Part B

START PAGE

MARIE SKŁODOWSKA–CURIE ACTIONS

Individual Fellowships (IF)
Call: H2020-MSCA-IF-2014

PART B

"CAPTA"

"Complexity through Automata, Proof Theory and Algebra"

This proposal is to be evaluated as:

[**Standard EF**]

Contents

List of Participants

Participants	Legal Entity Short Name	Academic (tick)	Non-academic (tick)	Country	Dept. / Division / Laboratory	Supervisor
University of Copenhagen	UCPH	✓		Denmark	Dept. of Computer Science	Prof. Jakob Grue Simonsen

START PAGE COUNT

1 Summary

COMPLEXITY THROUGH AUTOMATA, PROOF THEORY AND ALGEBRA (CAPTA) will provide implicit and intentional definitions of complexity classes, by taking intuitions and tools from Linear Logic and the Geometry of Interaction program, using strong links between classes of complexity and automata theory that have not been properly exploited yet. Abstract definitions of classes of complexity are given by observing the dynamics resulting from the interaction of programs with a specific kind of input, and by identifying the algebra where their interpretations live. Whereas all the previous results in Implicit Computational Complexity (ICC) provided extensional machine-independent characterizations by restricting the syntax, CAPTA will push the abstraction, the 'dematerialisation', further by intentionally quotienting programs of the same complexity.

The aims of CAPTA are to 1) provide semantic and implicit characterization of complexity classes in algebraic terms, a new tool to attack classical problems of inclusion and separation among complexity classes; 2) switch from a predicative to a functional setting, to develop a 'built-in' compositionality that abstracts away painful composition of space-limited programs; and 3) extract new syntactical restrictions on programming languages and type systems thanks to a 'reverse engineering' in the spirit of Geometry of Interaction.

It differs from previous approaches by evacuating the syntax and using results on automata that were never considered from an ICC perspective. It connects communities working in ICC with different approaches, Linear Logic for the French and Italian side, descriptional and algebraic complexity for the Danish and German side.

2 Excellence

2.1 Quality, innovative aspects and credibility of the research (including inter/multidisciplinary aspects)

2.1.1 Introduction, state-of-the-art, objectives and overview of the action

Background Computational complexity theory is a discipline at the intersection of mathematical logic and computer science that studies the fundamental limits on efficiency of solving problems by computers. This complexity is traditionally measured 'externally' by the space or time resources needed by an abstract machine (Turing machine, random access machine, etc.) to compute the answer to this problem.

Implicit Computational Complexity (ICC) aims at characterizing complexity classes by bounding a programming language without being tied to a model of computation, being a concrete piece of hardware (Turing machine) or formalism (lambda-calculus). This ancient approach[1] traditionally imposes constrains to a programming language[2], so that all the programs written in it are exactly of such complexity. If one consider finite model theory to be a programming language, then ICC dates back to the introduction of descriptive complexity[3], that characterizes complexity classes by the type of logic needed to express languages in them.

[1] A. Cobham. 'The intrinsic computational difficulty of functions'. In: *Logic, methodology and philosophy of science: Proceedings of the 1964 international congress held at the Hebrew university of Jerusalem, Israel, from August 26 to September 2, 1964*. Ed. by Y. Bar-Hillel. Studies in Logic and the foundations of mathematics. North-Holland Publishing Company, 1965, pp. 24–30.

[2] S. J. Bellantoni and S. A. Cook. 'A New Recursion-Theoretic Characterization of the Polytime Functions'. In: *Computational Complexity* 2 (1992), pp. 97–110; D. Leivant. 'Stratified Functional Programs and Computational Complexity'. In: *Conference Record of the Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. Ed. by M. S. Van Deusen and B. Lang. Charleston, South Carolina, USA: ACM Press, 1993, pp. 325–333.

[3] R. Fagin. 'Generalized First-Order Spectra and Polynomial-Time Recognizable Sets'. In: *Complexity of Computation*. Ed. by R. M. Karp. Vol. 7. SIAM-AMS Proceedings. American Mathematical Society, 1974, pp. 27–41.

Linear Logic[4] studies proofs by taking into account the resources needed to derive formulas, and hence the complexity of computing the normal form of a proof, the result of a program according to the Curry-Howard correspondence. Fragments of Linear Logic were proven to capture several complexity classes, by bounding the resources needed to prove a formula[5], by constraining the types[6], by studying the geometrical properties of parallel expression of proofs[7], to name a few.

All those approaches agree that ICC is a perspective where 'classes are not given by constraining the amount of resources a machine is allowed to use, but rather by imposing linguistic constraints on the way algorithms are formulated.'[8] However, the *quality* of the resources can also be constrained: for instance if the tapes of a Turing machine are read-only, one gets a characterization of **L** (log-space)[9], and if one add a 'Last-in First-Out' pushdown tape to the model, one gets a characterization of **P** (polynomial time)[10]: this focus was studied in automata theory.

Limitations The first ICC characterizations[11] obtained by constraining programming languages allowed to get rid of models of computation, providing a step towards an abstract definition of complexity classes. But syntactical restrictions prevent from making any quotient on the programs, from extracting any generality about the classes captured or from getting rid of the algorithmic complexity of log-space composition[12].

The lack of attention paid to the inputs is also an issue: they are a parameter of the complexity as well as the programs, but are not considered as such. For instance, the definition of a 'reasonable way of representing the inputs' is circular: for a program, any two representations of the input are equivalent *as long as a program of the same complexity can transform one in the other*. A recent outburst[13] shaken the folkloric knowledge inherited from automata[14] that 'pointers are **L** ', by proving that this equality was wrong in the case of partially ordered inputs.

The dialogue between complexity and automata offered numerous results[15] that were never considered from an ICC perspective. However, as the resources limitations of automata are 'built-in' —limited number of heads, additional pushdown-stack, etc.— there is no need to constrain their programming language: capturing the semantics of the execution of a specific model immediately provides a characterization of the corresponding complexity class.

[4] J.-Y. Girard. 'Linear logic'. In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101.

[5] U. Dal Lago and M. Hofmann. 'Bounded Linear Logic, Revisited'. In: *Logical Methods in Computer Science* 6.4 (2010).

[6] P. Baillot and K. Terui. 'Light types for polynomial time computation in lambda calculus'. In: *Information and Computation* 207.1 (2009), pp. 41–62.

[7] P. Baillot and D. Mazza. 'Linear Logic by Levels and Bounded Time Complexity'. In: *Theoretical Computer Science* 411.2 (2010), pp. 470–503; C. Aubert. 'Sublogarithmic uniform Boolean proof nets'. In: *Proceedings Second Workshop on Developments in Implicit Computational Complexity*. Ed. by J.-Y. Marion. Vol. 75. Electronic Proceedings in Theoretical Computer Science. 2011, pp. 15–27.

[8] U. Dal Lago. 'A Short Introduction to Implicit Computational Complexity'. In: *Lectures on Logic and Computation - ESSLLI 2010 Copenhagen, Denmark, August 2010, ESSLLI 2011, Ljubljana, Slovenia, August 2011, Selected Lecture Notes*. Ed. by N. Bezhanishvili and V. Goranko. Vol. 7388. Lecture Notes in Computer Science. Springer, 2011, pp. 89–109, p. 90.

[9] J. Hartmanis. 'On Non-Determinancy in Simple Computing Devices'. In: *Acta Informatica* 1.4 (1972), pp. 336–344.

[10] S. A. Cook. 'Characterizations of Pushdown Machines in Terms of Time-Bounded Computers'. In: *Journal of the ACM* 18.1 (1971), pp. 4–18.

