An Allegorical Account of Operational Semantics

Francesco Gavazzo
University of Padua

Operational semantics is one of the most successful approaches for defining and reasoning about the computational behaviour of syntax, whether of phrases of ordinary programming languages, proof assistants, or even formal calculi behind them—*e.g.* modern type theories. In all these cases, n operational semantics allows one to model many behaviours, such as type inhabitation or program evaluation, *computationally*, often providing more effective alternatives to denotational and logic-based approaches.

State-of-the-art operational semantics are inherently *proof-theoretic*, providing formal definitions of relations on the (abstract) syntactic phrases of the language through *formal deductive systems*. Typical examples involve judgments of the form "in this context, this phrase has this type" or "in this context, this phrase evaluates to this value." While this proof-theoretic viewpoint makes operational reasoning practical, effective, and well-suited for (assisted) mechanization by machines, it also makes operational techniques and specifications inherently syntax-specific. As a result, operational definitions and proofs often need to be repeated from scratch whenever a new language or syntactic feature is introduced.

To mitigate this drawback, *category theory* has been strongly advocated as a *lingua franca* to build a unifying perspective on operational methods. In response, categorical notions and results have been successfully applied to develop general (algebraic) theories of syntax—through monads, initial algebras, combinatorial species, PROPs, etc.—as well as coalgebraic theories of computational behaviors—think about categorical generalisations of transition systems and rewriting. The deductive systems used in operational specifications can then be understood as ways to relate syntax and behaviours, and formalised as certain *morphisms* between algebras and coalgebras, as well as in terms of *categorical logic*. Unfortunately, despite their elegance and conceptual appeal, *none* of the available categorical accounts of operational semantics are used in practice, primarily because the categorical structures employed are too abstract to be directly related to their proof-theoretic counterparts.

In this talk, we explore a different perspective on "abstract operational semantics", one rooted in the concept of a *syntax relation* and of its formalisation in terms of *allegory*—rather than category—*theory*. Accordingly, rather than viewing deductive systems and their inference rules as the keystone of an operational specification, we focus on what these rules define, certain relations on syntax dubbed *syntax relations*. Following this perspective, we investigate the mathematical structure of syntax relations, showing how they naturally emerge from canonical relational extensions of the very categorical constructions modelling syntax. without any need to introduce any further structure, such as coalgebras, distributive laws, *etc.* These results open the door to a more radical relational approach, where operational notions are defined natively as certain constructions on suitable allegories, independently on any underlying model of syntax. Remarkably, this allegorical approach is not only highly expressive—it encompasses first- and second-order syntax, both typed (simple, polymorphic, and dependent types) and untyped, as well diagrammatic syntax—but also practically usable: in fact, all the allegorical constructions used to model operational notions are characterised by simple (quasi-)equational laws that make operational reasoning essentially *calculational*.