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

Action number: CA20111

Applicant name: Violet Ka I Pun

Details of the STSM

Title: Static guarantees of confluence in actor languages

Start and end date: 16/04/2023 to 23/04/2023

Goals of the STSM

Purpose and summary of the STSM.

Languages based on the Actor model are known to be data race free because actors (and active objects) encapsulate internal state and restrict local state access to one method at a time, which eliminate such low-level races. However, these systems are prone to high-level communication races which result in a non-deterministic order of execution for methods on an actor in the system.

The goal of this STSM is to investigate guaranteed deterministic behaviour for concurrent actor programs; the approach we take is to identify static properties that can guarantee confluence properties between the execution steps of the language semantics.

This STSM supports an ongoing collaboration between **the applicant** (**Violet Ka I Pun**, Western Norway University of Applied Sciences (HVL), Bergen, Norway), Einar Broch Johnsen (University of Oslo, Norway), and **the host** (**Ludovic Henrio**, LIP laboratory, Univ de Lyon/Inria/CNRS/ENS Lyon) on deterministic behavior for actor languages. In previous work, a simple type system was proposed to restrict the communication topology between objects to guarantee deterministic behaviour [2]. In this STSM, we plan to (1) relax the communication topology using linearity and an ownership type system using, e.g., borrowing, and (2) adopt De Bruijn's proof framework for confluence [1] to show runtime confluence properties. We also intend prove properties in this setting using the proof assistant Coq.

Working Plan

Description of the work to be carried out by the applicant.

In this STSM, we plan to extend the earlier work by the applicant and the host on guaranteeing deterministic behaviour of actor (and active object) programs [2] to the much more general concept of confluence properties in the operational semantics and static guarantees for actor and active object languages that can guarantee such confluence properties. We approach this problem from three perspectives: (1) relaxing the communication topology between objects of our previous work; (2) using

[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.

De Bruijn's proof framework for confluence for families of rewrite relations in abstract reduction systems [1] to show the confluence properties in the operational semantics of actor languages; and (3) proving the properties using Coq.

The earlier work in [2] uses a simple type system to enforce the communication topology as a tree structure for objects. Such a tree structure is rather static and restrictive. In the aspect, we plan to relax structure by using e.g., linearity and ownership type systems with borrowing to introduce more dynamic trees [3], and add primitive to attach and detach tree to the object dependence graph, in order to constantly ensure a tree structure, but allow the structure of the tree to evolve at runtime.

In De Bruijn's proof framework [1], the reduction relations for abstraction reduction systems are indexed by a well-founded total order. The proof of the weak diamond properties is shown by induction on this order. To adopt De Bruijn's proof framework to show confluence properties for actor languages, we structure the semantics of the core actor language into different levels. Naïvely, we first divide the reduction relation into two levels: one level for the deterministic kernel where the reductions can be reordered, e.g., reductions by stateless objects can commute, while reductions in the other level maintain the tree structure introduced in [2]. We can than show the confluence properties for the active object languages with the operational semantics organised in these two levels. Afterwards, we plan to further explore other conditions under which we can divide the latter level in order to identify more commuting reduction patterns.

In addition to leveraging De Bruijn's proof framework to show confluence properties for actor languages, we plan to formalise the proof of such properties using the proof assistant Coq, by working closely with Yannick Zakowski, Henrio's colleague at LIP laboratory.

Einar Broch Johnsen (University of Oslo, Norway) is similarly applying for an STSM from EuroProofNet for this work.

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 proposed research is relevant to Working Group 3 on Program verification and Working Group 6 on Type theory.

One expected result of the STSM is an ownership type system using, e.g., permissions and borrowing to allow a dynamic structure of actors that ensures deterministic behaviour, i.e., confluence properties, of actor languages.

Another outcome would be an illustration of how operational semantics of actor languages can be restructured to prove confluence properties using De Bruijn's framework, which facilitates formal verification of actor programs.

Although it is possibly unrealistic to have the confluence proofs for the actor semantics completely formalised in Coq during this STSM, we expect this will be the eventual result.

[1] Endrullis, J. and Klop, J. W.: De Bruijn's Weak Diamond Property Revisited. Indagationes Mathematicae, 24(4):1050 – 1072, 2013. In memory of N.G. (Dick) de Bruijn (1918–2012). https://doi.org/10.1016/j.indag.2013.08.005

[2] Henrio, L., Johnsen, E.B., Pun, V.K.I: Active Objects with Deterministic Behaviour. In: Dongol, B., Troubitsyna, E. (eds.) IFM 2020. LNCS, vol. 12546. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63461-2_10

[3] Haller, P., Odersky, M.: Capabilities for uniqueness and borrowing. In: D'Hondt, T. (ed.) ECOOP 2010. LNCS, vol. 6183, pp. 354–378. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14107-2_17