[11] Bellantoni and Cook, 'A New Recursion-Theoretic Characterization of the Polytime Functions'; Leivant, 'Stratified Functional Programs and Computational Complexity'; N. D. Jones. 'LOGSPACE and PTIME Characterized by Programming Languages'. In: *Theoretical Computer Science* 228.1–2 (1999), pp. 151–174.

[12] U. Dal Lago and U. Schöpp. 'Functional Programming in Sublinear Space'. In: *Programming Languages and Systems, 19th European Symposium on Programming, ESOP 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*. Ed. by A. D. Gordon. Vol. 6012. Lecture Notes in Computer Science. Springer, 2010, pp. 205–225.

[13] M. Hofmann and U. Schöpp. 'Pure Pointer Programs with Iteration'. In: *ACM Transactions on Computational Logic* 11.4 (2010).

[14] Hartmanis, 'On Non-Determinancy in Simple Computing Devices'.

[15] M. A. Harrison and O. H. Ibarra. 'Multi-Tape and Multi-Head Pushdown Automata'. In: *Information and Control* 13.5 (1968), pp. 433–470; B. Monien. 'Two-Way Multihead Automata Over a One-Letter Alphabet'. In: *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications* 14.1 (1980), pp. 67–82.

New challenges CAPTA will bridge those gaps thanks to a focus on the dynamics of inputs and programs that connect together execution (for automata), normalisation (for proofs), execution formula (for operators) and resolution (for logic programs). This ambitious correspondence is tight together through algebraic structures that can indifferently represent any of these objects and hence account for the complexity of programs. This implicit characterizations of complexity classes by semantics means allows to 1) quotient programs and compare complexity classes; 2) take inputs as parameters of the complexity; and to 3) extract syntactical rules.

Quotienting by defining complexity classes as sets of counter-tests: a program represented algebraically (an 'observation') belongs to such class if it interacts with a certain kind of input. This intentional behaviour of the observations is reflected in the structure of the associated mathematical object (a von Neumann algebra[16] or the unification semiring[17]). This yields abstract definitions of time-, space-bounded as well as parallel complexity classes in algebraic terms, suitable for new techniques of comparison. From the definition, compositions of functional observations in an algebra belong to that algebra, avoiding tedious algorithms for compositing space-bounded programs.

Taking the input as a part of the definition is possible following the Church (functional) representation of data types (integers, trees, etc.), that considers inputs as programs dialoguing with algorithms. This breaks the circularity in the equivalence between representations and enforce that different inputs (1-way[18], partially[19] or totally ordered, etc.) yields different classes. This is a second step towards a mathematical and totally machine-independent definition of complexity classes, that focus on the dynamics of execution.

Giving new insights on Term Rewriting System (TRS) : for instance, seeing **L** as programs making permutations on circular binary words[20] permits to develop an innovative constrain on the syntax of logic programs[21]. It is also tackled the question of the undecidability of the boundedness problem: whereas determining the fixed point of a logic program independently of the facts inputs is impossible in general[22], precise bounds are known for 'balanced' programs[23]. Such journeys will be reproduced for other complexity classes.

2.1.2 Research methodology and approach

New tools The Geometry of Interaction (GoI) is a project that builds a fully mathematical model of execution (cut-elimination) starting from its dynamics. Proofs, and hence programs, are represented by first-order term[24] or operators[25] depending on the desired orientation: logic programming or mathematical

[16] C. Aubert and T. Seiller. 'Logarithmic Space and Permutations'. In: *Arxiv preprint* abs/1301.3189 (2013). arXiv: 1301.3189 [cs.LO].

[17] C. Aubert et al. 'Logic Programming and Logarithmic Space'. In: *Arxiv preprint* abs/1406.2110 (2014). arXiv: 1406.2110 [cs.LO]. Accepted to APLAS 2014.

[18] S. Ginsburg, S. A. Greibach, and M. A. Harrison. 'One-way stack automata'. In: *Journal of the ACM* 14.2 (1967).

[19] Hofmann and Schöpp, 'Pure Pointer Programs with Iteration'.

[20] C. Aubert and T. Seiller. 'Characterizing co-NL by a group action'. In: *Arxiv preprint* abs/1209.3422 (2012). arXiv: 1209.3422 [cs.LO].

[21] C. Aubert and M. Bagnol. 'Unification and Logarithmic Space'. In: *Rewriting and Typed Lambda Calculi - Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*. Ed. by G. Dowek. Vol. 8650. Lecture Notes in Computer Science. Springer, 2014, pp. 77–92; Aubert et al., 'Logic Programming and Logarithmic Space'.

[22] G. G. Hillebrand et al. 'Undecidable Boundedness Problems for Datalog Programs'. In: *Journal of Logic Programming* 25.2 (1995), pp. 163–190.

[23] Aubert et al., 'Logic Programming and Logarithmic Space'.

[24] J.-Y. Girard. 'Geometry of interaction III: accommodating the additives'. In: *Advances in Linear Logic*. Ed. by J.-Y. Girard, Y. Lafont, and L. Regnier. London Mathematical Society Lecture Note Series 222. Cambridge University Press, 06/1995, pp. 329–389.

[25] J.-Y. Girard. 'Geometry of Interaction V: logic in the hyperfinite factor'. In: *Theoretical Computer Science* 412.20 (04/2011): *Girard's Festschrift*. Ed. by T. Ehrhard, C. Faggian, and O. Laurent, pp. 1860–1883.

abstraction. The characterization of **L** and **NL** in a series of articles[26] written with young co-authors pushes close to ICC perspectives this line of work suggested by Girard[27].

Complexity classes being defined intentionally as sets of programs interacting with a specific representation of inputs, their study is one of the core of the action. I will focus on classical variations in automata regarding the inputs, the cardinality of their alphabet, their number of ways, and imagine what would be non-deterministic, trees-like or graphs inputs.

CAPTA will pay an extra attention to previous results obtained in automata theory: they offer a solid ground to represent the semantics of programs in algebras, and thanks to transducers, to switch to a functional setting. By inspecting the variety of results and models in automata theory, CAPTA will seamlessly find the appropriate target to be embedded in an algebra, thus providing algebraic definitions of complexity classes.

A growing inter/multi-disciplinary community Whereas automata theory is a subject for a large variety of researchers, the algebraic considerations on ICC mixing TRS can only be found in the University of Copenhagen (UCPH), in the European state-of-the-art 'Complexity via Logic and Algebra' (COLA) project, currently led in Copenhagen by the scientist in charge. A particular attention will be paid to the progress of J. Frey, a recent post-doc at the Department of Computer Science at the University of Copenhagen (DIKU) that study complexity theory using topos-theoretic characterizations.

The presence of T. Heindel, leader of the Rubyx project funded by the Marie-Curie Action and supposed to start in october 2014 at DIKU, will connect CAPTA to biological and graph-theoretical perspectives through common interest in complexity that dates back to the time we were co-workers during the Implicit Computational Complexity, Concurrency and Extraction (Complice) project.

2.1.3 Originality and innovative aspects of the research programme

Originality CAPTA reverses the classical ICC approach by capturing programs intentionally and then abstracting syntactical rules from them. It re-introduces fundamentals tools such as automata, that gave an underestimated series of results, and brings back the question of the input, that is merely debated among complexity theorists.

Innovation CAPTA uses latest developments in Geometry of Interaction to connect automata and logic programs, through an ambitious correspondence between proofs, programs and mathematical structures such as von Neumann algebras and topoi. These are state-of-the-art subjects in Mathematics, that just begin to be considered as pertinent objects to study ICC.

