2024-04-09

Application

Draft	Curzi, Gianluca

Information about applicant

Project leader: Gianluca Curzi

Doctoral degree: 2020-06-12

Birthdate: 19910129

Academic title: Doctor

Gender: Male

Employer: Göteborgs universitet

Administrating organisation: Göteborgs universitet

Project site: Filosofi, lingvistik och vetenskapsteori, inst för

Information about application

Call name: Research Grants Open call 2024 (Natural and Engineering Sciences)

Type of grant: Starting Grant

Focus: Natural and engineering sciences

Call for proposals subject area: NE

Project title: Proofs with Cycles in Computation

Project start: 2025-01-01

Project end: 2028-12-31

Review panel applied for: NT-Q, NT-S, NT-Q

Classification code: 10201. Computer Sciences, 10103. Algebra and Logic

Keywords: Cyclic proofs, Recursion theory, Computational complexity, Mathematical logic, Proof theory

Funds applied for

Year:	2025	2026	2027	2028	Total amount applied
Amount:	1,100,000	1,100,000	1,100,000	1,100,000	4,400,000

Descriptive information

Project title (Swedish)

Bevis med cykler för beräkning

Project title (English)

Proofs with Cycles in Computation

Abstract and popular science description

Please note that the abstract may be used when distributing the application to the reviewers that will make the scientific assessment of it.

Abstract (English)

This research project is centred around the computational aspects of cyclic proofs, a generalisation of the notion of mathematical proof that offers a natural and systematic approach for modelling recursion. Its main objective is to develop a broad-spectrum of methodologies and techniques for investigating recursion mechanisms and their computational complexity within the setting of cyclic proof theory. The project is motivated by (and based on) a series of preliminary results achieved during the applicant's previous research, which will serve as proof-of-concept, and will be developed by benefiting from a series of national and international collaborations with the foremost experts in the area.

The programme covers a four-year time span and is organised in two work packages, each one divided into tasks. The proposed research has a strong interdisciplinary nature, encompassing a number of theoretical and applied topics within the areas of mathematical logic and computer science, such as Proof Theory, Recursion Theory, Type Theory and Computational Complexity. In the last part of the programme, the research will pivot towards implementation-oriented goals related to Formal Verification and Theorem Proving, aiming at developing quantitative tools for static certification of runtime efficiency of programs. Given its innovative and ambitious nature, the proposed research will impact not only theoretical areas but also software development and verification in the long term.

Popular science description (Swedish)

Detta forskningsprojekt kommer att fördjupa den matematiska förståelsen av *rekursion* - förmågan hos ett program eller en process att instansiera sig själv från sin egen kod eller beskrivning. Tillgång till denna förmåga hos ett programmeringsspråk ger möjlighet att lösa beräkningsproblem genom att bryta ner uppgiften till enklare men liknande delproblem och är en viktig källa till språkets beräkningskraft. Att bemästra sådan beräkningsförmåga är nyckeln till att lösa problem inom komplexitetsteori såsom det berömda *P vs. NP* problemet: är varje beräkningsuppgift med effektivt verifierbar lösning också effektivt lösbar? Att närma sig svaret på denna fråga är inte bara av teoretiskt stor vikt utan skulle direkt påverka många aspekter av det moderna digitala samhället. Faktum är att säkerheten hos exempelvis dagens banksystem är direkt avhängig den algoritmiska svårigheten att knäcka kommunikationsprotokoll med asymmetrisk kryptering.

Trots sin centrala roll har rekursion betraktats som en "magisk låda", en primitiv odelbar princip som inte går att analysera djupare, så till den grad att standardformuleringen av formell bevisföring inte kan redogöra för resonemang med cirkularitet såsom rekursion. Dessa begränsningar kan dock kringås tack vare en modern utvekling och generalisering av matematisk bevisföring kallad *cyklisk bevisföring*, just ämnad för att tillåta cirkulära resonemang.

Med stöd från sökandes preliminära forkskningsresultat ämnar det föreslagna projektet att utarbeta ett enhetligt ramverk för utforskningen av rekursion med cyklisk bevisföring som verktyg. Projektet kommer utveckla ett stort antal nya metoder och fördjupa den matematiska förståelsen för beräkningsteori och komplexitetsteori. Det föreslagna forskningsprogrammet spänner över många områden inom matematik och datavetenskap och ämnar yterlligare stärka sin interdiciplinära natur med tillämpningar av cyklisk bevisföring för mjukvaruutveckling och formell verifikation.

Planned use of research infrastructure

Specify national/international infrastructures funded by the Swedish Research Council, not local core facilities.

Planned use of research infrastructure

No

Other applications and grants

Describe if any of the items below are relevant to you.

- You are applying for or intend to apply for other grants from the Swedish Research Council.
- You are receiving an ongoing grant from the Swedish Research Council with a grant period that wholly or partly overlaps with the grant you are now applying for.
- Applications or grants relating to the same project idea from the Swedish Research Council or other funding bodies (from you or another researcher).

If you answer Yes, you should, in all cases, justify why you submit one or more applications and also describe the relationship between the different projects.

Are any of the items relevant to you?

No

Deductible time

If applicable, enter deductible time according to the instructions in the call text.

Deductible time

Cause	Months
Career age is equivalent to the time from your first doctoral degree until the last application day of the call. Your career age changes if you indicate deductible time due to a reason approved by the funder. For some calls there are restrictions in the allowed career age.	

Research description

Ethical aspects: Legal and formal requirements

State the specific legal and formal requirements that may be relevant for your application.

The research includes experiments on animals that requires ethical approval under the Animal Welfare Act (2018:1192)

No

The research includes studies on humans and/or biological material from humans and requires approval under the Act (2003:460) on ethical review of research relating to humans

No

The research includes the processing of personal data in accordance with the General Data Protection Regulation

No

If you answered Yes to one or more of the questions above, you should describe how you plan to obtain relevant approvals and permits before the project starts

Regardless if you answered Yes or No to the questions above, state whether other approvals or permits, in addition to the above, are relevant to your application.

If no approvals or permits affect you application please, state this.

Description of approvals and permits (English)

No approval or permit is required for this application.

Ethical considerations

Specify any ethical issues raised by the project (or equivalent) and describe how they will be addressed, according to the guidelines in the call text. If no ethical issues are raised, justify this.

Description of ethical considerations (English)

There are no ethical issues or implications in relation to this project, and I confirm that no ethical approval needs to be sought. The research does not involve children or animals, nor does it present an indirect risk to humans or animals. There are no potential ethical issues due to the social or environmental implications of the study and the research does not use sensitive personal data.

Sex and gender perspectives

Please state whether sex and gender perspectives are applicable in your planned research, and justify your decision, according to the instructions in the call text.

Sex and gender perspectives in the proposed research

No

Motivate your answer (English)

Sex and gender perspectives are not applicable in this project. The research does not specifically target or exclude any gender identities or sexual orientations, nor does it present an indirect risk to individuals based on their sex or gender. There are no potential issues due to the sex or gender-related implications of the study, and the research does not use sensitive personal data concerning individuals' sex or gender identity.

Research plan (English)

See following page for attachment

Proofs with Cycles and Computation

Research plan

Gianluca Curzi

1 Purpose and aims

The proposed project aims at developing general theoretical frameworks and novel methodologies for studying cyclic proofs from a computational perspective. Its main objective is to break down the present barriers in the recursion-theoretic reading of cyclic proofs, where proof techniques are either missing or bespoke and tailored to specific settings. The programme has a strong interdisciplinarity nature, ranging from mathematical logic to applied computer science, and builds on a series of preliminary and previous investigations conducted by the applicant in the context of fixed point logics and computational complexity.

