# Report STSM

## Glasgow 04/09/2022 – 18/09/2022

During the STSM I have been working on the Agda mechanization of both a type inference algorithm for the pi-calculus and a library for unification that can be used in different type inference scenarios.

The basis of the STSM was the work I previously started during the STSM on May, 25th. There, I started mechanizing the completeness theorem for the type inference algorithm for the shared pi-calculus. Clearly such problem reduced to prove the completeness of the unification algorithm. After the STSM in May I managed to conclude the unification and I decided to deal with the research topic as follows:

1. Use the completeness of unification to prove the completeness of inference in the shared pi-calculus
2. Generalize the unification to build a library that can be easily instantiated to other contexts, eg the linear pi-calculus
3. Provide an inference algorithm for the linear pi-calculus as well as its soundness and completeness proofs

I spent the first week dealing with 1. Indeed, as it often happens, the Agda proof was non trivial. Technically speaking, the proof technique was easy to find out (induction on the well typedness proof) but from the formalization point of view it involved many equalities (Agda equality type). When we use induction, we usually end up using some inductive constructor of some datatype. This is not the case when one has to deal with equalities. In fact, the results have to be properly combined using the equality properties which is often hard to reason about, especially when functions are involved. Moreover, the completeness proof can be split in two different sub-theorems:

1. Given a well typedness proof, the inference algorithm must successfully terminate
2. Some property of the output of the inference algorithm must be proved

At the end of the week I managed to conclude the main cases of the proof while the others can be obtained straightforwardly. During this activity I also integrated some results in the unification part that were missing.

During the second week I started working on the linear pi-calculus. First, it needed a generalized unification library that takes into account the resources multiplicities as well. While the actual development is based on a such general library, I made some trials to reformulate the main definitions to fit the inputs of the previous sound/complete library. However, I put this problem on hold in order to focus on inference which was the hardest part of the research. Differently from the shared case, the algorithm returns a set of constraints due to the addition of multiplicities. The soundness theorem should take care of such constraints by trying to solve them; if it manages to do so then a well typedness proof can be provided. The solution of the constraints hides a challenge: it might happen that a solution of a single constraint makes the others that depend on that one unsolvable. For this reason, constraints and their solution algorithm must be defined in a proper way.

**Results**

1. *Sound* and *Complete* unification library based on a single set of terms
2. *Sound* and *Complete* inference algorithm for the shared pi-calculus
3. *Sound* unification library based on an arbitrary syntax

4. *Inference algorithm* for the linear pi-calculus

The actual development can be checked here https://github.com/LcicC/CoContextualPi

**Open Problems**

1. Is the library used in the shared case general enough to be used in the linear case as well?
    a. YES – Nothing left to do
    b. NO – Write the completeness proof in the more general case and use it for both the shared and the linear pi-calculus
2. Can constraints be properly defined to avoid the need of backtracking?
    a. Constraints are constructors of terms as well (actual development)
3. Provide constraint solving algorithm for the linear pi-calculus
4. Prove soundness and completeness of the inference in the linear pi-calculus

## Conclusions

The research project that I started with Ornela Dardha in May has now clear objectives. We aim at publishing the results in a formalization-specific conference like ITP or CPP. The general idea is to consider the unification library as the main contribution and present shared/linear pi-calculus as use cases. We are also interested in applying the obtained results in other contexts live GV.

## Personal Considerations

I planned another visit to the University of Glasgow in the end of October to continue the present work. This project and these two visits conclude my PhD as my contract ends November 1st. As many unlucky students, I had the chance to personally meet colleagues from other universities only in 2022. Beside a standard acknowledgment in a possible future publication, I would like to thank EuroProofNet for giving me the possibilities to integrate the funds that the university provided me in order take advance as much as possible of the research collaboration with Ornela Dardha's research group.

Glasgow, 26th September 2022

Ornela Dardha