2.2 Clarity and quality of transfer of knowledge/training for the development of the researcher in light of the research objectives

2.2.1 Training

The training will bring raw material for follow-ups as well as experience in the following areas:

Algebraic characterization of complexity classes[28] I will be guided by the scientist in charge through the Ph.D. courses dispensed by DIKU and consider attending courses on Topos, C*-algebras and Categories and Topology (between 20 and 30 hours each), in the first six months of the action. In parallel, during the two years of the action, an individual training through research with the scientist in charge and its co-authors will be organized to connect those objects to their usage in complexity theory.

[26] Aubert and Seiller, 'Characterizing co-NL by a group action'; Aubert and Seiller, 'Logarithmic Space and Permutations'; Aubert and Bagnol, 'Unification and Logarithmic Space'; Aubert et al., 'Logic Programming and Logarithmic Space'.

[27] J.-Y. Girard. 'Normativity in Logic'. In: *Epistemology versus Ontology*. Ed. by P. Dybjer et al. Vol. 27. Logic, Epistemology, and the Unity of Science. Springer, 2012, pp. 243–263.

[28] One full work package is dedicated to this ambitious training, see p. 12.

Fundamental results in automata theory Among others, R. Glück and H. B. Axelsen from DIKU's Algorithms and Programming Languages are currently working on fundamental results on automata, even restating a fundamental result CAPTA will use[29]. They will guide me in the literature and recommend courses if needed.

Project management and supervision of students I will attend the Graduate School of Science's workshop on Project Leadership (2 days) and Teaching development courses (175 hours), in addition to the Faculty of Science's bi-yearly course aimed at assistant professors that will teach me university pedagogics as preparation for a future permanent position. I will learn project management by co-managing CAPTA with the scientist in charge and regularly discussing planning and risk management.

2.2.2 Knowledge transfers

Communities and networks Europe has a strong leadership in ICC: community independently appeared in France, Italy, Austria and Germany, each with their own specialities and techniques (Linear Logic, Term Rewriting System, Algebraic characterization, Quasi-interpretation, etc.). Even if the language, problems, workshops[30] and journals[31] are common, they still have difficulties to learn from each other. CAPTA gathers themes at the intersection of their approaches, combining proof theory, algebraic perspectives and TRS.

I know well the orientations taken in France and Italy, through collaborations and discussions with J.-Y. Moyen, S. Guerrini (LIPN, Paris 13), U. Dal Lago (INRIA, Bologna) and J.-Y. Marion (LORIA, Université de Lorraine). I will have access to the scientist in charge's network of ICC researchers using programming-language approaches: L. Kristiansen (Department of Informatics, Oslo), G. Moser (Institute of Computer Science, Innsbruck), F. Henglein (DIKU, Copenhagen), head of the *Kleene Meets Church* project.

Specialized knowledge I rapidly became an expert in ICC and specifically in their interactions with Linear Logic and Geometry of Interaction, from the 'syntactical'[32] (extensional) as well as the 'semantical'[33] (intentional) point of view. CAPTA will make of me an expert in the interaction between automata, proof and complexity theories, an intersection left aside in the ICC community, including but not limited to the host university.

Transfer of knowledge This knowledge will be disseminated to the host institution through seminars and interaction with the members of the COLA project as well as by co-publications: a community of interest already exists and will be kept tight by the plurality of approaches its members represents (Term Rewriting System for the scientist in charge, dynamic of execution for T. Heindel, complexity theory using topos-theoretic characterizations for J. Frey). Those members have a basic knowledge regarding Geometry of Interaction that will be extended thanks to CAPTA.

2.3 Quality of the supervision and the hosting arrangements

The Department of Computer Science at the University of Copenhagen (DIKU) was founded in 1970 by later Turing Award Winner P. Naur. The department hosts undergraduate and graduate studies in

[29] R. Glück. 'Simulation of Two-Way Pushdown Automata Revisited'. In: *Festschrift for Dave Schmidt*. Ed. by A. Banerjee et al. Vol. 129. Electronic Proceedings in Theoretical Computer Science. 2013, pp. 250–258.

[30] As 'Developments in Implicit Computational Complexity' (DICE), a part of European Joint Conferences on Theory and Practice of Software.

[31] For instance the special issues in ACM Transactions on Computational Logic (August 2009, Volume 10 Issue 4), Theoretical Computer Science (2004, Volume 318, Numbers 1-2), and to come in Information & Computation.

[32] Aubert, 'Sublogarithmic uniform Boolean proof nets'.

[33] Aubert and Seiller, 'Characterizing co-NL by a group action'; Aubert and Seiller, 'Logarithmic Space and Permutations'; Aubert and Bagnol, 'Unification and Logarithmic Space'; Aubert et al., 'Logic Programming and Logarithmic Space'.

the fields of Algorithms and Programming Languages, Image Processing and Machine Learning, and Human-Centered Computing, with local interdisciplinary collaboration with the departments of Physics, Mathematics, Economics, and the Department of Media, Cognition and Communication, as well as a number of international research collaborations.

The department hosts 32 permanent scientific staff members, and approximately 25 post docs, and 35 Ph.D. students, and has a sustained record of academic excellence evident both by international research awards (e.g., the Turing award and the ACM Programming Languages Achievement Award) and the consistently high international profile of its scientific alumni (e.g. the vice chancellor of the Danish IT University and the Director of the Fraunhofer Institute for Software and Systems Engineering in Dortmund).

I will be part of the research group around the scientist in charge (currently consisting of two post docs, two Ph.D. students and a research programmer), and will like the scientist in charge be formally embedded in both the Human-Centered Computing Group (providing office space, ancillary staff, and other local physical support), and the Algorithms and Programming Languages Group (providing scientific sparring and mentoring). Both research groups have an extended international track record, having collectively published more than two hundred papers in the last five years at top international venues such as ACM POPL and ACM STOC.

2.3.1 Qualifications and experience of the supervisor (s)

The scientist in charge is an internationally recognized expert on the mathematical treatment of computation and programming languages and has published a substantial number of papers in highly respected scholarly journals in Theoretical Computer Science (TCS) such as ACM Transactions on Computational Logic, Information and Computation, and Theory of Computing Systems.

The scientist in charge has published extensively in the area of term rewriting and logic, particularly concerning superrecursive complexity hierarchies, and has a track record of publications in computational complexity as applied to dynamical systems[34], and has ongoing, sustained research collaborations with several international experts from the area of computational complexity (notably at the Academic College Tel-Aviv Yafo, the Department of Mathematics at the University of Oslo, and the Department of Computer Science at the University of Innsbruck). The scientist in charge holds a Ph.D. degree from the University of Copenhagen and is one of only four scientists at the Department of Computer Sicence to hold a habilitation degree, and has been the supervisor of more than thirty-five master's students, four doctoral students, and a post-doctoral scholar (with two more post-doctoral scholars joining his group in late 2014); the scientist in charge has ample management experience, currently serving as the Department of Computer Science's Head of Research, and holding a master's degree in business administration from Heriot-Watt University. The scientist in charge is the recipient of the Danish government's Sapere Aude Elite Research Leader grant and is currently principal investigator of the 800.000€ COLA project in implicit complexity theory.

