

**Start Page**

**MARIE SKODOWSKA-CURIE ACTIONS**

**Individual Fellowships (IF)  
Call: H2020-MSCA-IF-2015**

**PART B**

**“SATIR”**

**“STATIC ANALYSIS VIA TYPE INFERENCE”**

**THIS PROPOSAL IS TO BE EVALUATED AS:**

**[GLOBAL FELLOWSHIP (GF)]**

# Contents

<b>1</b>	<b>Summary</b>	<b>3</b>
<b>2</b>	<b>Excellence</b>	<b>3</b>
2.1	Quality, innovative aspects and credibility of the research (including inter/multidisciplinary aspects) . . . . .	3
2.2	Clarity and quality of transfer of knowledge/training for the development of the researcher in light of the research objectives . . . . .	6
2.3	Quality of the supervision and the hosting arrangements . . . . .	6
2.4	Capacity of the researcher to reach and re-enforce a position of professional maturity in research . . . .	8
<b>3</b>	<b>Impact</b>	<b>8</b>
3.1	Enhancing research- and innovation-related human resources, skills, and working conditions to realize the potential of individuals and to provide new career perspectives . . . . .	8
3.2	Effectiveness of the proposed measures for communication and results dissemination . . . . .	8
<b>4</b>	<b>Implementation</b>	<b>8</b>
4.1	Overall coherence and effectiveness of the work plan, including appropriateness of the allocation of tasks and resources . . . . .	8
4.2	Appropriateness of the management structure and procedures, including quality management and risk management . . . . .	9
4.3	Appropriateness of the institutional environment (infrastructure) . . . . .	9
4.4	Competences, experience and complementarity of the participating organisations and institutional com- mitment . . . . .	10
<b>5</b>	<b>CV of the Experienced Researcher</b>	<b>10</b>
<b>6</b>	<b>Capacities of the Participating Organisations</b>	<b>12</b>
<b>7</b>	<b>Ethical Aspects</b>	<b>12</b>
<b>8</b>	<b>Letters of Commitment of Partner Organisations</b>	<b>12</b>

## List of Participants

Participants	Legal Entity Short Name	Academic (tick)	non-Academic (tick)	Country	Dept./ Division/ Laboratory	Supervisor	Role of partner organization
<u>Outgoing host</u>							
University of Pennsylvania	UPenn	✓		USA	Department of Computer and Information Science	Steve Zdancewic	
<u>Return host</u>							
University of Birmingham	Bham	✓		UK	School of Computer Science	Dan R. Ghica	

Sections 1,2,3 and 4 are limited to a total of 10 pages.

# 1 Summary

Software is becoming increasingly complex and critical (autonomous vehicles, the internet of things, financial systems, etc.). Bugs, security vulnerabilities, or inefficient resource management are costly. *Type systems* are a scalable way to analyze software but is hindered by a certain lack of expressiveness. *Graded type systems* can improve the state-of-the-art by including more information in the types and by incorporating some inspiration from *abstract interpretation*, another *static analysis* method which is extremely expressive, but as opposed to types lacks compositionality. *Compositionality* is important because it gives great support for *scalability*.

Functional programming languages often use complex type systems describing the behavior of the programs. These types are statically inferred before the compilation, acting as a first test for the correctness of the program. The advantage of type systems is their inherent compositionality. This means that libraries come with pre-computed types which are easily checkable. However, the verification power is low, restricted to the fact that composition of programs never fails due to incompatibility of format.

More refined static, whole-program analysis can generally be performed by abstract interpretation (AI). Those techniques are very efficient but fail to compose. This means that any new analysis has to be performed through the whole program, including libraries. Such a global analysis has two flaws: it is often too resource consuming to be used on a regular basis, and the behaviors of common libraries are approximated while it could be interesting to manually optimize their interpretation once and for all.

In parallel, several functional languages such as Coq, Agda, F\* or Idris offer to associate to programs certain proofs of correctness encoded in a dependent type system. These dependent type systems are extremely expressive: not only can they catch up to any abstract interpretation analyses but they can describe many more refined behaviors. Moreover, these systems are still compositional and can produce certificates. However, the automatic inference is hardly ever automatisable, so the programmer has to produce the proofs mostly manually, even though the system can provide some help.

Notice, also, that performing statical analysis under richly typed languages should be advantageous: advanced type systems offer several ways to encode semantical information inside the type (for readability or for safety). However, actual tools are basically forgetting the type of a program before running the analysis, losing all this semantical information given by the programmer.

In this project, we propose to investigate a way to recover as much information as possible from type inference which is one of the most complex kinds of compositional and static analysis. We do not intend to stamp out global analyses that are proven more efficient in the general case. The expected outcome would be a static analyzer for functional languages that do not fall too far behind existing ones, while being more scalable and able to interact with other type constraints.

For this purpose, we are focusing on emerging parameterized type systems. The parameters represent either qualitative statements or quantitative information over resources. Such type systems will be called *graded types systems*.

A good point of comparison would be Hoare triples with preconditions and postconditions that need to match during composition, but some differences must be emphasized:

- A program is not only associated with a precondition and a post-condition corresponding to the parameterization of the input and output types, but also with **higher order conditions** parameterizing every subtype (especially arrow types).
- All of these conditions are dependent over “resource variables” that are instantiated by unification during the composition (adding some new constraints). In particular, this allows any program to have a **most general type** that characterizes it; this most general type is computed once and can be reused each time the program is used in a larger project.
- Constraints are not logical propositions but element of an **algebraic structure**. These structures take shape of ordered monoids or ordered semirings. They are reminiscent of abstract interpretation domains.