Cyclic proofs generalise the traditional notion of (logical or mathematical) proof. By default, a proof of a theorem T is an inductively defined object, typically represented as a finite tree in which T is the root, axioms are leaves, and inference rules are inner nodes (Figure 1, left). Cyclic proofs are rather particular infinitary objects. Specifically, a cyclic proof is a non-wellfounded proof with a *regular* structure, that is, it only contains finitely many distinct subproofs. Because of this feature, cyclic proofs can be presented as *finite* trees with backlinks (Figure 1, right).

A paramount role in the literature is given by *fixed point logics*, being particularly well-suited for a cyclic proof-theoretic treatment. They typically extend a given logic with operators μ (least fixed point) and ν (greatest fixed point) binding propositional variables occurring positively, e.g., admitting the formula $\mu X(\top \vee X)$ but ruling out formulas like $\mu X(\top \vee \neg X)$. Semantically, fixed point logics are modelled via the Knaster-Taski's theorem on monotone functions over complete lattices. A notable example of fixed point logic is the modal μ-calculus. Other relevant fixed point logics are those based on intuitionistic logic (μLJ) [13] and linear logic (μMALL and μLL) [8], which admit an interpretation of fixed point formulas in terms of (co)inductive types and (co)recursion mechanisms.

The computational reading of proofs is driven by the celebrated *Curry-Howard isomorphisms* paradigm, according to which a proof of a formula T can be seen as a functional program having T as type specification, and program execution corresponds to proof rewriting. This computational approach to proofs can be extended to cyclic proofs as well, which have a natural counterpart in special programs able to introduce complex and intertwined “loop” structures, where functions can call themselves from within their own code. Furthermore, in the setting of cyclic proofs, program termination can be duly expressed by appropriate global proof-theoretic conditions called *progressivity criteria*, which can be effectively certified by reduction to universality checking for Büchi automata over infinite words.

By virtue of these peculiar features, cyclic proofs have established themselves as an ideal framework to investigate *recursion* in a natural and systematic way [30, 20, 18, 16]. What is more, thanks to their looping capacity, cyclic proof systems can subsume a number of sophisticated forms of recursion mechanisms that can hardly be manipulated by ordinary inductive systems, and so they can be source of great algorithmic expressivity.

Adapting these considerations to the realm of resource-aware computation, in a recent series of papers [15, 17] the applicant presented novel methods to tame the computational power of cyclic proofs and recast the complexity-theoretic notion of *computational efficiency*. To the

Figure 1: The structure of inductive and cyclic proofs compared.

best of our knowledge, these are the first characterisations of complexity classes using the technology of cyclic proofs. Furthermore, such results are formulated in the style of Implicit Computational Complexity, i.e., in a purely abstract, machine-free setting and without imposing those time/space bounds that typically define complexity classes. The applicant's work substantially launched a new topic in computational complexity: *Cyclic Implicit Complexity* (CIC).

Despite the great advances in cyclic and non-wellfounded proof theory, the analysis of the computational features of cyclic proofs is surprisingly underdeveloped, only partly explained by its recent inception. Indeed, the infinitary nature of cyclic proofs lies at the very heart of the challenging aspects of the topic. On the one hand, more complex reasoning principles (e.g., infinitary pigeonhole principle and König's lemma) and proof methods (relying, e.g., on coinduction and bisimulation) replace the usual induction-based techniques. On the other hand, the inherent difficulties in the recursion-theoretic reading of cyclic proof systems have prevented the development of general and non-bespoke methods for, e.g., identifying the computational expressivity of a cyclic system, extracting recursion mechanisms from cyclic proofs, or proving infinitary cut-elimination results.

The considerations above outlined have led the applicant to the following desideratum:

Purpose of the project. *Develop a unifying and comprehensive framework for analysing the computational and complexity-theoretic properties of cyclic proofs.*

The research programme is meant to investigate the computational aspects of cyclic proofs in a wide range of settings, spanning from very complex forms of ordinal and higher-order recursion down to quickly converging recursion schemes for complexity. The goals of the project are manifold, and can be divided into two main areas of interest elaborated in the next sections:

(G1) Computational analysis of fixed point logics.

(G2) Theory and applications of Cyclic Implicit Complexity.

I am a world expert on the interface between cyclic proofs and computation, and a pioneer of the cyclic proof-theoretic approaches to complexity, as witness by my aforementioned founding works on CIC. My scientific profile is then the best-suited for developing this research project.

2 State-of-the-art

Finitisation methods. A fundamental question in cyclic proof-theory is whether cyclic and inductive proofs are "equally powerful", e.g. in terms of provability, or from the viewpoint of their computational strength. This question dates back to the so-called *Brotherston-Simpson conjecture* within the context of Martin-Löf's system of inductive definitions [12]. Since then, it has been prompting a variety of methods for turning cyclic proofs into inductive ones, which we will call *finitisation methods*. A commonly adopted technique is to formalise a soundness argument for a cyclic arithmetic or the totality argument for a cyclic proof (or type) system in subsystems of second-order arithmetic, using reflection and conservativity results. This approach has been first proposed by Simpson for Peano arithmetic [40], later refined by Das [19]. It was then extended to arithmetic with (finitely iterated) inductive definitions ($\mathsf{ID}_{<\omega}$) by Das and Melgaard [21], and to the generalised inductive definitions ($\mu\mathsf{PA}$) by Das and myself [16]. The literature contains other more direct translation methods that do not step through metamathematical arguments formalised within arithmetical theories. An approach for embedding cyclic into inductive proofs is to "unravel" a cyclic proof until the inductive hypothesis can simply be "read off" the structure of the proof. The inductive proof resulting from this procedure thus naturally corresponds to an unfolding of the initial cyclic proof. This method was first employed in [41] for first-order logic with fixed points, and later by Afshari and Leigh in the setting of the modal μ-calculus [5, 4]. Finally, in a recent paper [15] I introduced a more direct translation that does not make use of unfoldings, applied to an implication-free fragment of Gödel's system T.

Infinitary cut-elimination techniques. Early attempts to study cut-elimination in an infinitary setting can be traced back to Mints' famous "continuous cut-elimination" [34]. Fortier and Santocanale seem to have been the first of the modern era to give a cut-elimination result for cyclic proofs, namely for the additive fragment of linear logic with least and greatest fixed points ($\mu\mathsf{ALL}$) [24]. This was generalised in later work to the logic including multiplicatives too, specifically for the non-wellfounded formulation of $\mu\mathsf{MALL}$ in [7], and with more expressive progressing conditions more recently in [6]. All these works prove that the infinitary cut-elimination strategy is "productive" (i.e., it returns a well-defined cut-free proof in the limit) and that the progressivity criterion is preserved in the limit. Deepening this Curry-Howard viewpoint, presentations of "proof nets" (a graphic description of linear logic proofs) have appeared [23], yielding a form of Natural Deduction for cyclic proofs. In a recent work, Saurin has shown how to extend cut-elimination for non-wellfounded proofs from $\mu\mathsf{MALL}$ to $\mu\mathsf{LL}$, the latter being the fully-fledged system of linear logic with fixed points [39]. The proof relies on compression mechanisms for turning a transfinite cut-elimination sequence into an ω-long one, essentially borrowing standards methods from infinitary rewriting. Cut-elimination techniques for the cyclic modal μ-calculus can be found in [4, 3]. Finally, in [1] I showed an infinitary cut-elimination result for a non-wellfounded version of *Parsimonious Linear Logics* (a fragment of linear logic), using notions from domain theory and alternative cut-elimination methods compared to [7, 6].

