Short-Term Scientific Mission Grant
- APPLICATION FORM[1] -

Action number: CA20111

Applicant name: Andreia Mordido

Details of the STSM

Title: Algebraic Session Types

Start and end date: 30/05/2022 to 06/06/2022

Goals of the STSM

Session types enable the specification and verification of communication protocols. After the first proposal of session types, several extensions to this theory have allowed more programs to be typed and more complex message-passing protocols to find a representative type. Context-free session types capture communication patterns recognized by context-free languages and were recently extended with System F in joint work involving the grant applicant and the host of the STSM.

This mission is aimed at boosting research on *algebraic session types*—session types whose expressivity is similar to that of context-free session types and where recursive protocols are modelled as an extension of recursive algebraic datatypes under a nominal rather than structural interpretation of types. By interpreting types nominally, we will avoid the expensive type equivalence algorithms needed for structural type equivalence, reducing the complexity from doubly-exponential to linear time. This complexity decrease is expected to have an impact on the compiler performance. Algebraic session types shall provide new ways of parametrizing protocols.

In this work, we plan to design the theory of algebraic session types, a term language along with its statics and dynamics, as well as results of type soundness, progress and decidability of type equivalence.

Working Plan

Taking advantage of the expressive power of context-free session types, which allow to serialize and deserialize tree-like structures in a type-safe way, we plan to adapt the well-known syntax for datatype definitions to define parametric protocols. In this scenario, one should be able to define a (template) protocol `Tree` α that specifies the behaviour of a channel that sends/receives a tree, without committing to a concrete direction. Such protocol definitions might be recursive and enable the specification of opposite directions of communication within the same

[1] This form is part of the application for a grant to visit a host organisation located in a different country than the country of affiliation. It is submitted to the COST Action MC via e-COST. The Grant Awarding Coordinator coordinates the evaluation on behalf of the Action MC and informs the Grant Holder of the result of the evaluation for issuing the Grant Letter.

protocol. By enabling the definition of parametric protocols, we specify properties of communication protocols as easily as we specify datatypes in functional programming languages.

- Types

  We will start by exploring the syntax of types. The syntactic categories should distinguish data types from types that prescribe communication. The proposed type formation rules should take polarities into account in protocol definitions. Under a nominal interpretation of the type system, we plan to design a type equivalence algorithm, less complex than a similar algorithm for a structural type system with the same expressivity. Results on complexity should be obtained.

- Expressions and processes

  We will define the term language, its statics and dynamics. The operational semantics will rely on a reduction relation either underneath evaluation contexts (e.g. [3]), or under a labelled transition system governed by actions for message exchanges and internal synchronization actions, as used e.g. by Fowler et al. [4]. Results on subject reduction, progress and type safety shall be proved.

- Analysis of expressivity

  We will make a thorough comparison of the proposed type system with other systems whose expressivity captures patterns that live beyond the traces characterized by regular languages, including context-free session types [1,3] and nested session types [2].

- Applications

  Throughout this work, we will keep the focus on applications. We will illustrate the expressivity of types by specifying programs that take advantage of the abstraction of communication protocols through parametric protocol definitions.

The design of the type system has already started in a joint collaboration between Andreia Mordido (the grant applicant), Peter Thiemann (the host) and Vasco T. Vasconcelos. This short-term scientific mission is an excellent opportunity to realize our plans, facilitate the joint work and provide a considerable boost in this research. The collaboration will not cease with the end of the mission. On the contrary, the work plan presented may have to be completed in the weeks after.

References:

[1] Bernardo Almeida, Andreia Mordido, Peter Thiemann and Vasco T. Vasconcelos. “Polymorphic Context-free Session Types”. *arXiv preprint arXiv:2106.06658.* Submitted for publication. 2021.

[2] Ankush Das, Henry DeYoung, Andreia Mordido, Frank Pfenning. “Nested Session Types”. ESOP 2021. 2021.

[3] Peter Thiemann and Vasco T. Vasconcelos. “Context-free session types”. *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*. 2016.

[4] Simon Fowler, Wen Kokke, Ornela Dardha, Sam Lindley and J. Garrett Morris. “Separating Sessions Smoothly”. *arXiv preprint arXiv:2105.08996.* 2021.

Expected outputs and contribution to the Action MoU objectives and deliverables.

(Main expected results and their contribution to the progress towards the Action objectives (either research coordination and/or capacity building objectives) and deliverables.)

The main output of this project is a type system for algebraic session types that enables the specification of communication protocols as easily as the definition of datatypes in functional programming languages. This work will be followed by the implementation of a programming language with the proposed type system, adopting a syntax similar to Haskell. By proposing a nominal type system, we get closer to the usual interpretation of types in

mainstream languages (e.g., Java, C, Haskell), thus facilitating the adoption of the proposed types and programming language in the industry.

The development of type systems for programming languages fosters the correct implementation of programs *by design*, where the compiler ensures the static verification of programs. The more expressive the types, the richer the programs can be. Algebraic session types are aimed to be expressive types that allow the specification of communication protocols whose patterns go beyond regular languages, enabling, for instance, the serialization of tree-structured data (e.g., JSON or XML). In short:

- with more expressive types we enable a larger number of properties to be expressed as types;
- by promoting parametric protocol definitions, we allow these properties to be specified simply as datatypes, which are widely known and used in functional programming languages.

For these reasons, this work aims at contributing to one of the goals of Cost Action: make techniques for program verification more effective and more accessible to all stakeholders.

We also aim at providing a valuable contribution to WG3, namely contributing to the goal of lowering the computational complexity and increasing the expressiveness of techniques for the verification of program correctness. Two of the main contributions of this work consist exactly in (1) reducing the computational complexity underlying the verification of correctness of programs with types whose expressive power goes beyond regular languages, namely by reducing the complexity of type equivalence checking and (2) increasing the accessibility of the proposed types (and programming language) to stakeholders, by extending the well-know syntax for datatypes definition to specify communication protocols.

All in all, we will make sure that this mission will also allow us to address some capacity building objectives, namely: we hope to maintain this collaboration, guaranteeing its extension beyond the lifetime of the action, and we will seek to facilitate access to formal verification techniques for other areas of science.