The SATIR project has three objectives. The **first** (and main) **objective** is to develop the theory of graded type systems in order to be as expressive as possible, to adapt to existing functional languages and to recover AI’s constructions (Galois relations and widening). The **second objective** is to track existing instances of graded type systems and to integrate them in our general framework. The **third objective** is to select a fragment of the system for the purpose of implementation: this fragment need to be inferenceable up-to the use of some widening and calls over an SMT solver (to resolve algebraic constraints).

## 2 Excellence

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

The SATIR project intends to specify a particular subclass of abstract interpretation that we call graded type systems. Another way to state it is that SATIR project intends to reinject the whole power of abstract interpretation inside

a type system by substituting the choice of the domain by the choice of some more refined algebraic structures called grading structures.

The project can be separated in three distinct objectives. The **Objective 1** is the main objective which intends to build up a comprehensive **theory of graded types** in its most general version. The **Objective 2** pursues real **applications**. The intent behind this objective is both to get some inspiration for the general case and to disseminate our results. Finally, **Objective 3** is the delivery of an **implementation** for the end of the three years project.

**State of the art.** In 2001, Patrick Cousot was writing that “the most severe restrictions [of type inference] are on the considered properties”<sup>1</sup>. The situation has now changed and type systems are able to treat a large variety of problems.

Our first example was developed in late 90’s in the security community: the use of types for information-flow analyses.<sup>2</sup> The idea consists in automatically inferring the security level of a program. For example, a program  $\lambda xy.x : l.int \rightarrow l.int \rightarrow l.int$  use low security assertion and only take lowly secured input resulting in a lowly secured output. The apparent weakness of this type system is that a program may have a lot of different types, in particular  $\lambda xy.x$  can also have the types  $h.int \rightarrow l.int \rightarrow h.int$  or  $l.int \rightarrow h.int \rightarrow l.int$ . To resolve this weakness, we remark that a lowly secured assumption can always be assumed highly secured so that  $l.int \rightarrow \tau$  is less precise than  $h.int \rightarrow \tau$ ; moreover, one will never create highly secured objects, so that we always assume that the result is lowly secured by default. With these assumptions, the main type<sup>3</sup> of  $\lambda xy.x$  is  $l.int \rightarrow h.int \rightarrow int$ .

The above is (one of) the most basic instance of BLL-like type systems that we will develop later. Nowadays, information-flow analysis has been developed further along four axes:<sup>4</sup> the expressiveness, the concurrency, the covert channel (modularity regarding the observation function of the attacker) and the security policy (allowing restricted declassifications). A special consideration will be applied to strengthen graded type system toward each of these directions (WP 1.3.1, WP 2.2.2 and WP 2.3.2).

Flavien to Steve: I kinda remember that you had several papers in this area, maybe can you add a few sentences/references???

Linear types were introduced in different areas, and in particular for finer information-flow analysis. It consists at insuring that some argument is used exactly once (or at most once), which is useful for a more efficient compilation or for security. A more refined version of a linear types system is a multilinear type system where arguments are tagged with an integer bounding their number of use. Even more interesting, this tag can be inferred automatically via type inference. Below is an example of a type derivation in call by name:

$$\frac{x : 1.int \vdash \lambda f.f (f x) : 2.(1.int \rightarrow int) \rightarrow int \quad x : 1.int \vdash \lambda y.y + x : 1.int \rightarrow int}{x : 3.int \vdash (\lambda f.f (f x)) (\lambda y.y + x) : int} \quad 2*1+1=3$$

The function  $\lambda f.f (f x)$  has type  $2.(1.int \rightarrow int) \rightarrow int$  meaning that it uses twice its argument  $f$  which, itself is using its argument once. In the end, we know that the argument  $x$  is used **3** times: once called by the function  $\lambda f.f (f x)$  and twice called by the function  $(\lambda y.y + x)$  (that is linear but used twice).

Being biased toward linear logic, I fear that if I give a reference here I would not be fair.

One of the culminating point of information-flow security area is the language DFuzz<sup>5</sup>. DFuzz was developed by the host team conjointly with the security team and a former Marie SKODOWSKA-CURIE fellow. This language extends over the covert channel direction by providing only differentially secured programs. Extending the idea of linearity to probabilities, programs of DFuzz are of type  $r.\sigma \rightarrow \tau$ , where  $r \in \mathbb{R}^{\geq 0}$  is the expected value over the number of use of the argument. A novelty, here, is that the resource  $r$  can also depend over some “resource variable” corresponding, for example, to the number of loop in a recursion.

DFuzz was also openly inspired by a fairly older work independent from information-flow type system: the logic BLL.<sup>6</sup> BLL is a rich logic which, as a type system, insure the polynomial complexity of typed terms. The resulting language is then sufficiently expressive to allow the resolution of any polynomial problem. Basically, a BLL function type can be seen as  $p.\sigma \rightarrow \tau$  where  $p$  is a polynomial. But this naive description is far from enough to explain BLL’s expressiveness.