Complexity and cyclic proofs. Implicit Computational Complexity (see Section 1) is a branch of computational complexity that originates from the seminal paper by Bellantoni and Cook [9]. This work introduces an algebra of functions based on a weaker and bound-free version of recursion, called *safe recursion*, able to capture the class of polynomial time computable functions (**FP**). This characterisation of **FP** was later extended to finite types with the system of *higher-order safe linear recursion* (SLR) [26], and to Peano arithmetic with the theory of *feasible arithmetic* ($\mathcal{A}^1_2$) [10]. In joint work with Anupam Das [15] I bridged the gap between Implicit Computational Complexity and cyclic proof theory. In that paper I introduced cyclic proof systems based on safe recursion that capture **FP** and the class of elementary functions (**FELEMENTARY**). Furthermore, in a follow up work [17], I extended these results to characterise the class of functions computable by polynomial size circuits (**FP**/**poly**). Finally, in [2] I applied CIC to the setting of linear logic, presenting cyclic proof systems for **FP** and **FP**/**poly**.

Formal verification and automated theorem proving. Cyclic proof theory has gained significant attention in the field of program verification and automated theorem-proving. Various work on verifying programs and inductive properties, especially in the context of Separation Logic, has been based on cyclic reasoning techniques [38, 42], leading to the development of the Cyclist [11, 14], Songbird (see `https://songbird-prover.github.io`), and Inductor provers [27]. These techniques have also been employed in the Cypress program synthesis tool [28], in the CycleQ prover for equational reasoning about functional programs [29], and for automatically verifying termination of functional programs [33].

3 Significance and scientific novelty

The proposed research aims at developing a broad-spectrum of methodologies and techniques for the computational and complexity-theoretic analysis cyclic proofs, and to explore its applications. The project has a strong interdisciplinary nature, covering a variety of theoretical and applied topics within the areas of mathematical logic and computer science. It's short term impact will be primarily on theoretical fields such as **Proof Theory**, **Recursion Theory**, **Theories of Arithmetic**, **Type Theory**, and **Computational Complexity**. The project also includes implementation-oriented goals within **Formal Verification** and **Automated Theorem Proving** (see **T2.4** of **WP2** in Section 5.1). Indeed, one of the objectives of this project (and the major practical goal of CIC) is to scale down verification methods to the realm of complexity, developing quantitative tools for static certification of programs' *runtime efficiency*. It is worth mentioning that formal verification has immensely benefited from deductive approaches introduced by proof theory (see, e.g., [25]). Furthermore, recent work has shown the enormous potential of cyclic proof theory for the analysis of program correctness, providing e.g. applications to safe termination of imperative programs [38, 42] and developing a general automated theorem-prover software implementing cyclic proof systems [11] (see Section 2). Thus, we expect that our research will impact the software development and verification community in the long term.

4 Preliminary and previous results

The proposed programme is motivated by (and based on) a series of preliminary results achieved by the applicant, which will serve as **proof-of-concept** of this project.

In [16], I answered a *fundamental open question* about the computational strength of fixed point logics: what class of computable functions do these systems represent? I showed that cyclic and inductive fixed point logics (such as μLJ, μMALL, and μLL) all represent the same functions (on natural numbers), which are precisely those provably recursive in μPA. This result generalises the finitisation methods previously introduced in the literature, e.g., for system T [18, 20], Peano arihtmetic [40, 19], and (finitely iterated) inductive definition [21] (see Section 2). Such methods will contribute to **WP1** and, in particular, to **T1.1** (see Section 5.1).

In a series of joint works with Anupam Das [15, 17] I presented a new topic in computational complexity, called *Cyclic Implicit Complexity* (CIC). Specifically, I introduced cyclic proof systems implementing ideas from Bellantoni and Cook's safe recursion [9] and characterising some of the most relevant complexity classes, such as **FP**, **FELEMENTARY** and **FP/poly** (see Sections 1 and 2). This preliminary work also introduces novel and more direct methods for translating cyclic proofs to inductive ones. Overall, my foundational results on CIC give evidence to viability of **WP2** (see Section 5.1), and set the groundwork for its development.

5 Project description

This section describes the work plan of the project to achieve the goals **G1** and **G2** of Section 1. It also discusses technical details related to the project implementation and organisation.

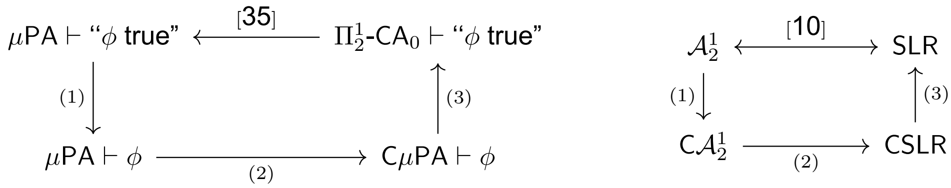

Figure 2: Diagrams relating the computational strength of various inductive and cyclic systems.

5.1 Theory and method

The research objectives of the project are organised into two *work packages* (**WP**) designed to address, respectively, goals **G1** and **G2** (see Section 1). Each work package is divided into four technical *tasks* (**T**), numbered accordingly to facilitate referencing.

WP1. Computational analysis of fixed point logics

This work package aims at developing general and abstract methods for studying the computational aspects of non-wellfounded, cyclic and inductive fixed point logics.

T1.1 *Computational strength of (cyclic) arithmetic with fixed point (**low-risk**).* I will show that Peano arithmetic with fixed points (μPA) and its cyclic formulation (CμPA) have the same computational expressivity. To this end, I will generalise previous results on arithmetical (finitely iterated) inductive definitions ($\mathrm{ID}_{<\omega}$) [21]. The proof methods are illustrated in Figure 2 (left), where the arrows compare the systems w.r.t. their classes of provably recursive functions. In particular, (1) relies on the reflection principle, and (2) requires a simulation property relating inductive and cyclic systems. The key result is (3), in which I will formalise metamathematically the soundness argument for CμPA within the theory Π^1_2-CA_0 (i.e., the subsystem of second-order arithmetic with Π^1_2-comprehension and set induction). For this step I will exploit the reverse mathematics of the Knaster-Tarski theorem developed in my previous work [16]. The well-known conservativity results for Π^1_2-CA_0 and μPA [35] will conclude the grand tour diagram.

T1.2 *Computational strength of (cyclic) proof systems with fixed point (**medium-risk**).* In concert with **T1.1**, I will extend my previous results from [16] by investigating the computational expressivity of two fragments of intuitionistic logic with fixed points (μLJ) in their cyclic setting: intuitionistic logic with *strictly positive fixed points* and its restriction to *implication-free fixed points*. Such subsystems are of great importance for the implementation perspectives of the project, as proof assistants (such as Coq or Agda) typically enforce strict positivity conditions on (co)inductive types for consistency reasons. I suspect that the strict positivity requirement does not affect the computational strength of μLJ. On the other hand, I conjecture that the lack of implications within fixed points reduces the computational power of μLJ to that of Gödel's celebrated system T. The major obstacle of this task is that the argument I developed in [16] cannot be scaled down to these fragments because it involves double-negation translations, which break the strict positivity (and implication-freeness) conditions. So, alternative routes are needed which rely on sophisticated proof-theoretic translations.

