LOG1 - LOGIC AND COMPUTABILITY

Organizers
Mauricio Guillermo, Universidad de la República, Uruguay
Martin Hyland, University of Cambridge, United Kingdom

LOG1-1 **Alejandro Díaz-Caro**, Universidad Nacional de Quilmes, Argentina

Towards a quantum lambda calculus with quantum control

There are two main trends on the study of functional quantum programming languages: On one hand, a well developed line follows the scheme of quantum-data/classical-control. That is a model where the actual quantum computation runs in a quantum memory while the program controlling which operations to apply and when, runs in a classical computer. This scheme counts with a recent semantical study for higher-order quantum computation, as well as several prototypes such as QML or the more scalable and recent Quipper. On the other hand, there is the scheme of quantum data and control. Its origins can be traced back to the linear-algebraic lambda calculus (Lineal) and its multiple type systems. This model, while less suitable to produce a scalable quantum programming language nowadays, may give better insights on the quantum properties and the quantum operations.

The work I will present is inscribed in this second line. We propose an extension of lambda calculus to handle some properties of quantum computing. The starting point is to consider the quantum superposition as a commutative pair and the quantum measurement as a non-deterministic projection over it. Destructive interferences are achieved by introducing an inverse symbol with respect to pairs. The no-cloning property is ensured by using a combination of syntactic linearity, which is enough for unitary gates, with linear logic, to forbid measure functions from duplicating their arguments.

LOG1-2 **Peter Dybjer**, Universidad de Chalmers, Sweden

Game Semantics and Normalization by Evaluation

Game semantics and normalization by evaluation have both been active fields of research since the 1990s. In game semantics the computation of a program is viewed as the actions of a player playing a game against the enviroment. Among other things game semantics has given rise to a solution to the long standing full abstraction problem in semantics. Normalization by evaluation on the other hand is a technique for computing normal forms in lambda calculi by interpreting terms in a model and then "reifying" the semantic value to a normal form. We shall here show a new way to present Hyland and Ong's game semantics for PCF by using normalization by evaluation (nbe). We use the bijective correspondence between innocent well-bracketed strategies and PCF Bohm trees, and show how operations on PCF Bohm trees, such as composition, can be computed lazily and simply by nbe. The usual equations characteristic of games follow from the nbe construction without reference to low-level game-theoretic machinery. As an illustration, we give a Haskell program computing the application of innocent strategies.

Joint work with Pierre Clairambault, CNRS, ENS Lyon.

LOG1-3 **Walter Ferrer Santos**, Universidad de la República, Uruguay

Ordered Combinatory Algebras and Realizability

We propose the new concept of Krivine ordered combinatory algebra (KOCA) as foundation for the categorical study of Krivine's classical realizability, as initiated by Streicher (Streicher (2013)). We show that KOCA's are equivalent to Streicher's abstract Krivine structures for the purpose of modeling higher-order logic, in the precise sense that they give rise to the same class of triposes. The difference between the two representations is that the elements of a KOCA play both the role of truth values and realizers, whereas truth values are sets of realizers in AKSs.

Joint work with: Jonas Frey, Mauricio Guillermo
Octavio Malherbe and Alexandre Miquel

LOG1-4 **Santiago Figueira**, Universidad de Buenos Aires, Argentina

Model Theory of XPath with data tests

We present recent results on the model theory of XPath —arguably the most widely used XML query language— with data (in)equality tests over the class of data trees. These are trees where each node contains a label from a finite alphabet and a data value from an infinite domain. We provide notions of bisimulations for some fragments of XPath with data tests and we show that these notions precisely capture the logical equivalence relation associated with each fragment. We show two results where the tool of bisimulation plays a central role: characterization and definability. Bisimulations work both for data trees or data graphs. While the problem of computing the largest bisimulation over the class of finite data trees is PTIME, it becomes more difficult when finite data graphs are considered: we show that in general the problem is PSPACE-complete, but identify several restrictions that yield better complexity bounds. Finally, we give a sound and complete axiomatization for a simple fragment of XPath with data tests.

LOG1-5 **Jonas Frey**, University of Copenhagen, Denmark

Classical realizability and implicit computational complexity

Classical realizability was introduced in the early 2000s by J.-L. Krivine as a model construction for ZF set theory extending Cohen's forcing method, and has since been studied from different angles by researchers in logic, computer science, and category theory.

I will present an approach to use classical realizability as a tool in implicit complexity theory, based on joint ongoing work with J. Grue Simonsen. Specifically, we associate classical realizability models to computational complexity classes, and study the ensuing models using tools of topos theory.

The topos theoretic approach allows to view these models as geometric structures, broadly speaking supporting the intuition that complexity classes can be regarded as generalized topologies on the set of integers.

LOG1-6 **Stéphane Graham-Lengrand**, Centre National de la Recherche Scientifique, France

A proof-theoretical approach to satisfiability solving

Proof theory has been very successful at supporting the development of software for proof-checking, and of tableaux-based software for automated reasoning.

It is more rarely used as a source of inspiration for other vast areas of automated reasoning, such as SAT- and SAT-Modulo-Theories solving: these have rather developed their own model-based techniques for designing and combining decision procedures for quantifier-free problems in specific theories (and adding to them some support for quantifiers).

We provide here a proof-theoretic description of the core mechanism of SAT- and SMT-solvers, which usually implement variants of the DPLL procedure or, more precisely, the CDCL procedure (Conflict-Driven Clause Learning). This mechanism alternates model construction steps and conflict analysis steps, which correspond to top-down and bottom-up proof construction steps.

Through that lense, we promote for satisfiability-solving the proof-theoretical idea of identifying core inference possibilities that are distinct from the strategies used to apply them, and describe how Milner's LCF approach for guaranteeing correctness in theorem proving can be lifted to the field of SMT-solving. We show how it can be generalised to guarantee (both "provable" and "non-provable") answers in a prover architecture where numerous components (e.g. for different theories) interact and collaborate, some of which are trusted and some of which are not.

Finally, we discuss how this proof-theoretical approach to automated reasoning could open a safe way to exploit distributed implementation, and possibly machine learning.

LOG1-7 **Alexandre Miquel**, Universidad de la República, Uruguay

Implicative algebras for generalizing forcing

In this talk, we present work in progress in the direction of merging the theory of forcing (in the sense of Cohen) with the theory of classical realizability (in the sense of Krivine).

For that, we introduce the notion of implicative algebra, a very simple algebraic structure that subsumes both complete Heyting (or Boolean) algebras and classical realizability structures such as introduced by Krivine.

We show in particular how this structure (that is organized around the order of subtyping) allows us to lift the operations of the lambda-calculus to the level of truth values, so that they can be manipulated as generalized realizers.

In the second part of the talk, we explain how implicative algebras can be used to factorize the construction of Boolean-valued models of ZF (equivalent to forcing) and the construction of classical realizability models of ZF. We conclude by presenting the open problems that are naturally raised by this approach.

LOG1-8 **Antonio Montalbán**, University of California at Berkeley, USA

Natural Objects in Computability Theory

Here is an interesting phenomena that occurs in computability theory and other parts of logic: Some natural class of objects is defined but the class turns out to be badly behaved. However, when one restricts oneself to the "natural" objects within the class, the class is very well-behaved. We will talk about this phenomena in general, and show our results for the case of many-one degrees.