2.4 Capacity of the researcher to reach and re-enforce a position of professional maturity in research

I hold a Licence (Bachelor's degree) in History (Univ. Champagne-Ardennes), a Licence and a first year master's in Philosophy (Paris 1), a master's degree in Mathematics (Paris 7) and a Ph.D. in Computer

[34] S. B. Andersen and J. G. Simonsen. 'Term Rewriting Systems as Topological Dynamical Systems'. In: *23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan*. Ed. by A. Tiwari. Vol. 15. Leibniz International Proceedings in Informatics. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012, pp. 53–68; A. M. Ben-Amram, N. H. Christensen, and J. G. Simonsen. 'Computational Models with No Linear Speedup'. In: *Chicago J. Theor. Comput. Sci.* 2012 (2012); Andersen and Simonsen, 'Term Rewriting Systems as Topological Dynamical Systems'; D. de Carvalho and J. G. Simonsen. 'An Implicit Characterization of the Polynomial-Time Decidable Sets by Cons-Free Rewriting'. In: *Rewriting and Typed Lambda Calculi - Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*. Ed. by G. Dowek. Vol. 8650. Lecture Notes in Computer Science. Springer, 2014, pp. 179–193.

Science (Paris 13)[35]. I went from Humanities to Mathematics and Theoretical Computer Science (TCS) thanks to their connections drawn by Logic and grasped a hard subject where I had no prior knowledge (the von Neumann Algebra) during the writing of my Ph.D thesis[36]. I also began to study automata theory by myself and my Ph.D. thesis contains material for a paper on the links between automata and complexity classes that already exists as a technical report.

I am an active member of the TCS community with the writing of 5 papers[37], in collaboration with Ph.D. students or recent Ph.D. laureate. None of them were written with my supervisors, one of them is single-authored, and they were published in the most prestigious conference in Term Rewriting System (RTA-TLCA), the most important workshop dedicated to ICC (DICE), and one of them was recently accepted to one of the top Computer Science conference in Asia (APLAS). Two long papers (28 and 32 pages) are submitted to first-class journals (MSCS and Inform. & Comput.).

In four years, I was invited about ten times to seminars in Europe, and took every opportunity I had to make talks in workshops and conferences. I am an active member of seven projects funded by the *Agence Nationale de la Recherche* and several young researchers' projects.

While I am still a very young Ph.D.-age (I began my Ph.D. in 2010 and defended it 3 years later), I have been actively pursuing and developing a research programme of my own, and starts to reap some fruits from it, evidenced by submissions and preprints. This programme will drive me to a state of full maturity for a permanent position in Europe: a fellowship at DIKU will widen my themes by expanding them toward TRS and descriptional complexity, expand my network with an international experience and a new community, and sharpen my decisional capacities by giving me the opportunity to co-manage the project in an outstanding environment.

3 Impact

3.1 Enhancing research- and innovation-related human resources, skills, and working conditions to realise the potential of individuals and to provide new career perspectives

Scientific impact CAPTA will boost the COLA project, started in 2012, by providing, thanks to Linear Logic, different perspectives on complexity-related problems, as representation of inputs and compositionnality of programs. It will take the most out of automata theory and to transfer it to ICC, thus giving new perspectives to its community: the striking point of CAPTA's technique is its reproducibility. It is easy to look for an appropriate model (such that evolving graph-structures[38]) that entails a specific bound and represent it in an algebraic setting. Such variations on CAPTA's plan will considerably increase the common knowledge on complexity classes by providing classes of equi-expressive algebras.

Realizing the potential I always worked with co-authors of the same seniority as me: while this group of young researchers have innovative and ambitious ideas, it still lacks of experience in terms of quality of presentations, organization and advanced knowledge in complexity theory. I have close interactions with the members of my Ph.D. panel, but my network is essentially French-centred and focused on Linear Logic, while my research activities have broaden that scale.

By the end of CAPTA, I will have acquired an experience that has 1) widen my network, giving it an European scale, 2) allowed me to work in a mid-scale group of researchers with a dynamics of publication,

[35] The detail is provided in my resume p. 15.

[36] C. Aubert. 'Linear Logic and Sub-polynomial Classes of Complexity'. PhD thesis. Université Paris 13–Sorbonne Paris Cité, 11/2013.

[37] Aubert, 'Sublogarithmic uniform Boolean proof nets'; Aubert and Seiller, 'Characterizing co-NL by a group action'; Aubert and Seiller, 'Logarithmic Space and Permutations'; Aubert and Bagnol, 'Unification and Logarithmic Space'; Aubert et al., 'Logic Programming and Logarithmic Space'.

[38] D. Leivant and J.-Y. Marion. 'Evolving Graph-Structures and Their Implicit Computational Complexity'. In: *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II.* ed. by F. V. Fomin et al. Vol. 7966. Lecture Notes in Computer Science. Springer, 07/2013, pp. 349–360.

3) benefited form advanced researchers to learn to supervise students.

3.2 Effectiveness of the proposed measures for communication and results dissemination

3.2.1 Communication and public engagement strategy of the action

The results of CAPTA will be theoretical, and made visible to experts and the general public using modern media, with the help of the permanent full-time staff member of DIKU and the communications unit of the Faculty of Science (4 members) aiding with communication to the public. More precisely, the dynamics of execution between inputs and representation of automata is perfectly suited to develop toy-examples that are easily presented by web-based animations and simple drawings. Apart from its playful dimension, such representations of dynamics systems catch the attention by representing how the quality of resources and the presentation of inputs alter the computational capacities.

Simple presentations will be made public, for instance during the Danish *Forskningens Døgn*[39], and then included in conference's slides: the reception of the general public will be a benchmark for the accessibility of the presentation, that will be used in return to catch the attention of experts.

I will pursue the redaction of small notes on my website regarding all dimensions of my activities and improve the Wikipedia entries relative to ICC, so that CAPTA will be visible to experts and the general public alike. My experiences as volunteer in cultural and technological mediation will be valuable skills in those tasks.

3.2.2 Dissemination of the research results

A publication is to be expected for every work package (detailed p. 12) except for one of the work package dedicated to my training (W.P. 1.1).

The characterization of polynomial time (**P**, W.P. 1.2) is the typical example of result deserving rapid communication in conference, to draw attention on CAPTA and attest for its appropriateness: a wide and visible conference as Logic In Computer Science (LICS) is a reasonable objective.

On parallel, some long-standing work packages cannot be delivered as compact participations to proceedings: for instance, the study of variations of automata (W.P. 1.3) will produce a survey with a comprehensive bibliography that will be submitted to a first-class and open journal as Theory of Computing. Such a publication manifests the renewal of interest in those results and, by its visibility and exhaustiveness, attracts precious comments on results I could have missed.

Partial results and achievements will be proposed to pertinent workshops as part of the premier venues for research on TCS (ETAPS, RTA or LICS): they provide rapid and pertinent feedbacks, test the reception of the whole enterprise.

3.2.3 Exploitation of results and intellectual property

The research plan being theoretical, no direct exploitation will be developed. If any application were to happen, I would consult UCPH's tech transfer office. The research material will always be public, under a free-licence if possible, on my web-page as well as on arXiv and hal, following EU's Horizon 2020 policies regarding open access.

4 Implementation

4.1 Overall coherence and effectiveness of the work plan, including appropriateness of the allocation of tasks and resources

A modular approach in complexity… The starting point of CAPTA is to extend previous results to other variations of automata, and thus to other complexity classes. The two natural targets are adding a

[39] I already participated to a French equivalent, *Savante banlieue*, in the *Plaine Commune*.

stack to the model (W.P. 1.2) and parallel execution (W.P. 2.2), providing algebras representing the most important classes of complexity (**P**, **AC**, **NC**).