T1.3 *Finitisation methods (**medium-risk**).* I will first study direct embeddings from cyclic to inductive system T, adapting to our setting Dominik Wehr's results for cyclic Heyting arithmetic, which in turn rely on Sprenger and Dam's *unfolding method via induction orders* (see Section 2). The main hurdle of this deliverable is the presence of an explicit contraction rule in system T, which requires adapting the notion of induction order. I will then generalise this embedding to relevant fragments of μLJ, such as its restriction to implication-free fixed points (see **T1.2**).

T1.4 *Continuous cut-elimination techniques (**high-risk**).* I will develop a general infinitary cut-elimination method for the non-wellfounded (and cyclic) formulation of μLJ that relies on *reducibility candidates* à la Girard (see, e.g., [8]). Specifically, cut-elimination will amount to showing that any non-wellfounded proof Π of μLJ belongs to an appropriate reducibility candidate. To this end, I will proceed by an argument towards contradiction that resorts to assigning ordinals to fixed point formulas in Π, taking inspiration from soundness and totality arguments (see, e.g., [22, 20]). This task is a cornerstone of the project. If successful it would represent a major advance in the topic, introducing general and non-bespoke infinitary cut-elimination methods for cyclic and non-wellfounded proofs.

WP2. Theory and applications of Cyclic Implicit Complexity

This work package aims at developing basic and advanced theory of CIC, exploring its applications to Formal Verification and Theorem Proving, also covering some implementation aspects.

T2.1 CIC *within space complexity (**low-risk**).* I will extend my previous works on CIC [15, 17] to space complexity. First, I will characterise the classes **FL** (functions computable in logarithmic space) and **FL**/**poly** (functions computable by polynomial size branching programs). To this end, I will identify appropriate *linearity conditions* on the cyclic proof systems for **FP** and **FP**/**poly**, taking inspiration from function algebras for logarithmic space [36]. Secondly, I will design a cyclic proof system for **FPSPACE** by reformulating in the context of CIC the recursion-theoretic characterisation from [37].

T2.2 CIC *in higher types and feasible arithmetic (**medium-risk**).* I will design cyclic formulations of the type system SLR (called CSLR) and of the theory of *feasible arithmetic* $\mathcal{A}^1_2$ (called C$\mathcal{A}^1_2$), which are polynomial time versions of Gödel's T and PA respectively (see Section 2). The goal is to establish the results illustrated by the diagram in Figure 2 (right). In particular, (1) shows that cyclic proofs can simulate inductive proofs, and (3) builds on techniques I introduced [15], being CSLR a higher-order version of the cyclic proof systems presented in that paper. The challenging direction is (2), where I will formulate Gödel's functional interpretation for cyclic proofs. This result will pave the way to the analysis of the constructive content of "feasible" arithmetic and program extraction methods (e.g., proof mining) in a cyclic proof-theoretic setting. What is more, because of the expressivity power of cyclic proofs discussed in Section 1, I expect that CSLR will be able to model algorithms of practical interest that can hardly be found in SLR, e.g., sorting algorithms [31].

T2.3 CIC *in alternative paradigms of computation (**high-risk**).* This task is the second cornerstone of the project. I will apply CIC to randomised, parallel, and higher-order algorithms. The main goal is to use the principles of CIC to model the notion of efficiency within alternative paradigms of computation. First, I will investigate the probabilistic complexity class **BPP** (*bounded-error probabilistic polynomial time*), a subclass of **FP**/**poly**. A striking aspect of this class is that the error probability can in principle be reduced at will while incurring only a polynomial slowdown, so increasing the reliability of the answer without affecting efficiency. I will use as workbench the cyclic proof system for **FP**/**poly** I presented in [17]. On top of it, I will identify global proof-theoretic conditions that induce the appropriate error bounds for capturing **BPP**. If successful, this deliverable will produce the first fully "implicit" characterisation of this class, i.e., our systems will *not* rely on external bounds for modelling errors. This will solve an open problem in *implicit complexity* [31]. As a second deliverable, I will investigate CIC-like characterisations of complexity classes based on parallel computation, such as **NC** (*Nick's class*). To this end, I will design a cyclic proof system with generalised inference rules modelling proof-theoretically "divide-and-conquer" safe recursion mechanisms and forms of parallel composition, two main ingredients for capturing **NC** [32]. Last, I will characterise in the style of CIC the higher-order complexity class $\mathbf{BFF}_2$ (*type 2 basic feasible functionals*), a sort of higher-order analogue of the class **FP**. Higher-order complexity is frontier research: as such, the results produced will represent a major breakthrough.

T2.4 *Applications to formal verification and theorem proving (**medium-risk**).* In the last part of the project I will pivot the research towards applications of CIC, also considering some implementation tasks. I will develop a formal verification framework for bound-free static analysis of runtime efficiency via cyclic deductive reasoning. The latter will be defined on top of CSLR (see **T2.2**), which will be integrated with rules manipulating, e.g., temporal modalities, along with a set of rules that simulate program execution steps, where cycles in proofs will model recursive behaviour, as done in [42]. The resulting cyclic systems will be implemented within CYCLIST, an automated theorem prover realising a fully general setting for constructing cyclic proofs and for verifying global correctness criteria [11], basically using techniques from [42, 38].

5.2 Time plan and implementation

I discuss some aspects related to the work plan and its implementation.

Workplan. The workplan is illustrated in the Gantt chart in Figure 3. The two work packages are developed in parallel over a four-year time span, each one comprising four tasks coloured along the following risk code: **low-risk**, **medium-risk**, and **high-risk**. Interdependencies between tasks are displayed as well. Concerning **WP1**, the fragments of μLJ investigated in **T1.2** will be used as workbenches for **T1.3** and **T1.4**. As for **WP2**, I expect that the characterisation of **FPSPACE** in **T2.1**, which resorts to parallel composition operators, will offer insightful proof-theoretic tools for applying CIC to parallel computation (**T2.3**). Finally, the system CSLR studied in **T2.2** will provide the core framework for developing the formal verification aspects of **T2.4**.

Risk assessment and mitigation. The first 18 months of the project are devoted to low and medium risk tasks (**T1.1-3** and **T2.1-2**). In particular, towards the end of month 18 I will have developed the main foundational results and methodologies to conduct high-risk research, which will play a prominent role in the rest of the project (**T1.4** and **T2.3**). Furthermore, by month 36 I will have achieved some key milestones of the work plan and started driving forward the application-oriented part of the project (**T2.4**). Therefore, I consider months 18 and 36 as monitoring periods where major hurdles should be evident and the progress made can be duly assessed.

To mitigate risks and facilitate re-orientation of the project, I have organised the main tasks of the project to include lower risk deliverables and suggest alternative routes. As an example, in case of a failure of the finitisation result for fragments of μLJ in **T1.3**, I will focus on the embedding from cyclic to inductive proofs of T, which represent a low-risk and independent goal. Also, if difficulties in achieving **T2.2** arise, I will concentrate on point (3) of Figure 2 (right), which will adapt to a higher-order setting methods from [15], so that **T2.4** will not be compromised.

Key collaborations. The first two tasks of **WP1** will be developed in collaboration with members of the *Theory of Computation* group of the University of Birmingham (UK), the hosting institution of the applicant's first postdoc (2020-2023). This group is one of the largest in the world to focus on the logical and mathematical foundations of computing. In particular, research in cyclic and non-wellfounded proof theory within the *proof theory team* is obtaining a prominent status, thanks to an increasing number of members working in that area. I will initiate collaborations with Lukas Melgaard (**T1.1**), who agreed to undertake a short **visiting period** (1 month) at the University of Gothenburg, and Anupam Das (**T1.2**).

