pp. 521–545. DOI: 10.1007/s00236-016-0291-0.

[32] G. Amato, M. C. Meo, and F. Scozzari. "A Taxonomy of Program Analyses". In: *Proceedings of the 19th Italian Conference on Theoretical Computer Science, Urbino, Italy, September 18-20, 2018*. Ed. by A. Aldini and M. Bernardo. Vol. 2243. CEUR Workshop Proceeding. 2018, pp. 213–217. URL: http://ceur-ws.org/Vol-2243/paper21.pdf.

[33] G. Amato, M. Hasler, et al. "Primitive abundant and weird numbers with many prime factors". In: *Journal of Number Theory* 201 (2019), pp. 436–459. DOI: 10.1016/j.jnt.2019.02.027.

[34] F. Morandin, G. Amato, et al. "SAI a Sensible Artificial Intelligence that plays Go". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. July 2019, pp. 1–8. DOI: 10.1109/IJCNN.2019.8852266.

[35] G. Amato, M. Maggesi, et al. "Universal Algebra in UniMath". In: *Workshop on Homotopy Type Theory/ Univalent Foundations*. 2020.

[36] G. Amato, M. C. Meo, and F. Scozzari. "On collecting semantics for program analysis." In: *Theoretical Computer Science* 823 (2020), pp. 1–25.

[37] F. Morandin, G. Amato, et al. "SAI: a Sensible Artificial Intelligence that plays with handicap and targets high scores in 9x9 Go". In: *Proceedings of the 2020 European Conference on Artifical Intelligence, ECAI 2020*. Vol. 325. Frontiers in Artificial Intelligence and Applications. Amsterdam, The Netherlands: IOS Press, 2020, pp. 403–410.

[38] F. Morandin, G. Amato, et al. *SAI, a Sensible Artificial Intelligence that plays Go*. arXiv: 1809.03928 [cs].

[39] F. Morandin, G. Amato, et al. *SAI: a Sensible Artificial Intelligence that plays with handicap and targets high scores in 9x9 Go (extended version)*. arXiv: 1905.10863 [cs].

Other scientific work

[40] G. Amato. "Proprietà del reticolo degli osservabili in programmazione logica". In italian. MA thesis. Università di Pisa, July 1996.

[41] G. Amato. "Sequent Calculi and Indexed Categories as a Foundation for Logic Programming". PhD thesis. Dipartimento di Informatica: Università di Pisa, Mar. 2001.

Pescara, 2022-09-05