…inspired by automata… This modularity will be supported by the study of the variations of automata (W.P. 1.3): depending on the quality of the space-storage (read-only, pushdown stack, pushdown store, counter, etc.) and the input (cardinal of the alphabet, number of ways, sweeping, oblivious, rotating, etc.), one gets a myriad of variations that corresponds to different complexity classes and will be embedded in algebras seamlessly. This package will lead to the study of transducers (functional automata) and initiate the research on the functional setting (W.P. 2.1) with solid basis.

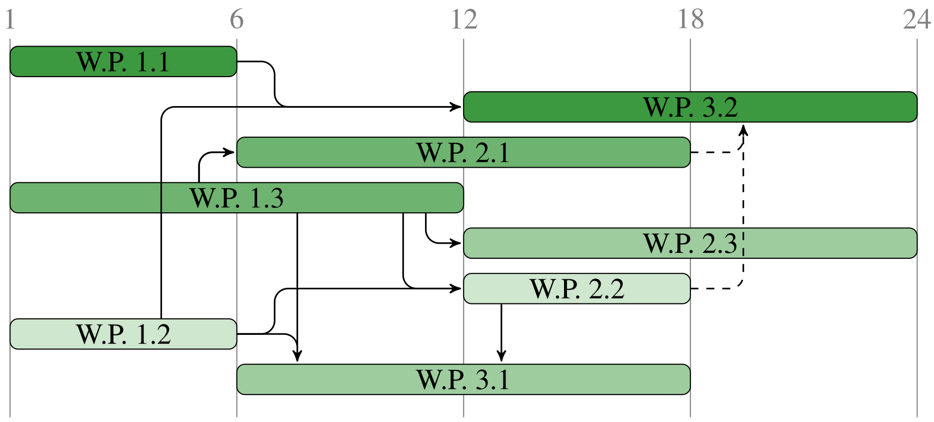
…that provides new insights on ICC… Once different automata are embedded in different algebras, one gets new perspectives on their inner mechanisms thanks to proof theory. As the type of an observation is the dual of the type of the input, the shape of its derivation is known, and a precise study of their proofs will lead to syntactical characterization (W.P. 3.1). Such a 'reverse engineering' will enlighten classical questions on TRS (W.P. 2.3) by drawing bridges between termination for observations (nilpotency) and fixed point for logic programs (boundedness).

…and aims at an algebraic definition of complexity… The governing principle of CAPTA is a mathematical perspective on complexity. It will benefit from a constant exchange with the COLA project (W.P. 1.1) and recollects the results of most of the previous work packages (W.P. 3.2). The ultimate goal is to prove that all structures isomorphic to structures capturing complexity classes have the same expressiveness, and that a strict inclusion in algebras corresponds to a strict inclusion in the corresponding classes. Even if weak monotonicity seems more reasonable, CAPTA will study carefully the known separations between classes to reproduce such a result.

…thanks to an inter/multidisciplinary approach and multiple collaborations Those tasks will require collaborations with all the COLA's members (W.P. 1.1), my previous co-authors (W.P. 1.2) and the expertise of the Algorithms and Programming Languages group in automata (W.P. 1.3, W.P. 2.1, W.P. 2.2).

W.P. 1.1	CATCH UP WITH COLA'S PREVIOUS WORK	Months 1–6
	Dependencies: None Risk Handling: Risk of getting lost, extend the duration of the W.P. and concentrate on specific articles	Risk: Low-to-medium
W.P. 1.2	CAPTURING P-TIME COMPLEXITY	Months 1–6
	Dependencies: None Risk Handling: Transform into a negative result	Risk: Low
W.P. 1.3	EMBEDDING CLASSICAL VARIATIONS OF AUTOMATA	Months 1–12
	Dependencies: None Risk Handling: Restrict to 1-head 1-way automata	Risk: Low-to-Medium
W.P. 2.1	FUNCTIONAL SETTING & log-SPACE-COMPOSITION	Months 6–18
	Dependencies: W.P. 1.3, concerning transducers Risk Handling: Get Inspirations from W.P. 3.1	Risk: Medium-to-High
W.P. 2.2	PARALLEL EXECUTION OF OBSERVATIONS	Months 12–18
	Dependencies: W.P. 1.2, W.P. 1.3 Risk Handling: Interpret syntactical characterization as von Neumann algebra	Risk: Medium-to-High
W.P. 2.3	QUOTIENT ON LOGIC PROGRAM, NILPOTENCY AS BOUNDEDNESS	Months 12–24
	Dependencies: W.P. 1.3 Risk Handling: Step back to the GoI approach (execution formula)	Risk: Medium

W.P. 3.1	SYNTACTICAL EQUIVALENT TO ALGEBRAIC CHARACTERIZATION	Months 6–18
	Dependencies: W.P. 1.2, W.P. 1.3, W.P. 2.2	Risk: High
	Risk Handling: Get inspiration from classical ICC	
W.P. 3.2	SEPARATION FROM AN ALGEBRAIC POINT OF VIEW	Months 12–24
	Dependencies: W.P. 1.1, W.P. 1.2 (partially W.P. 2.1 and W.P. 2.2)	Risk: High
	Risk Handling: Get inspiration from the COLA's project	



Work packages centred on ○ Complexity, ● algebraic perspective, ○ Proof Theory and ● automata.

4.2 Appropriateness of the management structure and procedures, including quality management and risk management

Project organisation and management structure UCPH is a first-class university, with an administration devoted to help their researchers, securing employment contract in due time and for the whole period. They will accompany the action in all its dimensions and are used to handle Marie-Curie fellows (the Faculty of Science welcomed 81 of them under FP7). They have lodging facilities (managed by UCPH's Housing foundation) and a solid experience in welcoming foreign researchers (with the International staff mobility, a part of UCPH's Human Resources Department). The staff dedicated to handling finances of the project (both at the Faculty and department level) will regularly make sure that the project is financially on track by providing monthly financial reports and revise if needed the initial plan.

The scientist in charge is gathering a team thanks to the COLA project, organizing seminars as well as sessions of interdisciplinary research on a weekly basis. I will have formalized weekly meeting with him to work on the project, plan forward and handle risks. The research support (2.5 persons) will help me reporting and running research projects.

Risks that might endanger reaching project objectives Every work package mention a risk level and a risk management, but the global contingency plan —that will be adapted in collaboration with the scientist in charge— depends on the level of the work packages.

Work packages 1.x are *safe*, for they develop existing techniques, are preparatory works and provide multiple attack angles. More specifically, W.P. 1.1 would focus on the most pertinent previous advanced and step back to general methodologies. If W.P. 1.2 were to fail, a precise investigation of the deadlocks has to be pursued. Regarding W.P. 1.3, the opposite approach would be developed: rather than looking for extensions, I would focus on subcases (1-way, 1-head) of the model of automata previously[40] embedded.

[40] Aubert et al., 'Logic Programming and Logarithmic Space'.

Work package 2.x are *promising*, they will be pursued independently and all offers alternative approaches. In the worst case, W.P. 2.1 would lead to a reformulation of the classical composition of log-space, embedding the algorithmic approach in algebra. Another angle to tackle W.P. 2.2 would be to interpret proof circuits[41], a Linear Logic-inspired model of parallel computation, in a von Neumann algebra. If the W.P. 2.3 does not provide the expected results, the extension to logic programming would be dropped, in order to focus on their proof-theoretical equivalent, nilpotency, and to track where the correspondence ceased to work.

Work packages 3.x are *speculative* and really ambitious, and as such more risky. The problems they raise are central, but if the proposed techniques were to fail, the contingency plan is to get inspiration from COLA's results: the questions are close to the interest of J. Frey (regarding complexity through topoi) and the scientist in charge (ICC characterizations from TRS constraints) so that they still can be addressed with their techniques.