The proposed project will also benefit from a series of collaborations with the group *Preuves, programmes et systèmes* (*PPS*) from IRIF (Institut de Recherche en Informatique fondamentale), Université Paris Cité (France). This centre is renowned for leading research in the compu-

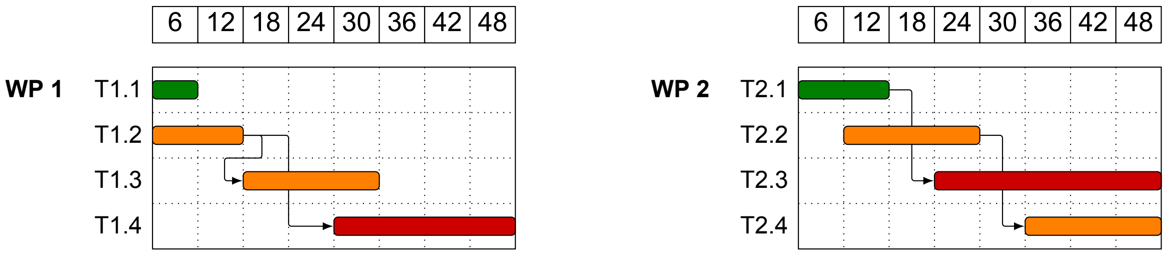

Figure 3: Gantt chart summarising the work plan of the project.

tational aspects of logic. In particular, cyclic and non-wellfounded proof theory is an established topic within the research group. I will deliver key tasks of the project by leveraging on the long-standing network of scientific interactions with the group I initiated during my PhD, when I was member of IRIF. I have outlined collaborations with Alexis Saurin and Hugo Férée to deliver the tasks **T1.4** and **T2.3**, respectively. In particular, for achieving **T1.4** I envisage undertaking a medium-term **visiting period** (3 months) with Alexis Saurin at the University of Paris Cité.

Finally, during the proposed programme, I will benefit from local collaborations at the University of Gothenburg. To deliver **T1.3**, I will initiate a joint work with both Graham Leigh and Bahareh Afshari. I am also intrigued by potential collaborations with Dominik Wehr, who is knowledgeable in proof theory and constructive type theory.

Research outcomes. The scientific outcomes of the proposed project will be published in the proceedings of peer-reviewed international top conferences, such as LICS, POPL, CSL and CPP, and high-impact scientific journals such as TCS, LMCS and MSCS. The University of Gothenburg will support my commitment to open access publishing during this research project, for which adequate funding is being requested.

5.3 Project organisation

The proposed project will be hosted at the Department of Philosophy, Linguistics and Theory of Science at the University of Gothenburg. I will be project leader and sole investigator, as no research partner is involved in this project. I will dedicate 75% of time to the proposed research project. In the first two years (2025 and 2026) my salary will be complemented by the research project *Taming Jörmungandr: The Logical Foundations of Circularity* (2021–2026), funded by the Wallenberg Academy Fellowship Prolongation grant (Project code: 251080003) and led by my current postdoctoral supervisor Graham Leigh. The remaining 25% of my time will be financially supported by other research funding and teaching (see **Justification of the Budget**).

6 Independent line of research

My research on CIC and the computational aspects of fixed point logics was initiated during my first postdoc at the University of Birmingham, under the supervision of Associate Professor Anupam Das. Also, cyclic proofs represent a core aspect of my present postdoc at the University of Gothenburg, in a research project led by Associate Professor Graham Leigh. The proposed project is meant to expand these research topics towards novel directions that are not been investigated and not being developed by my past and current principal investigators.

7 Equipment

The University of Gothenburg will provide me office space and full access to resources at the Faculty of Humanities, such as usual office equipment (e.g. whiteboard, laptop etc.), and state of the art IT facilities. I will not require any complex computational power, software licences or intellectual property infrastructure. The main professional expenditures in my area are to attend conferences and workshops, for visiting activities and fees for open access publishing, for which adequate funding is being requested within this application.

8 Need for research infrastructure

No research infrastructure is need for the proposed project.

9 International and national collaboration

The proposed project will greatly benefit from a number of national and international collaborations (as anticipated in Section 5.3), consolidating and expanding the current research network of the applicant. I will also discuss the applicant's competences and expertise for each **WP**.

WP1. To develop **T1.1** I will initiate a collaboration with Lukas Melgaard, since that task is meant to generalise his results on cyclic systems for arithmetical inductive definitions [21]. Also, I will achieve **T1.2** by continuing my longstanding and fruitful collaboration with Anupam Das, a world leader on cyclic proofs and their computational aspects. The goal will be to extend and refine our previous joint research contributions [16]. Concerning **T1.3**, I will collaborate with Graham Leigh and Bahareh Afshari at the University of Gothenburg, two world leaders on cyclic proof-theory and finitisation methods of fixed point logics. Finally, to achieve **T1.4**, one of the pillars of the proposed project, I will start a collaboration with Alexis Saurin (Université Paris Cité), who has largely contributed to infinitary cut-elimination techniques for fixed point logics. During my postdoctoral research (2020-2024) I have acquired considerable experience working with theories of second-order arithmetic and Reverse Mathematics, introducing general methods for the analysis of the computational strength of cyclic proof systems with fixed points [16]. Consequently, I will bring to these collaborations a wide-ranging research portfolio of competences. In particular, thanks to my recent contributions on infinitary cut-elimination techniques for non-wellfounded proof systems [1], I will complement Alexis Saurin's expertise within **T1.4**.

WP2. The first two tasks of this work package (**T2.1** and **T2.2**) will be delivered in joint research with a Master student and PhD student. I expect that **T2.2** will require a series of scientific interactions with Thomas Powell (University of Bath), who is a foremost expert in proof interpretations (such as Gödel's *Dialectica*) and proof mining. The cornerstone of this work package (**T2.3**) will be delivered by a series of collaborations with Ugo Dal Lago (University of Bologna) and Hugo Férée (Université Paris Cité), who are leading experts in probabilistic computation and higher-order complexity, respectively. Finally, I will benefit from collaborations with Reuben Rowe (Royal Holloway), who pioneered cyclic automated reasoning and verification, to drive forward the implementation-oriented part of the programme (**T2.4**). As mentioned earlier, my foundational contributions on the complexity-theoretic aspects of cyclic proofs laid the groundwork for CIC [15, 17]. I will exploit these competences for pursuing research within **WP2**. Also, **T2.1-2** will rely on my ongoing supervision experience (see **Curriculum Vitae**).

Key References

[1] M. Acclavio, G. Curzi, and G. Guerrieri. Infinitary cut-elimination via finite approximations. In *CSL*, 2024.

[2] M. Acclavio, G. Curzi, and G. Guerrieri. Non-wellfounded parsimonious proofs and non-uniform complexity. *CoRR*, abs/2404.03311, 2024.

[3] B. Afshari, S. Enqvist, and G. Leigh. Cyclic proofs for the first-order μ-calculus. *Log. J. IGPL*, 2024.

[4] B. Afshari, G. Leigh, and G. Menéndez Turata. Demystifying μ. *CoRR*, abs/2401.01096, 2024.

[5] B. Afshari and G. E. Leigh. Cut-free completeness for modal mu-calculus. In *LICS*, 2017.

[6] D. Baelde, A. Doumane, D. Kuperberg, and A. Saurin. Bouncing threads for circular and non-wellfounded proofs: Towards compositionality with circular proofs. In *LICS*, 2022.

[7] D. Baelde, A. Doumane, and A. Saurin. Infinitary proof theory: the multiplicative additive case. In *CSL*, 2016.