The real power of BLL is to allow a dependency over resources that DFuzz’s notion of dependency, in the sense that resource variables are bound by the parameters. Indeed, functions terms are in fact of types  $(\alpha < p).\sigma \rightarrow \tau$  where  $\alpha$  is a resource variable free in  $\sigma$  and bound in  $(\alpha < p).\sigma$ , the substitution is then performed along the type derivation (during dereliction phase mainly). If DFuzz’s dependency is a “resource polymorphism”, this dependency is a real “dependant resource” comparable to dependant types. This point is still not understood enough and the only other system using a real resource dependency is DℓPCF<sup>7</sup> which is directly derivated from BLL.

<sup>1</sup>Patrick Cousot. “Abstract interpretation based formal methods and future challenges (Electronique version)”. In LNCS 2001.

<sup>2</sup>F. Nielson, H.R. Nielson and C. Hankin. “Principles of program analysis”, 2015, In Springer

<sup>3</sup>assuming that  $y$  have to be an integer

<sup>4</sup>A. Sabelfeld and A.C. Myers, “Language-based information-flow security”, 2003, In IEEE journal on selected areas in communications

<sup>5</sup>M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan and B.C. Pierce

<sup>6</sup>G-Y. Girard, A. Scedrov and P. Scott. “Bounded linear logic: a modular approach to polynomial-time computability”. 1992. In Theoretical computer science.

<sup>7</sup>U.D.Lago and M. Gaboardi. “Linear dependent types and relative completeness”. 2011. In Logic in Computer Science (LICS).

Inspired by all these previous languages, two groups of peoples<sup>89</sup> (which both host supervisors are part of) did independantly generalized them into a sole system parametrised by an (ordered) semiring  $\mathcal{S}$ : The BLL-like systems or the graded comonad. This system is obtain by a parameterization (or grading) of the exponential comonad of the linear logic. In particular, the sum of the semiring corresponds exactly to the contraction and the multiplication correspond exactly to the digging.

Functions types, in these systems, are of the form  $s.\sigma \rightarrow \tau$  for  $s$  an element of a chosen semiring  $\mathcal{S}$  which can be, for example, Booleans (for information-flow security), natural numbers (for linearity), reals number (for probabilities) or polynomials (for complexity). Ghica and Smith even introduced a new important application: the use of contractive affine transformations to characterize the Sequentiality of a program (with scheduling objectives). Notice, however, that graded comonads definitely lack of any kind of dependency.

A dual to graded comonads has been independantly introduced: the graded monads.<sup>10</sup> If graded comonads represent the backward flow of information and the requirements over the context, graded monads represent the forward flow and the actions preformed by the program. Even if we said that graded monads are dual of graded comonads, those are simpler: they are parametrised by a monoid rather than a semiring since the linearity is not at stack for forward flow analysis.

Graded monads and comonads can be used together through a graded distributive law. This phenomenon has been described in a submitted paper by the fellow and co-authors.<sup>11</sup> This interaction is very rich and permit to take into account both the actions for the requirements' calculation, and the consumptions (requirements) for the productions' calculation. Moreover the choices of the considered monad and comonad are independent, adding a lot of modularity.

The requirements and actions (also called coeffects and effects), can be seen as pre- and post-conditions. However, since a part of the information can flow forward and the other can flow backward, the symbolic representations for pre- and post-conditions are disjoint. This is to oppose to other systems such as Haskell refinement types or Schopenhauer types.<sup>12</sup> A surprising point is the difficulty to naturally model these systems into the graded monad-comonad interaction. This comes from the intrinsically linear nature of graded comonad that can model non-linear behaviors but with a lack of naturality. This point has been captured explicitly by Petrieck *et al*'s notion of shapes.<sup>13</sup>

Finally, it is worth to notice the recent works of Grellois and Melliés that shows that higher-order model checking can be performed by a type inference in a system with a graded comonad extended to intersection types.<sup>14</sup> By integrating the algorithm in our language, we can hope to perform efficient shape analysis over functional data structures.

**Methodology for Objective 1 (Theory).** This objective can be split in three successive goals each of them containing three work packages. **Goal 1.1** consists in identifying the exact nature of grading structures. **Goal 1.2** aims at transporting abstract interpretation principles to graded type systems. Finally, **Goal 1.3** extends the result in several directions (generalizations to a full fledged language and to intersection type systems).

**Goal 1.1** is aiming to be the basis of the whole project. At such, we expect the resulting system to respect many strong properties. First, its construction should be logic- and semantic-oriented, which is the main interest of WP 1.1.1. Moreover, the systems will be presented with a strong soundness/completeness result between their operational semantic, categorical axiomatisation and concrete models.<sup>15</sup> Notice that the fellow is an expert of denotational semantics and found the first denotational model of graded comonads.<sup>16</sup>

Work package 1.1.1 is a package that focuses on grading non conventional models of linear logics, monads and comonads in order to investigate natural extensions of graded type systems. A lot of studies implies strong models of linear logic, of monads or of comonads, that are containing far more semantical information than the syntactical object they are modeling. We will try to extract a part of these information in a graded (co-)monad and see is there is natural syntactical way to extract the remaining information.<sup>17</sup>

Work package 1.1.2 has the objective to specify the full notion of dependence over grading structures. We aim at merging the notion of graded comonad and the exponential of the original BLL logic. This point is difficult, but early results over WP 1.1.1 shows that we can get close to this objective by generalizing the algebraic structure behind graded comonad from a semiring to a certain kind of semiringoid (that I called dependant semiring) with an external action.

