Construction of Transitory Nets

Roberto Maieli

Dipartimento di Matematica e Fisica - Universitá "Roma TRE"
roberto.maieli@uniroma3.it

We present some recent developments of a research programme which points to a theoretical foundation of a *computational programming paradigm based on the construction of proofs of linear logic* (Jean-Yves Girard, 1987). Naively, this paradigm relies on the following isomorphism: "proof = state" and "construction (or inference) step = state transition". Traditionally, this paradigm is presented as an incremental (bottom-up) construction of possibly *incomplete* (i.e., open or with proper axioms) proofs of the *bipolar focusing sequent calculus* This calculus satisfies the property that the complete (i.e., closed or with logical axioms) bipolar focusing proofs are fully representative of all the closed proofs of linear logic: this correspondence is, in general, not satisfied by the polarized fragments of linear logic. Bipolarity and focusing properties ensure more compact proofs since they get rid of some irrelevant intermediate steps during the proof search (or proof construction).

Now, while the view of sequent proof construction is well adapted to theorem proving, it is inadequate when we want to model some proof-theoretic intuitions behind, e.g., concurrent logic programming which requires very flexible and modular approaches. Due to their artificial sequential nature, sequent proofs are difficult to cut into modular (reusable) concurrent components. A much more appealing solution consists of using the technology offered by *proof nets* of linear logic or, more precisely, some forms of de-sequentialized (geometrical, indeed) proof structures in which the composition operation is simply given by (possibly, constrained) juxtaposition, obeying some correctness criteria. Actually, the proof net construction, as well the proof net cut reduction, can be performed in parallel (concurrently), but despite the cut reduction, there may not exist executable (i.e., sequentializable) construction steps: in other words, construction steps must satisfy a, possibly efficient, correctness criterion. A proof net is a particular "open" proof structure, called *transitory net*, that is incrementally built bottom-up by juxtaposing, via construction steps, simple proof structures or modules, called *bipoles*. Roughly, bipoles correspond to Prolog-like *methods* of Logic Programming Languages: the *head* is represented by a multiple trigger (i.e., a multiset of positive atoms) and the *body* is represented by a layer of negative connectives with negative atoms. We say that a construction step is correct (that is, a *transaction*) when it preserves, after the juxtaposition, the property of being a transitory net: that is the case when the abstract transitory structures *retracts* (by means of a finite sequence of rewriting steps) to elementary collapsed graphs (i.e., single nodes with only pending edges). Each retraction step consists of a simple (local) graph deformation or graph rewriting. The resulting rewriting system is shown to be convergent (i.e., terminating and confluent), moreover, it preserves, step by step, the property of being a transitory structure. Transitory nets (i.e., retractile structures) correspond to derivations of the focusing bipolar sequent calculus.

The first retraction algorithm for checking correctness of the proof structures of the pure multiplicative fragment of linear logic (MLL), was given by Vincent Danos in 1990; the complexity of this algorithm was later shown to be linear, in the size of the given proof structure, by Stefano Guerrini in 1999. Then, the retraction criterion was extended, respectively, by the author, in 2007, to the pure multiplicative and additive (MALL) proof nets with *boolean weights* and then by Christophe Fouqueré and Virgil Mogbil, in 2008, to polarized multiplicative and exponential proof structures.

Traditionally, concerning proof nets of linear logic, the main interest on the retraction system is oriented to study the complexity of correctness criteria or cut reduction. Here, our (original) point of view is rather to exploit retraction systems for incrementally building (correct) proof structures. Indeed, the convergence of our retraction system allows to focus on particular retraction strategies that turn out to be optimal (better, parsimonious) w.r.t. the problem of incrementally constructing transitory nets. Actually, checking correctness of an expanded proof structure is a task which may involve visiting (i.e., retracting) a large portion of the so obtained net: some good bound for these task would be welcome. Here, we show that checking correctness (retraction) of a transitory net, after a construction attempt, is a task that can be performed by restricting to some "minimal" (i.e., already partially retracted) transitory nets. The reason of that is that some subgraphs of the given transitory net will not play an active role in the construction process, since they are already correct and encapsulated (i.e., border free): so, their retraction can be performed regardless of the construction process.