UNIVERSITÀ DEGLI STUDI DI TORINO

World Wide Style 2 Program

Duration	Scientific area	Scientific responsible	Host Department	Type of activity	Start/End of mobility	Language
6 months	Theoretical Computer Science	Simona Ronchi della Rocca	Dipartamento di Informatica	Research/ Teaching	December 2015 June 2016	English

Description of the research project

Quantum computing, from a purely algorithmic point of view, set up a new model of computation. This model incorporates new ways of thinking the algorithms. Key concepts for it are the *quantum entanglement*: the property for which the state of some multi-parties systems cannot be described independently and the *superposition principle*: the ability of a system to be in more than one state at once. These concepts are unequalled to any classical one, although there are some common points with probabilistic models, as well as non-deterministic ones.

This theoretical computer science branch has its origin in physics; more precisely in the physicist Richard Feynman, who in 1981 gave a seminar in the Massachusetts Institute of Technology (MIT) about the problem of simulation of quantum physics by classical computers. Far from proposing any solution, he opened the doors for new questions never asked before. What gain could be achieved if the computers were governed by the quantum mechanics laws?. The answer did not come until the propositions by Grover [1996] and Shor [1997]. The former is an algorithm to perform a search over an unordered register, with a quadratic gain in temporal complexity with respect to any known classical algorithm. The latter is an algorithm for fast factorisation, with an even more impressive exponential gain. These results generated a great interest from computer scientist from all the fields. From a more fundamental perspective, the logical foundations behind quantum computing, remain still a mystery. Although there exists a quantum logic [Birkhoff and VonNeumann 1936], it has been proposed much before quantum computing, and so their relation is hard to find.

Within the classical models for computing, the λ-calculus [Church 1936] highlights. It can be seen as the most simple and universal programming language, yet the very concept of computability can be defined in terms of λ-calculus. Moreover, the Curry-Howard correspondence establishes a direct relation between typed λ-calculus and proofs in constructive mathematics. Pursuing this correspondence, for quantum computing, has been the main motivation of the field of quantum programming languages.

In particular, Lineal [Arrighi and Dowek 2008] is an extension of λ-calculus aiming to naturally express quantum programs. Whilst partly unexplained, it is nevertheless clear that the algorithmic speed-up of quantum computation arises by tapping into the parallelism granted to us 'for free' by the *superposition principle*; which establishes that some linear combination of the possible states of a system, are also states. The idea of vector space of λ-terms over an arbitrary scalar field arises quite naturally in this context. Such was the motivation behind Lineal.

The Vectorial calculus λ^{vec} is the typed version of Lineal [Arrighi, Diaz-Caro and Valiron 2013]. This calculus introduces a type system where the types, in the same way as the terms, form a vectorial space, being able to type quantum programs, and providing information about the vectorial structure of the normal form of the terms

In particular, λ^{vec} develops on the following idea: In lambda calculus both programs

		and data are values, hence, since the data on quantum computing (the qubits) are vectors, the programs in a quantum lambda-calculus must be vectors too. Our main aim in this project is to develop a new type system where the types characterises the superpositions. We divide the project in the following goals: **Goal:** *Quantum language with intersection and union types*. Intersection and union types are being studied for three decades. The main idea, naturally taken from sets, is that if a proof can prove two different propositions, it can also prove its intersection. And if a proof can prove one proposition, it is also a proof of this proposition in union with a second one. The path to follow is to consider a language where the intersection captures the concept of superposition while the union captures the concept of mixed state, that is, a partial state, part of a composed state in entanglement, which cannot be described by a single vector. **Goal:** *Identifying isomorphisms in intersection and union types*. In [Díaz-Caro and Dowek 2015] the authors introduced λ_+, a non-deterministic λ-calculus, in which the non-determinism arises naturally by considering the isomorphisms between types to be equalities. In fact, already in the study of λ^{vec} some isomorphisms are considered as equalities (in particular, those related to the associativity and commutativity of the addition in a vectorial space). In λ_+ they went further and took the equality to all the isomorphisms of simply typed lambda calculus. The resulting calculus includes a non-deterministic projector, which in a more complex theory with quantum superpositions can be considered as a measurement operator. If a quantum calculus with intersection and union types will be developed, one interesting aspect is to identify its isomorphisms, in order to, in a next step, extend such a language with a rewrite system modulo these isomorphisms. The first results in this direction have already been published [Coppo, Dezani, Margaria and Zacchi 2013, 2014, 2014b]. The plan is to continue building upon these results.
	Profile Description	The ideal candidate should have a strong background in the design of type systems for different porpoises (quantum computing, pattern-calculus, etc). To have experience on quantum computing is a plus, but not mandatory, since the goal two does not require, a priori, any knowledge on this area. The experience with intersection and/or union types is highly appreciated.
	Research objectives	At UniTo there are some of the most renewed researchers in Intersection and Union type, hence the candidate will combine his experience in the design of type systems with our experience in the proposed setting.
	Teaching activities (if applicable)	In case the recruited fellow has a strong background in quantum programming languages, we propose he teach a course on “Fundaments of quantum programming languages”. In case the recruited fellow has a strong background in any other specify calculus (such as pattern-calculus), a course on it will be demanded.
	Website and Contact	**Simona Ronchi della Rocca** Dipartimento di Informatica, Università di Torino Corso Svizzera 185, I-10149 Torino, Italy Phone: +39 011 6706734 Email: ronchi@di.unito.it – Website: http://www.di.unito.it/~ronchi