[8] D. Baelde and D. Miller. Least and greatest fixed points in linear logic. In *LPAR*, 2007.

[9] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational complexity*, 1992.

[10] S. Bellantoni and M. Hofmann. A new ”feasible” arithmetic. *Journal of Symbolic Logic*, 2002.

[11] J. Brotherston, N. Gorogiannis, and R. L. Petersen. A generic cyclic theorem prover. In *APLAS*, 2012.

[12] J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *J. Log. Comput.*, 2011.

[13] P. Clairambault. Least and greatest fixpoints in game semantics. In *FICS*, 2009.

[14] L. Cohen, A. Jabarin, A. Popescu, and R. N. S. Rowe. The complex(ity) landscape of checking infinite descent. *POPL*, 2024.

[15] G. Curzi and A. Das. Cyclic implicit complexity. In *LICS*, 2022.

[16] G. Curzi and A. Das. Computational expressivity of (circular) proofs with fixed points. In *LICS*, 2023.

[17] G. Curzi and A. Das. Non-uniform complexity via non-wellfounded proofs. In *CSL*, 2023.

[18] A. Das. A circular version of Gödel’s T and its abstraction complexity. *CoRR*, abs/2012.14421, 2020.

[19] A. Das. On the logical complexity of cyclic arithmetic. *Log. Methods Comput. Sci.*, 2020.

[20] A. Das. On the logical strength of confluence and normalisation for cyclic proofs. In *FSCD*, 2021.

[21] A. Das and L. Melgaard. Cyclic proofs for arithmetical inductive definitions. In *FSCD*, 2023.

[22] C. Dax, M. Hofmann, and M. Lange. A proof system for the linear time μ-calculus. In *FSTTCS*, 2006.

[23] A. De and A. Saurin. Infinets: The parallel syntax for non-wellfounded proof-theory. In *TABLEAUX*, 2019.

[24] J. Fortier and L. Santocanale. Cuts for circular proofs: semantics and cut-elimination. In *CSL*, 2013.

[25] J. Harrison. Formal verification of floating point trigonometric functions. In *FMCAD*, 2000.

[26] M. Hofmann. A mixed modal/linear lambda calculus with applications to Bellantoni-Cook safe recursion. In *CSL*, 1997.

[27] R. Iosif and C. Serban. A complete cyclic proof system for inductive entailments in first order logic. In *LPAR-22*, 2018.

[28] S. Itzhaky, H. Peleg, N. Polikarpova, R. Rowe, and I. Sergey. Deductive synthesis of programs with pointers: Techniques, challenges, opportunities. In *CAV*, 2021.

[29] E. Jones, C.-H. Luke Ong, and S. J. Ramsay. Cycleq: an efficient basis for cyclic equational reasoning. In *PLDI*, 2022.

[30] D. Kuperberg, L. Pinault, and D. Pous. Cyclic proofs, system T, and the power of contraction. *POPL*, 2021.

[31] U. Dal Lago and P. Toldin. A higher-order characterization of probabilistic polynomial time. *Inf. Comput.*, 2015.

[32] D. Leivant. A characterization of NC by tree recurrence. In *FOCS*, 1998.

[33] R. Lepigre and C. Raffalli. Practical subtyping for curry-style languages. *ACM*, 2019.

[34] G. E. Mints. Finite investigations of transfinite derivations. *Journal of Soviet Mathematics*, 1978.

[35] Michael Möllerfeld. *Generalized inductive definitions. The μ-calculus and Π^2_1 -comprehension*. PhD thesis, 2003.

[36] P. M. Neergaard. A functional language for logarithmic space. In *APLAS*, 2004.

[37] I. Oitavem. Implicit characterizations of pspace. In *Proof Theory in Computer Science*, 2001.

[38] R. Rowe and J. Brotherston. Automatic cyclic termination proofs for recursive procedures in separation logic. In *CPP*, 2017.

[39] A. Saurin. A linear perspective on cut-elimination for non-wellfounded sequent calculi with least and greatest fixed-points. In *TABLEAUX*, 2023.

[40] A. Simpson. Cyclic arithmetic is equivalent to Peano arithmetic. In *FOSSACS*, 2017.

[41] C. Sprenger and M. Dam. On global induction mechanisms in a μ-calculus with explicit approximations. *RAIRO*, 2003.

[42] G. Tellez and J. Brotherston. Automatically verifying temporal properties of pointer programs with cyclic proof. *Journal of Automated Reasoning*, 2019.

Description of merits

Describe how the merits you state in the application relate to the proposed research activity, according to the instructions in the call text.

Description of merits (English)

An aspect of my profile that suites the interdisplinary nature of the proposed project is my diverse background. Indeed, I obtained a Bachelor's and Master's degree in Philosophy before pursuing a Ph.D. in Computer Science. Moreover, throughout my studies, I attended several courses in logic, programming and mathematics.

Since my doctoral studies, I have been expanding and consolidating a scientific network of collaborations and research interactions across many countries, especially in France and Italy during my Ph.D. in joint thesis with Paris Diderot University, and in the UK during my first postdoc. The proposed project will greatly benefit from this scientific network for initiating key collaborations.

I am a talented early career scientist, witnessed by the strong record of high quality publications in the foremost conferences of the area, such as three papers published in the A*-ranked conference *LICS* (Logic in Computer Science) and journals, such as the A-ranked *TCS* (Theoretical Computer Science). I am also an independent researcher, having written single-authored papers (see list of publications).

Finally, I have been writing a number of grant applications and attended grant writing workshops. Examples are the application for the *Research Fellow in Science and Engineering of the Royal Commission for the Exhibition of 1851*, the *EPSRC Postdoctoral Fellowship in Engineering and Physical Sciences* (Scores: 6/6, 5/6, 4/6, 4/6), and the application for the permanent research position at the *National Centre for Scientific Research* (*CNRS*) in France, which requires writing a long-term research programme. The grant-writing skills I have been developing over the years will impact positively the proposed project, as they will help me gain a long-term research vision, self-assess my research outcomes, and train my leadership skills.

Publications and other research outputs

Applicant's publications and other research outputs (English)

See following page for attachment

List of publications

Gianluca Curzi

1 Bibliometric information

- total number of peer-reviewed original articles: 9
- total number of citations of the peer-reviewed original articles: 9
- i10 index: 0
- the database used for citation data: Scopus (`https://www.scopus.com`)

2 Selection of research outputs

- **Gianluca Curzi** and Anupam Das. Cyclic implicit complexity. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 19:1–19:13. ACM, 2022.

  Description. In this paper the applicant paved the way to a new topic in computational complexity called *Cyclic Implicit Complexity*, whose development is among the main goals of the proposed project (see **G2** in Section 1 of **Research Plan**). Overall, the paper gives evidence of the applicant's long-term research vision and theory-building skills.

- **Gianluca Curzi** and Anupam Das. Non-uniform complexity via non-wellfounded proofs. In Bartek Klin and Elaine Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic, CSL 2023, February 13-16, 2023, Warsaw, Poland, volume 252 of LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

  Description. This is a followup paper that originates from research hypotheses formulated by the applicant. It extends and generalises the results of the previous paper *Cyclic implicit complexity*. Furthermore, it introduces fundamental proof methods that will contribute to developing **WP2** and, in particular, **T2.3** (see Section 5.1 of **Research Plan**).