4.3 Appropriateness of the institutional environment (infrastructure)

I will be embedded in a local environment that provides immediate access to experts in the area of complexity, lambda calculus, and programming language theory, e.g. with Professors F. Henglein, A. Filinski and T. Æ. Mogensen. In addition, I will interact with local post docs T. Heindel and J. Frey working on aspects of complexity theory in collaboration with the scientist in charge.

While the gist of the project is mostly theoretical, I will have an office with desk, whiteboard and to state-of-the art IT facilities, and access laboratories, should the need arise. Both the local Faculty of Science and the Department of Computer Science maintain permanent support units for IT and work environment, and the research support for the reports and the running research will assist me. Moreover, the communication and public engagement will be eased by a dedicated permanent full-time staff member of DIKU and the communications unit of the Faculty of Science.

The Department of Computer Science and the nearby ancillary staff at the Faculty of Science will offer local administrative support, facilitated by the UCPH International Staff Mobility Unit. Both the department (1 Turing award winner as alumnus, 32 permanent scientific staff members) and the University (8 Nobel prize winners as alumni, 5000 permanent scientific staff members) have a long-standing record of academic excellence and sheperding of young scientists.

4.4 Competences, experience and complementarity of the participating organisations and institutional commitment

The Department of Computer Science offers a premium environment to interact with scientific staff members, it is the only place where the intersection of ICC, algebra and TRS is taken as a central point of investigation. I will benefit from their expertise as well as their will to draw new and ambitious lines between disciplines and perspectives in collaboration with the COLA project (that ends in winter 2016).

On the other hand, they are missing an expert in Linear Logic and more specifically Geometry of Interaction: while it is one of the most advanced approach connecting algebra and ICC, the members of the Department of Computer Science barely have notions regarding that point. The scientist in charge, his department and the university already testified of a strong interest in my application and will accompany it in all its dimensions. My presence will renew the perspective of the COLA project and allow to reach a critical mass for fruitful interactions.

STOP PAGE COUNT – MAX 10 PAGES

[41] Aubert, 'Sublogarithmic uniform Boolean proof nets'; Aubert, 'Linear Logic and Sub-polynomial Classes of Complexity'.

5 CV of the Experienced Researcher

2 rue de l'Averse
94 000 Créteil, France
(+33/0)6.16.92.81.58
(+33/0)1.48.99.31.39.
clement.aubert@math.cnrs.fr
aubert.perso.math.cnrs.fr
Born on 14/09/1984 in France

Clément Aubert

Resume

Research Topics

Proof Theory, Implicit Computational Complexity, Proof–Program Correspondence, Automata, Linear Logic, Geometry of Interaction, Term-Rewriting Systems

Current Position

2013-2014 **Post Doc.**, *Logique de la Programmation (L.D.P.) team, Institut de Mathématiques de Marseille (I.2.M.) — UMR 7373 CNRS*, funded by the ANR Realizability for classical logic, concurrency, references and rewriting (Recré).

Education and Training

2010–2013 **Ph.D. in Computer Science, with the highest honours ("mention Très honorable")**, "Linear Logic and sub-polynomial classes of complexity", *École Doctorale Galilée (146) — Université Paris 13 — Laboratoire d'Informatique de Paris Nord (LIPN), UMR 7030.*
Defended on 26th November 2013, composition of the panel:

Mr. Patrick Baillot	C.N.R.S., E.N.S. Lyon	(Examiner)
Mr. Arnaud Durand	Université Denis Diderot - Paris 7	(Chair)
Mr. Ugo Dal Lago	I.N.R.I.A., Università degli Studi di Bologna	(Examiner)
Ms. Claudia Faggian	C.N.R.S., Université Paris Diderot - Paris 7	
Mr. Stefano Guerrini	Institut Galilée - Université Paris 13	(Supervisor)
Mr. Jean-Yves Marion	Lo.R.I.A., Université de Lorraine	
Mr. Paul-André Melliès	C.N.R.S., Université Paris Diderot - Paris 7	
Mr. Virgile Mogbil	Institut Galilée - Université Paris 13	(Co-supervisor)

2009–2010 **Research Master's degree in Mathematics, *cum laude***, *Université Paris 7 — Denis Diderot.*
"Mathematical Logic and Foundations of Computer Science" (LMFI), specializations in Proof Theory (P.-L. Curien) and Lambda-Calcul (T. Joly)

2007–2010 **Licence then first year of Research Master's in Philosophy**, *Université Paris 1.*
"Logic, Philosophy, History and Sociology of Sciences" (LOPHISS), specialization in Logic

2006–2007 **Studies in History (Master's) and Mathematics (first year)**, *Université Paris 7 & 8, C.N.E.D.*

2005–2006 **Licence in History and Bachelor's degree in Philosophy**, *Université de Reims.*

2003–2005 **Khâgne & Hypokhâgne (*Preparatory classes to the* grandes écoles, focused on Humanities)**, *Lycée Jean-Jaurès — Reims.*
Specialization in History, Geography and Philosophy

2000–2003 ***Baccalauréat* in sciences, distinction in German ("*mention européenne Allemand*")**, *Lycée Jean-Jaurès — Reims.*

Langages

French Mother tongue, skills in typography
English Perfectly read and understood, fluent speaker, TOIEC's score: 975
German Goethe Institut's *Zertifikat Deutsch* in 2002, specialization and distinction in High school
Russian Basic concepts

Publications

There is no such thing as a "first author" in my community, the authors appear in alphabetical order. All the joint works were led with Ph.D. students or recent Ph.D. laureate, on a basis of mutual enrichment and participation.

2014 *Logic Programming and Logarithmic Space*, **C. Aubert**, M. Bagnol, P. Pistone, Th. Seiller, 18 pages, accepted to *APLAS 2014*, `arxiv.org/abs/1406.2110`

This work strengthens and lightens the connections between Logic Programming and Proof Theory and simplifies the embedding of multi-head automata in our algebraic construction. It was accepted at the Asian Symposium on Programming Languages and Systems (APLAS), which is one of the main computer science conferences in Asia.

2014 *Unification and Logarithmic Space*, **C. Aubert**, M. Bagnol. In: RTA-TLCA. Ed. by G. Dowek. Vol. 8650. LNCS. pp. 77–92. `doi:10.1007/978-3-319-08918-8_6`

This article sets the first layers of a new correspondence between Logic Programming and Proof Theory, through a result on complexity relying on automata. This work was accepted at the joint conference on Rewriting Techniques and Applications (RTA) and Typed Lambda Calculi and Applications (TLCA), which is the most important conference in Term-Rewriting System. Their proceedings were published in Lecture Notes in Computer Science, one of the largest series of computer science conference proceedings.

2013 *Linear Logic and Subpolynomial Classes of Complexity*, **C. Aubert**, Ph.D Thesis in Computer Science, 178 pages. Superisors: S. Guerrini, V. Mogbil (UMR CNRS 7030 — Paris 13)

This lengthy thesis (178 pages) was written in three years and mixes three distinct fields: the aim is to contribute to complexity theory, and the tools are mathematical (von Neumann algebra) and proof-theoretical (proof-nets, geometry of interaction). It recapitulates three papers (on Proof circuits, and the two in collaboration with T. Seiller), clarifies the presentation and still contains material for at last two articles: one currently polished on automata and complexity classes, the other on a new correspondence between alternating Turing machines and proof circuits.