In work package 1.1.3 we aim at decomposing further the separation between graded monad, comonad and dependency in order not to forget any kind of hidden structure. In particular, the issue from Petrieck *et al*'s notion of shapes.<sup>18</sup> regarding flat and structural shapes may indicate that the notion of graded comonad is not sufficiently fine yet. In order to perform this decomposition, we will destructure graded monads and comonads into graded adjunctions.

<sup>8</sup>A. Brunel, M. Gaboardi, D. Mazza and S. Zdancewic. "A core quantitative coeffect calculus". 2014. In ESOP.

<sup>9</sup>D.R. Ghica and A.I. Smith. "Bounded linear types in a resource semiring". 2014. In ESOP.

<sup>10</sup>S. Katsumata. "Parametric effect monads and semantics of effect systems". 2014. In POPL.

<sup>11</sup>F. Breuvar, M. Gaboardi, S-Y. Katsumata and D. Orchard. "Combining effects and coeffects". 2016.

<sup>12</sup>K. Hammond, M. Hofmann, S. Jost, H.-W. Loidl. "Static determination of quantitative resource usage for higher-order programs". In ACM SIGPLAN-SIGACT POPL10.

<sup>13</sup>T. Petricek, D. Orchard, and A. Mycroft. "Coeffects: A calculus of context-dependent computation". 2014. In CM SIG-PLAN ICFP

<sup>14</sup>C. Grellois and P-A. Melliés. "Relational semantics of linear logic and higher-order model-checking". 2015. In CSL.

<sup>15</sup>Or at least for several key examples since a general notion of operation is not expected.

<sup>16</sup>F. Breuvar, M. Pagani. "Modelling Coeffects in the Relational Semantics of Linear Logic". 2015. In CSL.

<sup>17</sup>See Section 3.2 of the fellow's thesis that will be the object of a forthcoming paper.

<sup>18</sup>T. Petricek, D. Orchard, and A. Mycroft. "Coeffects: A calculus of context-dependent computation". 2014. In CM SIG-PLAN ICFP

**Goal 1.2** also has three different work packages. Work package 1.2.1 is a learning-focused package that aims at expanding the knowledge of the applicant regarding abstract interpretation. Work package 1.2.2 explores in more detail the order that naturally arises from grading structures in order to link it with domain theory. Work package 1.2.3 is focusing on fixpoints, and in particular on the corresponding notion of widening.

In addition to these work package, a special attention will be given so that the final type system will have a strong property of more general types. In fact we will require that the set of types of a term ordered by subtype is a complete lattice with computable joints.<sup>19</sup> This means that several type analysis over a term can be combine in order to get a more general type. In particular, a type can be obtained by automatic inference, while an other can be given by the programmer as a type specification.

Description of WPs, precisising that we aim at classifying graded algebras.

**Goal 1.3** consists in optional extensions of the theoretical work. In particular, Work package 1.3.1 is aimed at extending the theory for a  $\lambda$ -calculus to a full fledged language and Work package 1.3.2 is investigating the extension to intersection types. \*\*\*

Description of WPs

**Methodology for Objective 2 (Applications).** There are three phases, or goals, regarding the applications. **Goal 2.1** aims at studying and internalizes actual fully formed instances of graded type systems such as those of BLL,  $D_\ell$ PCF or DFuzz.<sup>20</sup> **Goal 2.2** focuses on qualitative examples and **Goal 2.3** focuses on quantitative ones.

In construction

- information/control-flow
- HO model checking <sup>21</sup>
- Sequentiality <sup>22</sup>
- Probabilities <sup>23</sup>
- (complexity analysis) <sup>24</sup>

**Methodology for Objective 3 (Implementation).** This line of work also takes into account the studies relative to the whole type inference process which form **Goal 3.1**. The implementation itself is the **Goal 3.2**, we are aiming for a simplified functional language where the user only provides the grading structure and heuristics for the widening. The objective also contains **Goal 3.3** which is pure learning over later-phase and future complements such as the use of SMT-solvers.

In construction

Novelty, timeliness and quality.

In construction

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

## 2.3 Quality of the supervision and the hosting arrangements

**Outgoing Host Institution** The University of Pennsylvania, founded in 1740 by Benjamin Franklin, is a premier institution providing doctoral education in the United States, and has offered Ph.D. programs for over one hundred years. The University has pre-eminent scholars in all of its disciplines. 21 Nobel Prizes have been awarded to University of Pennsylvania faculty and alumni, and 6 Engineering faculty are in the National Academy of Engineering.

Founded in 1972, the Department of Computer and Information Science of the University of Pennsylvania is part of the school of School of Engineering and Applied Science and is one of the birthplace of the modern computer. It was here that the ENIAC, the world's first electronic, large-scale, general-purpose digital computer, was developed in 1946. Nowadays, the Department of Computer and Information Science combines energies of professors, researchers and doctoral students resulting from different research areas: Artificial Intelligence, Graphics, Information Management, Software Principles, Systems, Theory. The faculty of the Department of Computer and Information Science counts 30 primary faculty members as well as 500 undergradates, and 300 graduate students (Masters and PhD).