- **Gianluca Curzi** and Anupam Das. Computational expressivity of (circular) proofs with fixed points. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2023

  Description. This paper solves a fundamental open question about the computational content of (cyclic and inductive) fixed point logics. It gives evidence of the applicant's problem-solving skills and broad expertise. The methods introduced in this paper will play a key role in achieving main goals of the project (see **G1** in Section 1 of **Research Plan**).

- Matteo Acclavio, **Gianluca Curzi**, and Giulio Guerrieri. Infinitary cut-elimination via finite approximations. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, volume 288 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

Description. The applicant played a central role in this paper, by formulating the initial research hypotheses and motivations, and proving the main technical results. The proof methods developed in this paper will be used to achieve **T1.4** of **WP1** (see Section 5.1 of **Research Plan**).

- Matteo Acclavio, **Gianluca Curzi** and Giulio Guerrieri. Non-wellfounded parsimonious proofs and non-uniform complexity. *CoRR*, abs/2404.03311, 2024.

  Description. This is a recent preprint that explores applications of Cyclic Implicit Complexity to linear logic. The applicant developed the research hypotheses and proved the main results of this paper, which will be used for achieving key goals of **WP2** (see Section 5.1 of **Research Plan**).

3 Relevant peer-reviewed research outputs from 2016–2024

Journals

[1] Alessandro Aldini, **Gianluca Curzi**, Pierluigi Graziani, and Mirko Tagliaferri. A probabilistic modal logic for context-aware trust based on evidence. *International Journal of Approximate Reasoning*, page 109167, 2024.

[2] **Gianluca Curzi**. Linear additives. In Ugo Dal Lago and Valeria de Paiva, editors, *Proceedings Second Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity&TLLA@IJCAR-FSCD 2020, Online, 29-30 June 2020*, volume 353 of *EPTCS*, pages 74–93, 2020.

[3] **Gianluca Curzi** and Luca Roversi. A type-assignment of linear erasure and duplication. *Theoretical Computer Science*, 837:26–53, 2020.

Conference proceedings

[1] Matteo Acclavio, **Gianluca Curzi**, and Giulio Guerrieri. Infinitary cut-elimination via finite approximations. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, volume 288 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[2] **Gianluca Curzi** and Anupam Das. Non-uniform complexity via non-wellfounded proofs. In Bartek Klin and Elaine Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic, CSL 2023, February 13-16, 2023, Warsaw, Poland, volume 252 of LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[3] **Gianluca Curzi** and Anupam Das. Computational expressivity of (circular) proofs with fixed points. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2023

[4] **Gianluca Curzi** and Anupam Das. Cyclic implicit complexity. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 19:1–19:13. ACM, 2022.

[5] Alessandro Aldini, **Gianluca Curzi**, Pierluigi Graziani, and Mirko Tagliaferri. Trust evidence logic. In Jirina Vejnarová and Nic Wilson, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings*, volume 12897 of *Lecture Notes in Computer Science*, pages 575–589. Springer, 2021.

[6] **Gianluca Curzi** and Michele Pagani. The benefit of being non-lazy in probabilistic λ-calculus: Applicative bisimulation is fully abstract for non-lazy probabilistic call-by-name. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 327–340. ACM, 2020. 2

4 Relevant non peer-reviewed research outputs from 2016–2024

Preprints

[1] Matteo Acclavio, **Gianluca Curzi** and Giulio Guerrieri. Non-wellfounded parsimonious proofs and non-uniform complexity. *CoRR*, abs/2404.03311, 2024.

[2] **Gianluca Curzi** and Luca Roversi. Probabilistic soft type assignment. *CoRR*, abs/2007.01733, 2020.

Letter of support

Letter of support from the administrating organisation (English)

See following page for attachment

08/04/2024

UNIVERSITY OF GOTHENBURG

DEPARTMENT OF PHILOSOPHY, LINGUISTICS AND THEORY OF SCIENCE

Department of Philosophy, Linguistics and Theory of Science
Christine Howes
Professor in Computational Linguistics
Deputy Head of Department
Head of Research
christine.howes@gu.se

Swedish Research Council

Letter of Support for Gianluca Curzi's VR Starting Grant

It is my great pleasure to express our strong support for the research proposal *Proofs with Cycles and Computation* of Gianluca Curzi, who joined our department in Autumn 2023 as a two-year postdoctoral fellow.

Curzi obtained his Ph.D. in Computer Science at University of Turin in 2020 under the supervision of Luca Roversi (Turin) and Michele Pagani (Paris Diderot). His Ph.D. studies concerned the complexity of computation in probabilistic systems and strongly related proof calculi with important contributions to so-called lazy evaluation for such systems. During a three-year postdoctoral fellowship in Birmingham (2020–23), Curzi pursued new and challenging avenues of research, including reverse mathematics and cyclic proof theory, that has distinguished him among the community as a promising young researcher, witnessed by invitations to talk at 2023 GT Scalp (Structures formelles pour le calcul et les preuves), Orléans and 2024 MFPS (Mathematical Foundations of Programming Semantics), Oxford.

The coalescence of cyclic proofs and complexity theory is a research programme that developed out of Curzi's natural shift towards proof theory. The value of his work to the logic community speaks for itself giving rise to two successive publications in the ACM/IEEE symposium on Logic in Computer Science (LICS) and a publication in the EACSL conference on Computer Science Logic, the most prestigious conference venues in the area of Computational Logic. Taken together with outputs from his Ph.D. thesis, Curzi's publication record is truly outstanding for his career age: three publications at LICS and a journal article in *Theoretical Computer Science* out of a total of eight peer-reviewed articles.

Curzi's VR proposal outlines the next step in his research vision concerning the interaction of circularity and complexity. The goal of the project is to develop uniform techniques for relating cyclic proofs to paradigms of computation and assessing their complexity. In doing so, Curzi's project will lift the tight correspondence between proof and algorithm to the concepts of general induction and (co-)recursion. The proposed research will, without doubt, further cement Curzi's pioneering programme of research as an outstanding contribution of cyclic proof theory to mathematics and computer science.

Our department is an ideal host for the proposed VR Starting grant. With four senior lecturers in logic (including three docent), of which two, Graham E. Leigh and Bahareh Afshari, are leading experts on the theory and practise of cyclic proofs. The department also has an active Ph.D. programme, with five current Ph.D. students in Logic alone. At Gothenburg, Curzi's project will become part of one the of the largest and strongest logic groups in the Nordic countries and a world-leading centre in the research area of the proposal.

Employment

Curzi is currently employed as postdoctoral fellow at our department on a five-year research project lead by Docent in Logic Graham E. Leigh and funded by the Knut and Alice Wallenberg Foundation with co-funding from the University of Gothenburg. As a VR Starting Grant recipient, Curzi will be employed as Researcher (Forskare) funded by VR (50%), Leigh's research project (25%) and the University of Gothenburg (25%) for two years, followed by VR (75%) and University of Gothenburg (25%) after Leigh's project comes to a close in 2026. This arrangement will allow Curzi to maintain an activity level of at least 75% on his project for the entire four-year period.

The department offers education in logic at all levels and Curzi is already engaged in teaching at the second and third level and in our master's programme in Logic. In his short time in Gothenburg, Curzi has demonstrated himself to be a valuable asset as both teacher and supervisor in our education programmes and we foresee him to grow further in these roles as he continues his career development. As part of Curzi's career plan he will be nominated for pedagogical training courses at Gothenburg University and assigned a departmental mentor to support his progression towards obtaining docentship in the coming years.

Best Regards,

Christine Howes
Deputy Head of Department and Head of Research

Budget and research resources

Activity level in the project

Role in the project	Name	Percent of full time
1 Applicant	Gianluca Curzi	75%

Flat-rate amount

The applied amount in this call for proposals is a fixed amount wich is added automatically.