2013 *Logarithmic Space and Permutations*, **C. Aubert**, T. Seiller, 26 pages, submitted to Inform. and Comput. — Special Issue on Implicit Computational Complexity `arxiv.org/abs/1301.3189`

This work defines an ingenious constrain on algebra that is proven to capture all the deterministic log-space programs, thanks to an innovative proof that involve automata. It was submitted to a special issue (whose guest editors are Simona Ronchi Della Rocca and Virgile Mogbil) that is still in the process of being reviewed.

2012 *Characterizing co-NL by a Group Action*, **C. Aubert**, T. Seiller, 32 pages, submitted to MSCS, `arxiv.org/abs/1209.3422`

This article is a long (32 pages) and demanding work that is still under consideration for publication in one of the highest-ranked journals for Theoretical Computer Science, Mathematical Structures in Computer Science. While T. Seiller already had facilities with von Neumann algebra, I add to catch up with him as well as to develop a suitable model of computation ("Pointer Machines") to obtain the completeness of the model we developed.

2011 *Sublogarithmic uniform boolean proof nets*, **C. Aubert**, In: DICE. Ed. by J.-Y. Marion. Vol. 75. EPTCS. pp. 15–27. `doi:10.4204/EPTCS.75.2`

This work clarifies and extends to the uniform and constant-depth cases a parallel correspondence between a graph-theoretical presentation of proofs – Proof-nets – and boolean circuits. It was published in the proceedings of the "Developments in Implicit Computational Complexity" workshop, the reference in Implicit Computational Complexity and a part of European Joint Conferences on Theory and Practice of Software (ETAPS).

2010 *Boolean Proof Nets*, **C. Aubert**, Master's Thesis in Mathematics, 29 pages. Supervisors: V. Mogbil, P. Jacobé de Naurois (UMR CNRS 7030 — Paris 13)

2009 *Cut-elimination in the Constant-Domain Logic*, **C. Aubert**, Bachelor's Thesis in Philosophy, 29 pages. Supervisor: J.-B. Joinet (UMR CNRS 7126 — Paris 7)

Reviews

I am reviewer for
- the Italian Conference on Theoretical Computer Science,
- the journal "Information & Computation"

Invited Talks (selection)

The abstracts as well as the slides are on line at http://aubert.perso.math.cnrs.fr/#exposes.

2014 Logic, Computer Science, and Discrete Mathematics (L.I.M.D.) seminar — Laboratoire de Mathématiques de l'Université de Savoie (L.A.M.A.), Université de Savoie

Logique de la Programmation (L.D.P.) seminar — Institut de Mathématiques de Marseille (I.2.M.), Aix-Marseille Université

Seminar of the *Methodes formelles* team — Laboratoire lorrain de recherche en informatique et ses applications (Loria), Université de Lorraine

2013 International Workshop *Logic and Computational Complexity* (L.C.C.) 2013 — Turin

Complexité, Logique et Informatique (C.L.I.) seminar — Équipe de Logique Mathématique, Université Paris 7

Young Researchers' seminar — Laboratoire d'Informatique de paris Nord (L.I.P.N.), Université Paris 13

2012 9th project meeting of the ANR Implicit Computational Complexity, Concurrency and Extraction (Complice) — L.I.P.N., Université Paris 13

L.D.P. seminar — Institut de Mathématiques de Luminy (I.M.L.), Aix-Marseille Université

Foundations of Component-based Ubiquitous Systems (FoCUS) Meeting — Università Di Bologna

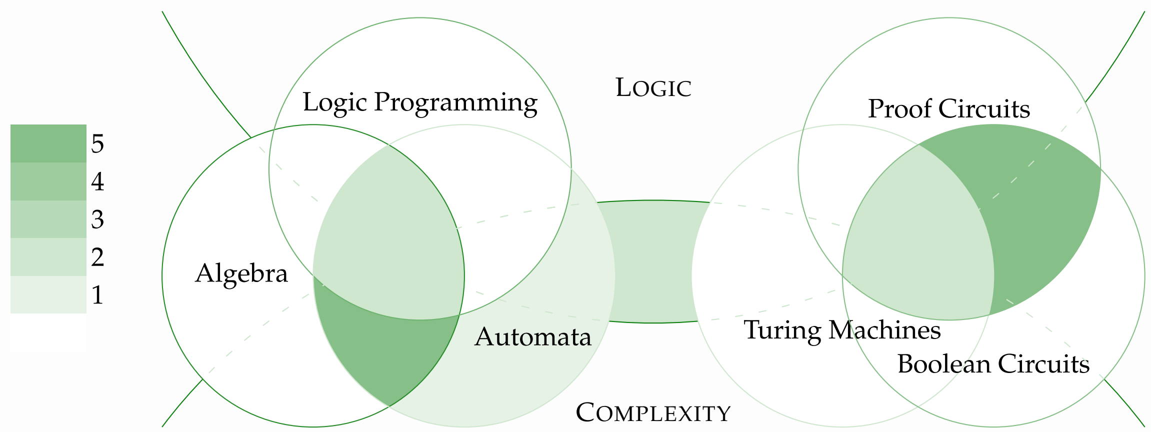
Logique, Calcul et Raisonnement (L.C.R.) seminar — L.I.P.N., Université Paris 13

Logic and Interactions 2012 — Centre International de Rencontres Mathématiques (C.I.R.M.)

2011 International Workshop Second Workshop on Developments in Implicit Computational Complexity (Dice 2011) — Saarbrücken, European Joint Conferences on Theory and Practice of Software (ETAPS) 2011

Multidisciplinary research group *Vérité et preuves* — Université Paris 1

16th meeting of the Logique, Algèbre et Calcul (L.A.C.) group — Preuves, Programmes, Systèmes (P.P.S.), Université Paris 7



Number of talks by subjects

Involvement in Projects

Productive member
- ANR Realizability for classical logic, concurrency, references and rewriting (Recré),
- ANR Logic and Geometry of Interaction (Logoi),
- ANR Implicit Computational Complexity, Concurrency and Extraction (Complice),
- Ph.D. Students working group *Verité et preuves*.
- Groupe de Recherche Informatique Mathématique (GDR IM)

Attending member
- ANR Programming reversible and dependable systems (Rever),
- ANR COmputing with QUAntitative Semantics (Coquas),
- ANR Parallel and Distributed Analysis (Panda),
- Curry-Howard: Logic and Computation (Chocola) meetings

Institutional Responsibilities

Responsible for a course: writing and assessing exams and exercises, coordination with other professors, leading role in the end-term meeting with the pedagogical team

Invited by the Evaluation Agency for Research and Higher education (AERES) to the panel of representative Ph.D.'s students during the evaluation of the Laboratoire d'Informatique de Paris Nord (LIPN), Paris 13.

Guest invited at a mid-term meeting with the Agence Nationale de la Recherche (ANR) regarding the Logic and Geometry of Interaction (Logoi) project

Teaching Experiences

During my Ph.D. thesis, I was in charge of some Lectures, exercise classes and lab sessions at the University Institute of Technology (I.U.T.) *of Villetaneuse, in the* Réseaux et Télécommunications *department.*

59 h. **Learning Differently**: Courses to help the students to grasp with different methods the content of their program in Mathematics. Also management of group projects and software engineering.

54 h. **Algorithms and Programming**: Introduction of a new course on algorithms and programming in C. The schedule alternated exercise classes explaining key concepts and lab sessions.

26 h. **Databases**: Theoretical models (UML, normalization, relational algebra), design practice and administration of databases: SQL queries, constraints, views, functions, programming C / SQL.

52 h. **System administration & network**: Introduction to scripting, databases, system administration, network configuration and monitoring. I actively helped the responsible to correct her lessons and provided numerous examples and ideas for subsequent evaluations.