Outside from the hosting group (Programming Language group), collaborations are expected with the "Logic and Computation group" (objective 1.1) and the "Security and Privacy group" (objectives 2.1 and 2.2) . Notice that these groups already share numeral active collaborations with the host group.

<sup>19</sup>in fact we only need an upper bound to be computable

<sup>20</sup>See paragraph "State of the art"

<sup>21</sup>Ong, Kobayashi, Mellies, Grellois

<sup>22</sup>Ghica, Smith

<sup>23</sup>DFuzz

<sup>24</sup>BLL, DIPCF

**Outgoing Phase Group** Among the different research groups, the Programming Language group (Penn PL club) works on several specific topics in programming language theory and practice related to the research topic of this project. In particular, it studies advanced type systems including subtyping, functional-oriented languages, ad hoc polymorphism, dependant types, polytypic programming, substructural type systems, language design and implementation, language-based security, metaprogramming, as well as foundations for concurrent, distributed, and mobile computing.

The PL club actually counts 3 professors (the supervisor, Benjamin C. Pierce and Stephanie Weirich) and more than 10 doctoral students and post-doc researchers. The group stand as one of the worldwide top theoretical programming language groups with yearly publications in ACM POPL.

**Outgoing Phase Supervisor** Steve Zdancewic is reconised as an international expert of functional programing languages and security. His research interests lie in theoretical computer science, programming language design and implementation (Coq verification of LLVM program transformations and randomized algorithms, type-directed program synthesis, linear types and GUI programming), language enforcing verification (language-based enforcement of information-flow policies, low-level code memory safety, understanding dynamic security policies, and authorization logic) as well as secure concurrent and distributed computing, functional programming languages, type theory, linear and modal logics, theorem proving and mechanized metatheory.

Steve Zdancewic received the Ph.D. in Computer Science from Cornell University in 2002 and imediatly joint the University of Pennsylvania as Assistant Professor. Hi is now full profesor in the University of Pennsylvania since 2014. He is the author of more than fifty publications in international journals and conferences.

**Return Host Institution** The School of Computer Science of the University of Birmingham is internationally leading in several areas, including foundations of computer science and security. The foundations group includes 9 Academic Staff and the security group 7. In REF 2014, 29% of its research output was judged in the 4\* (world-leading) category, and 57% was judged 3\* (internationally leading). Overall, 100% of its output is of international standard. It has a thriving research environment following its rapid research-led expansion in the last ten years, having c.60 research students, c.30 research fellows, and 36 academic staff, six regular seminar series.

**Return Host Group** In addition to the host supervisor, the Theory group includ numerous strong figures in fields that are particularly relevent for us: Krishnaswami, Escardo are reconised specialists of linear and dependent type systems which graded type system is a refinement. Levy, Reddy are specialists of categorical semantics of programming languages and monads in particular. Jung, Vickers (and Escardo) are specialist of domain theory and can help with the formalisation of graded algebras. Finally, Ritter is a former specialist in linear logic and is now working on security issues, as such he is a potential collaborator for Goals 2.2 and 2.3.