Flat-rate amount

Standard amount	2025	2026	2027	2028	Total
StandardAmount	1,100,000	1,100,000	1,100,000	1,100,000	4,400,000

Justification of the budget applied for (English)

I will dedicate 75% of the time to the proposed research project. In the first two years (2025 and 2026) my salary will be complemented by the research project *Taming Jörmungandr: The Logical Foundations of Circularity* (2021–2026), funded by the *Wallenberg Academy Fellowship Prolongation* grant (Project code: 251080003) and led by my current postdoctoral supervisor Graham Leigh. Specifically, Graham Leigh's project will cover 25% of my salary (126000SEK) in the first two years, while 50% of the salary will be covered by the VR grant. In the last two years, the VR grant will cover the full 75% of my research. The remaining 25% of my full-time employment for the entire duration of the project will be financially supported by other research funding and teaching.

The scientific outcomes of the proposed project will be published in the proceedings of peer-reviewed international top conferences, such as *LICS* (*Logic in Computer Science*), *POPL* (*Symposium on Principles of Programming Languages*), *CSL* (*Computer Science Logic*) and *CPP* (*Certified Programs and Proofs*), and high-impact scientific journals such as *TCS* (*Theoretical Computer Science*), *LMCS* (*Logical Methods in Computer Science*) and *MSCS* (*Mathematical Structures in Computer Science*). The overall budget estimation is 160000 SEK (40000SEK per year).

The project also includes visiting periods. One of my collaborators, Lukas Melgaard, will undertake a one-month visiting period in Gothenburg. The cost is roughly 20000SEK (12000 accommodation + 4000 flights + 4000 meals). Furthermore, I will undertake a three-month visiting period in Paris (Université Paris Cité) to collaborate with Alexis Saurin. The cost is around 50000SEK (38000 accommodation + 4000 flights + 8000 meals).

Finally, the University of Gothenburg will support my commitment to open access publishing during this research project. I will require 21100SEK for open access IEEE proceedings.

Other funding for this project

	Funder	Applicant/project leader	Type of grant	Status	Reg no or equiv.
1	Knut och Alice Wallenberg Stiftelse	Graham Leigh	Wallenberg Academy Fellowship Prolongation	Granted	251080003
Total					

	2025	2026	2027	2028	Total
1	126,000	126,000	0	0	252,000
Total	126,000	126,000	0	0	252,000

CV

CV - Gianluca Curzi

Project leader: Gianluca Curzi
Birthdate: 19910129
Gender: Male
Country: Sweden
Doctoral degree: 2020-06-12
Academic title: Doctor
Employer: Göteborgs universitet

Doctors degree

Examination	Organisation	Dissertation title (original language)	Supervisor
10201. Computer Sciences, 2020-06-12	University of Turin, Computer Science	Non-Laziness in Implicit Computational Complexity and Probabilistic λ-calculus	Luca Roversi

Educational history

Research education

Examination	Organisation	Dissertation title	Name of supervisor
PhD degree, 10201. Computer Sciences, 2020-06-12	University Paris 7 Diderot, France, Informatique	My PhD was based in the University of Turin, in joint thesis with Université Paris Diderot	Michele Pagani
PhD degree, 10201. Computer Sciences, 2020-06-12	University of Turin, Italy, Computer Science	Non-Laziness in Implicit Computational Complexity and Probabilistic λ-calculus	Luca Roversi

Basic education

Year	Examination
2016	60301. Philosophy, Degree of master (120 credits), University of Florence, Italy
2013	603. Philosophy, Ethics and Religion, Degree of Bachelor, University of Urbino, Italy

Professional history

Employments

Period	Position	Part of research in employment	Employer
september 2023 - augusti 2025 (Present)	Postdoctoral fellow, Project employment	100	University of Gothenburg, Sweden, Filosofi, lingvistik och vetenskapsteori, inst för
oktober 2020 - augusti 2023	Postdoctoral fellow, Project employment	100	University of Birmingham, United Kingdom, Computer Science

Post doctoral assignments

Period	Organisation	Subject
september 2023 - augusti 2025	University of Gothenburg, Sweden, Filosofi, lingvistik och vetenskapsteori, inst för	60301. Philosophy

Period	Organisation	Subject
oktober 2020 - augusti 2023	University of Birmingham, United Kingdom, Computer Science	10201. Computer Sciences

Merits and awards

Supervised persons

Year	Supervised persons	University (supervisee)	Role
2024	PhD student, Lukas Melgaard	University of Birmingham, United Kingdom, Computer Science	Secondary supervisor
2024	Student, Bente Gortworst	University of Gothenburg, Sweden, Filosofi, lingvistik och vetenskapsteori, inst för	Secondary supervisor

Awards and distinctions

Year	Country	Name of award/distinction	Issuer	Description
2018	Italy	Scholarship "Bando Vinci" (5500 euros).	Université Franco-Italienne	Description: The "Vinci" programme focuses on funding PhD students in joint thesis ("cotutelle") with a French university.

Other merits

Period	Type of merit	Description
2022 - 2024	grant writing	I have written several grant applications and attended grant writing workshops. In particular, in January 2022 I have applied for the Research Fellow in Science and Engineering of the Royal Commission for the Exhibition of 19851. In July 2022 I applied for the EPSRC Postdoctoral Fellowship in Engineering and Physical Sciences (Scores: 6/6, 5/6, 4/6, 4/6). I also applied for the permanent research position at the National Centre for Scientific Research (CNRS) in France, which requires writing a long-term research programme.
2018 - 2024	Paper review	I have been reviewing papers for top-ranked conferences (such as LICS, CSL, CIFMA, ESOP) and journals (such as ACM, Logic Journal of the IGPL).
2021 - 2022	PC member	I was member of the program committee for the workshop CIFMA (Cognition: Interdisciplinary Foundations, Models and Applications) from 2021 to 2022.
2019 - 2019	Subject-matter expert in Logic and Computer Science.	Department of Pure and Applied Sciences, University of Urbino, Italy Description: Subject-matter expert ("Cultore della Materia") is an Italian academic title that qualifies those members of the academic staff who are specialised on a particular field and work as teaching assistants and exam committee members on a voluntary basis.

Publications

Publications - Curzi, Gianluca

Project leader: Gianluca Curzi
Birthdate: 19910129
Gender: Male
Country: Sweden
Doctoral degree: 2020-06-12
Academic title: Doctor
Employer: Göteborgs universitet

Register

Terms and conditions

The application shall be signed by the applicant and also by an authorised representative of the administrating organisation. The representative is normally the head of the department where the research will be carried out, but this is dependent on the administrating organisation's structure.

The *applicant's* signature confirms that

- the information in the application is correct and complies with the Swedish Research Council's instructions
- secondary occupations and commercial ties have been reported to the administrating organisation and that nothing has emerged that breaches good research practice
- the permits and approvals required have been obtained before the research is started, such as permits from the Swedish Medical Products Agency or approval from The Swedish Ethical Review Authority or an ethical committee on animal experiments
- the applicant will comply with all other conditions applicable to the grant.

The signature of the *administrating organisation* confirms that

- the research or research-supporting activities described can be given room at the administrating organisation during the period and to the extent stated in the application
- the applicant will be employed by the administrating organisation during the period covered by the application
- the administrating organisation approves of the budget in the application
- the administrating organisation will comply with all other conditions applicable to the grant.

The above points shall have been discussed by the parties before the representative of the administrating organisation approves and signs the application.