During my Bachelor's degree in Philosophy (2008 – 2009), I was also in charge of writing exercises in Modal Logic with G. Sandu and J. Dubucs

Summer & Winter Schools

2014-04-07 ↳18 **Sémantique des preuves et des programmes et formalisation des mathématiques** — Luminy, with lectures of A. Miquel, T. Coquand, P.-L. Curien, . . .

2012-06-05 ↳08 **Réalisabilité à Chambéry #5** on Realizability — Bourget du Lac, with lectures of A. Miquel, M. Hofmann, J.-L. Krivine, H. Herbelin, . . .

2011-11-07 ↳11 **Workshop on Linear Logic** on Geometry of Interaction, Traced Monoidal Categories and Implicit Complexity — Kyoto, with lectures of J.-Y. Girard, S. Guerrini, U. Dal Lago, . . .

2011-06-16 ↳07-01 **10th Annual Oregon Programming Languages Summer School** on Types, Semantics and Verification — Eugene, with lectures of P.-L. Curien, H. Herbelin, X. Leroy, P.-A. Melliès, B. Pierce, D. Scott, . . .

2011-03-17 ↳19 **École d'été de l'ANR Logoi** on Geometry of Interaction, Operator Algebra — Carry-le-Rouet, with lectures of P.-L. Curien and J.-Y. Girard.

International visits & exchanges

Mar. 2011 (1 week) — Exchange with T. Seiller to work on Operator algebra and Complexity — Institut Mathématiques de Luminy, Université d'Aix-Mareille, funded by the GDR-IM

Nov. 2012 (2 weeks) — Visit to U. Dal Lago to work on Quantum Calculus, Geometry of Interaction and Implicit Complexity — FoCUS, Bologna, funded by the ANR Pics

Aug. 2014 (1 week) — Visit to J. G. Simonsen to work on Implicit Computational Complexity and Algebraic characterizations of complexity classes — Datalogisk Institut, Copenhague, funded by the COLA Project

Computer Skills

O.S. Linux, Mac OS, Windows

Prog. `C`, `SQL`, shell scripts, notions in `COQ` and `Prolog`

LaTeX Daily use, Ti*k*Z, Bussproofs, ...

Web dvpment `HTML5`, `CSS3`, W3C's specifications, WAI, `PHP`, `MySQL`

Volunteer Work

micr0lab — Founding member, member of the board, webmaster, and active participant in this ten-member non-benefit association that has existed for the past four years. I also had to develop leadership skills as well as practical organizational talent during the two editions of our festival (with 40 artists and an audience of 200 persons).

La goutte d'Ordi — Volunteer during a two years period, three to six hours per week: teaching classes and lab sessions for newly arrived immigrants, providing material to bridge the digital divide.

6 Capacities of the Participating Organisations

University of Copenhagen	
General Description	The scientist in charge internationally recognized expert on the mathematical treatment of computation and programming languages and has published a substantial number of papers in highly respected scholarly journals in theoretical computer science and is part of the Human-Centered Computing Group and Algorithms and Programming Languages group at the department; both groups have an extended international track record, with members having published more than two hundred papers in the last five years at top international venues such as ACM POPL and ACM STOC, and counting both a Turing award winner and ACM Programming Languages Achievement award winner among their alumni. The Department of Computer Science at the University of Copenhagen (DIKU) was founded in 1970 by later Turing Award Winner Peter Naur. The department hosts undergraduate and graduate studies in the fields of Algorithms and Programming Languages, Image Processing and Machine Learning, and Human-Centered Computing, with local interdisciplinary collaboration with the departments of Physics, Mathematics, Economics, and the Department of Media, Cognition and Communication, as well as a number of international research collaborations. The department hosts 32 permanent scientific staff members, and approximately 25 post docs, and 35 Ph.D. students, and has a sustained record of academic excellence evident both by international research awards (e.g., the Turing award and the ACM Programming Languages Achievement Award) and the consistently high international profile of its scientific alumni (e.g. the vice chancellor of the Danish IT University and the Director of the Fraunhofer Institute for Software and Systems Engineering in Dortmund). Founded in 1479, The University of Copenhagen is one of the oldest institutions of higher learning in Northern Europe. The University has more than 40000 students and 5000 academic staff; the University is consistently ranked among the top 100 universities in the world, and has a long-standing record of outstanding academic achievement, including 8 Nobel prize laureates.
Role and Commitment of key persons (supervisor)	The scientist in charge will collaborate scientifically with me, having formalized hourly weekly meetings. I will interact with local group members, including post docs J. Frey and T. Heindel, and have regular contact with Ph.D. students. The local research groups counts several members available for ad hoc consultancy, including experts in programming languages, lambda calculus, and computability theory, notably Pr. Henglein, Filinski and Mogensen.
Key Research Facilities, Infrastructure and Equipment	I will be given an office including desk, furniture, whiteboard and internet access. State-of-the-art computing facilities are maintained both by the Faculty of Science IT services and the Department of Computer Science.
Independent research premises?	Yes
Previous Involvement in Research and Training Programmes	The scientist in charge has supervised more than 35 master's theses, 4 Ph.D. students, a post doc, will host a Marie Curie post doc from the fall of 2014, and has participated as a faculty member in a number of formalized training networks for Ph.D. students and post docs, notably the Foundations for Innovative Research-based Software Technologies (FIRST) graduate training network involving the University of Copenhagen, IT University of Denmark, and the Technical University of Denmark.
Current involvement in Research and Training Programmes	The scientist in charge and I will be part of the APL and HCC sections of the department, both of which participate in several formal research training networks for doctoral and postdoctoral scholar, including roles as coordinating institution in ”Algorithms and Data Structures for Trees” (partnering with Princeton University's local coordinator, Turing award winner Robert Tarjan), and hosting role for the International School on Imaging and Machine Learning Research (IMLRS).
Relevant Publications and/or research/innovation products	*The name in bold are currently working at DIKU.* J. Ketema, **J.G. Simonsen**, ”Least Upper Bounds on the Size of Confluence and Church-Rosser Diagrams in Term Rewriting and Lambda-Calculus”. ACM Transactions on Computational Logic 14(4), paper 7, 2013. A.M. Ben-Amram, N.H. Christensen, **J.G. Simonsen**, ”Models of Computation with no Linear Speedup”. Chicago Journal of Theoretical Computer Science. Vol. 2012, article 7, 2012. **N.D. Jones**, **J.G. Simonsen**, ”Programs = Data = First-Class Citizens in a Computational World”. Philosophical Transactions of the Royal Society A 370, pp. 3305-3318, 2012 **J.G. Simonsen**. On the modularity of confluence in infinitary term rewriting. In Proceedings of the 15th International Conference on Rewriting Techniques and Applications (RTA '04), volume 3091 of Lecture Notes in Computer Science, pp. 185–199. Springer-Verlag, 2004. **R. Glück**. “Simulation of Two-Way Pushdown Automata Revisited”. In: Festschrift for D. Schmidt, volume 129 of Electronic Proceedings in Theoretical Computer Science, pp. 250–258, 2013.

7 Ethical Aspects

No ethical aspect to be considered.

8 Letters of Commitment of Partner Organisations

No partner organisation are involved in the proposal.

ENDPAGE

MARIE SKŁODOWSKA–CURIE ACTIONS

Individual Fellowships (IF)
Call: H2020-MSCA-IF-2014

PART B

"CAPTA"

"Complexity through Automata, Proof Theory and Algebra"

This proposal is to be evaluated as:

[**Standard EF**]