**Return Host Supervisor (Scientist in charge)** Dan Ghica

CV



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

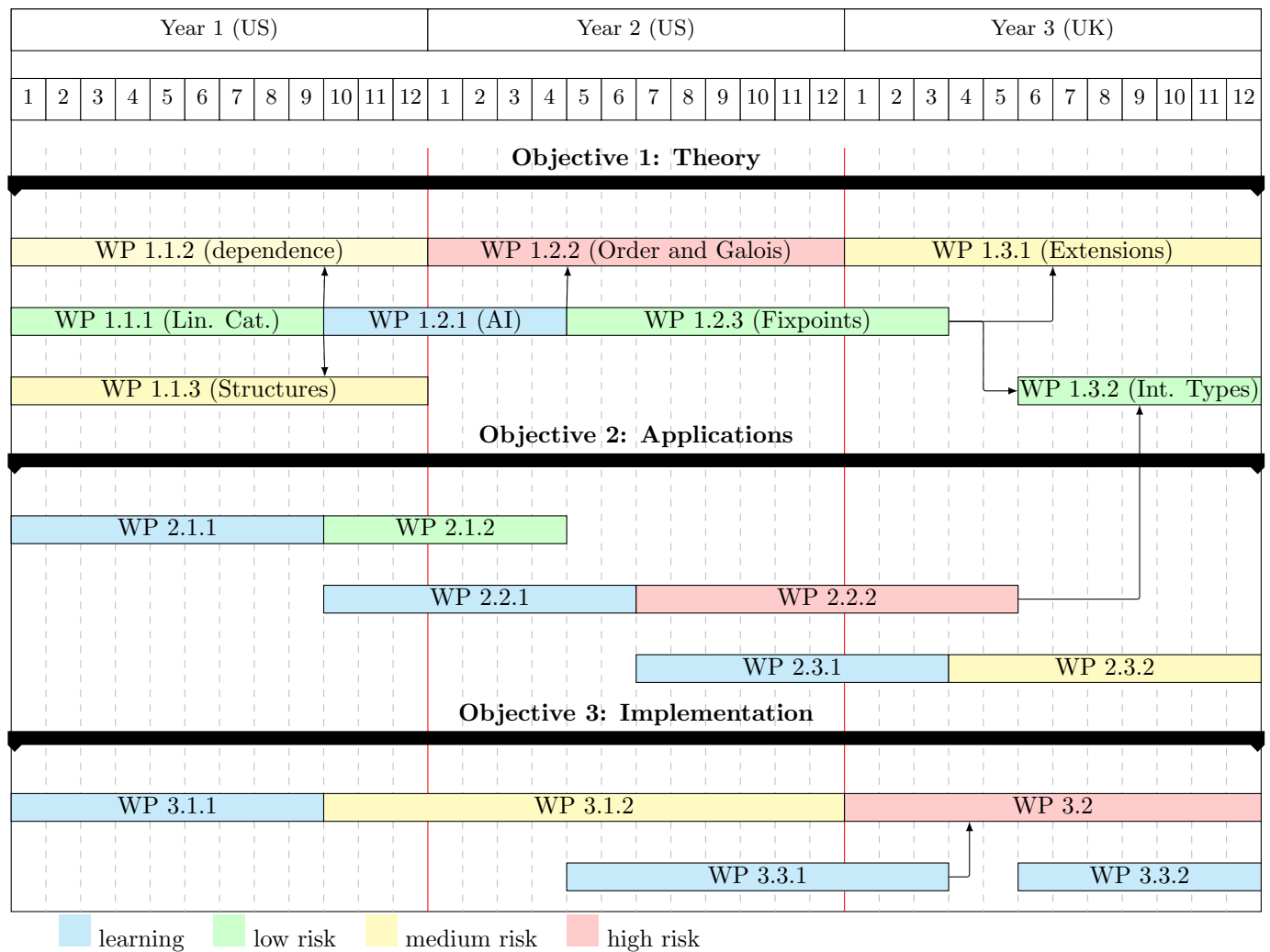
## 3 Impact

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

3.2 Effectiveness of the proposed measures for communication and results dissemination

## 4 Implementation

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



The project is divided in three parallel lines. The theory line focuses on developing the main theory behind our project. The Application line has for goal to understand related fields and to apply our result to those fields. The Implementation line is targeting an implementation of our language.

**Theory:** The main objective follows a sequential pattern.

Goal 1.1 is supposed to be the basis for all the remaining work, it is thus scheduled the first year. Notice, moreover, that no other research-focused work package has been planned in parallel (except for the three last months that are assumed to be in final phase). Among Goal 1.1's work packages, WP 1.1.1 has a support role regarding packages 1.1.2 and 1.1.3, and no important direct result are expected here. Work package 1.1.2 and Work package 1.1.3 should carry important result that will be presented in the form of two papers. Early results on these two directions are quite promising and several researchers are active in this area; thus expecting a reasonable formalisation before the two year deadline<sup>25</sup> can be considered medium risk.

<sup>25</sup>Recall that the fellow intend to begin the fellowship in October 2016.

Goal 1.2 is at the heart of this project and investigates the AI-related features of graded type systems. It is planned the second year, but is extended with preliminaries and some remaining work. We expect WP 1.2.2 to end with the publication of a notion of Galois relation between grading structures; probably the dangerous package, if needed, the study is allowed to overrun its allocated time by delaying WP 1.3.1 which is more modular. Work package 1.2.3 is focusing on fixpoints, and in particular on the corresponding notion of widening which should be the object of an article; this package is considered low risk once the Galois connection has been properly set.

Goal 1.3 consists in optional extensions of the theoretical work.

**Applications:** The applications line follows a classic 2-2-2 pattern corresponding to the three applicative objectives, each separated into a learning phase (9 months) and an active phase (duration varying with the difficulty level). Notice that this objective is composed only of collaborative material.

Goal 2.1 (work packages 2.1.1 and 2.2) targets the embedding of BLL,  $D_\ell$ PCF and DFuzz in our system. It is considered low risk as these languages are well established so that we roughly have to verify that we are taking into consideration each of its components. This objective is mainly aiming at supporting Goal 1.1 with real examples. Moreover, this goal has been scheduled just after one year spent in Focus team (Bologna) that created  $D_\ell$ PCF and in the UPen's programming language team that created DFuzz.

Goal 2.2 (work packages 2.2.1 and 2.2.2) targets the embedding of qualitative examples such as the control flow analyzes and<sup>26</sup> the higher order model checking. This objective is classed high risk for several reasons: this will be the first new application and the relation between graded type systems and higher order model checking is not yet fully established. Its scheduling at middle position among applicative goals permit to manage the high risk by potentially extending the deadline. Moreover, the parallelizing of Goal 2.2 and the work package 1.2.3 is important as the fixpoint construction is fundamental in higher order model checking application. Finally, its scheduling permit a potential additional investigation over intersection type systems at the end of the project.

Goal 2.3 (work packages 2.3.1 and 2.3.2) targets the embedding of quantitative examples such as probability or scheduling. This goal is classified medium risk despite their intrinsic difficulties since the applicant has some experience with these cases. Remark that this goal is planed in Birmingham with Dan Ghica that is an expert of the scheduling example (which we believe critical).

## Implementation:

Not yet fixed  
 WP 3.1.1: learning type inference  
 WP 3.1.2: working on type inference of graded type systems  
 WP 3.2: concrete implementation  
 WP 3.3.1: Learning over SMT solvers  
 WP 3.3.2: optional learning

**Milestones and deliverables:** There are three main milestones at the end of every year and three secondary milestones around midyear.

The first secondary milestone (around month 9) does not have a deliverable but it marks the end of the adjustment and warm-up period. In particular, WP 2.1 and WP 3.1 will switch from learning phase to active file while WP 1.1 will go to a writing phase. The end of this writing phase is marked by the **Milestone 1** with 2 deliverable articles corresponding to WP 1.1.2 and 1.1.3; at least one of these article should be a collaboration with the host team, corresponding with the ongoing WP 2.1.2.

The second secondary milestone (from month 16 to month 18) is a smooth transition with an collaborative article as delivery focused on Objective 2.1 and with early results of WP 3.1.2. **Milestone 2** corresponds to the delivery of two articles. The first is a collaborative article treating WP 1.2.2 and the first advancements over WP 2.2.2. The second is a single-authored article over WP 3.1.2.

The last secondary milestone (from month 27 to month 29) is an other fuzzy transition with a collaborative article as delivery around work packages 1.2.3 and 2.2.2; this milestone will also mark the point where an engineer will be hired to work on the implementation. **Milestone 3** marks the end of the project with the full implementation of WP 3.2 as delivery as well as a collaborative article over WP 2.3.2 together with results from WP 1.3.1.

or  
col.  
with  
Steve  
over  
WP  
1.1

to  
dis-  
cuss

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

## 4.3 Appropriateness of the institutional environment (infrastructure)

UPen and Bhm both have much experience in helping international employees, many of whom supported by Marie Curie grants. Services are devoted to help foreign employees in finding living accommodation as well as dealing with administrative tasks due to mobility (contracts, taxes, etc.). The fellow will therefore receive help not only on his

true?

<sup>26</sup>the intersection-free fragment of

arrival, but for the whole time of his stay, whenever necessary. This will allow him to attend to all administrative aspects of Marie Curie grants, such as preparation of deliverables in a best possible form.

The administrative management will further be supported by the EU Office of Bhm which has been involved in the management of several EU projects which handles the financial and personnel aspects of national and international research projects including EU projects. The Faculty of Science was the recipient of ??? Marie Curie Individual Fellowships and has great experience of project administration, leaving the fellow free to focus on scientific aspects of the project by lightening the administrative task load.

Furthermore, office space and supplies are provided by UPen's department of computer sciences, while research supplies are available through research grants. Each student is guaranteed access to a computer and the Internet from laboratories and from his or her office as well as to systems administration support for computing and data storage. The University Library System holds more than 5,000,000 books and serial volumes in hard copy and digitalized format. Penn strongly encourages students to make use of available university equipment and resources. Although space permits only this brief survey of SEAS and CIS equipment available to postdoctoral fellows, Penn has equipment and student equipment access policies which have been judged exemplary by its peers.

Similarly, Bhm will furnish office space, access to scientific papers from all major publishers, books, and computing facilities.

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

### 5 CV of the Experienced Researcher

#### Personal Informations

Name Flavien BREUVART

Birthday April 6th, 1988

Family Married

Email breuvart@pps.univ-paris-diderot.fr

Website <http://www.pps.univ-paris-diderot.fr/breuvart/>

#### (Future) PhD Thesis

title Could the syntax be the chicken and the semantics the egg?

subtitle From the old style H to the newborn quantitative coefficients.

supervisors Antonio Bucciarelli and Michele Pagani

defense Octobre 23th, 2015

jury Antonio Bucciarelli (Paris Diderot), Pierre-Louis Curien (CNRS), Dan Ghica (University of Birmingham), Giulio Manzonetto (University Paris 13), Guy McCusker (rapporteur, University of Bath), Michele Pagani (Paris Diderot), Simona Ronchi della Rocca (president, University of Turin), Thomas Streicher (rapporteur, Technische Universität Darmstadt)

#### Position

2015-2017: **Iria post-doctoral positions**, in *Focus Team*, Bologna, Italy

2012-2015: **PhD candidate**, with Antonio Bucciarelli and Michele Pagani, at *PPS, University Paris Diderot*, Paris, France

2012-2015: **Teaching Assistant**, at *UFR Informatique, University Paris Diderot*, Paris, France

2008-2012: **Fonctionnaire stagiaire**, at *ENS Cachan*, Cachan, France

#### Other research experience

2011-2012: **12 months internship**, with Antonio Bucciarelli, at *PPS, University Paris Diderot*, Paris, France

2011: **5 months internship**, with Thomas Ehrhard and Antonio Bucciarelli, at *PPS, University Paris Diderot*, Paris, France

2010: **5 months internship**, with Peter Selinger, at *Dalhousie University*, Halifax, Canada

2009: **2 months internship**, with Achim Jung, at *University of Birmingham*, Birmingham, UK

correct  
name?

find  
the  
num-  
ber

## Grants and Awards

- 2015: **Iria post-doctoral positions**, in *Focus Team*, Bologna, Italy
- 2014: **Kleene Award**, for best student paper, in *CSL-LICS*, Vienna, Austria
- 2012: **Three-Years PhD fellowship**, from the french "Ministry of Higher Education and Research", France

## Publications

### Conference papers

- CSL'15: **Breuvart Flavien and Pagani Michele**, *Berlin, Germany*  
"Modelling Coeffects in the Relational Semantics of Linear Logic"
- CSL-LICS'14 **Breuvart Flavien** *Vienna, Austria*  
"On the characterization of models of  $\mathcal{H}^*$ "
- TLCA'13 **Breuvart Flavien** *Eindhoven, Netherlands*  
"The resource lambda calculus is short-sighted in its relational model"

### Submitted

- 2016 **Breuvart Flavien, Marco Gaboardi, Shin-ya Katsumata and Dominic Orchard**,  
"Combining effects and coeffects"
- 2016 **Breuvart Flavien**, long version  
"On the characterization of models of  $\mathcal{H}^*$ "

### non peer-reviewed

- 2012 **Breuvart Flavien**, arXiv preprint arXiv:1205.4691,  
"On the discriminating power of tests in resource lambda-calculus"

## Responsibilities

- 2014: **workshop organization**, Participation at the organization of the international workshop Domain XI, at *PPS, University Paris Diderot*, Paris, France
- 2013: **seminar organization**, Organization of the temporary working group "Sémantique des calculs classiques", at *PPS, University Paris Diderot*, Paris, France
- 2012-2015: **Representative of PhD students**, in the Board of directors, of the laboratory *PPS*, Paris, France

## Teaching

- 2015: **advanced Java**, TA for L3 course named POOIG, at *university Paris Diderot*, Paris, France
- 2014: **advanced Java**, TA for M3 course for linguists named POO3, at *university Paris Diderot*, Paris, France
- 2014 & 2015: **architecture**, TA for M1 course, at *university Paris Diderot*, Paris, France
- 2014: **"UE libre" vulgarization**, In charge of the "UE libre" (optional short course) on scientific vulgarization in M1, at *university Paris Diderot*, Paris, France
- 2013 & 2014: **mid-beginner Java**, TA for L1 course named TO2, at *university Paris Diderot*, Paris, France
- 2012: **Java projects**, Supervising Java projects in L2 (more exactly for the "classe préparatoire intégrée de l'EIDD"), at *university Paris Diderot*, Paris, France
- 2008-2012: **vulgarization**, occasional (5-10 times per years) 1 hour of vulgarization on mathematics and computer sciences for middle school students, with the association *PRACCIS*, Cachan, France
- 2008-2009: **vulgarization of game theory**, 1h/week of vulgarization of game theory to high school students, with the association *Sciences Ouvertes*, Drancy, France

## Communications

- 10/09/2015 **CSL'2015**, Berlin, Germany  
"Modelling Coeffects in Rel"
- 03/09/2014 **seminar of logic team**, of LAMA, *University of Savoie*, Chambéry, France  
"On the characterization of models of  $\mathcal{H}^*$ "

- 17/07/2014 **CSL-LICS**, Vienna, Austria  
“On the characterization of models of  $\mathcal{H}^*$ ”
- 04/04/2014 **seminar of team LCR**, in LIPN, *Univerisity Paris Nort*, Paris, France  
“De la caract risation des mod les de  $\mathcal{H}^*$ ”
- 09/04/2014 **LIAFA-PPS’ PHD seminar**, *University Paris Diderot*, Paris,France  
“Basic notions on weighted models”
- 14/03/2014 **“Theory seminar”**, School of Computer Science, *University of Birmingham*, Birmingham, UK  
“A bridge between semirings”
- 03/12/2013 **BLL’workshop**, Fontainebleau, France  
“A bridge between semirings”
- 27/06/2013 **TLCA**, Eindhoven, Netherlands  
“The resource lambda calculus is short-sighted in its relational model”
- 15/02/2013 **GEOCAL Meeting**, Lyon, France  
“Une bien  trange caract risation de l’ad quation compl te dans les domaines de Scott”
- 24/10/2012 **LIAFA-PPS’ PHD seminar**, *University Paris Diderot*, Paris,France  
“Petites histoires entre mod les et syntaxes”
- 12/09/2012 **PPS’ anual meating**, *University Paris Diderot*, Trouville,France  
“Petites histoires entre mod les et syntaxes”
- 02/09/2011 **LIAFA-PPS’ PHD seminar**, *University Paris Diderot*, Paris,France  
“Et si les PPSiens se mettaient   faire de l’alg bre?!!”

## Education

- 2010-2011: **PHD thesis in Paris Diderot**, *Paris, France*
- 2010-2011: **M2 in MPRI**, *Cachan, France*
- 2009-2010: **M1 in MPRI**, *Cachan, France*
- 2008-2009: **L3 in computer sciences and in mathematics**, *at ENS Cachan, Cachan, France*
- 2008: **admission at ENS Cachan**, *Cachan, France*
- 2006-2008: **CPGE MPSI and MP\***, *at Lyc e F nelon, Paris 5<sup>iem</sup>, France*
- 2006: **French baccalaureate**, “mention bien”, *at Lyc e W.A.Mozart,, Le Blanc Mesnil, France*

## Associative activities

- 2010-2011: **president of PRACCIS project**, in *SONO, Cachan, France*
- 2009-2010: **active member**, of *BDE ENS Cachan, with high implication in the clubs BDA, Mediatek, PRACCIS Cachan,France*
- 2007-2008: **active member** of *Sciences Ouvertes Drancy, France*
- 2005-2007: **board of directors** of the association *MJC Daniel Andr  Drancy, France*

## 6 Capacities of the Participating Organisations

## 7 Ethical Aspects

There are no ethical sensitive issues associated with the subject of this proposal.

## 8 Letters of Commitment of Partner Organisations