Rewriting and call-time choice

Juan Rodríguez Hortalá

12/05/2008

Abstract

Non-confluent and non-terminating rewrite systems are interesting from the point of view of programming. In particular, existing functional logic languages use such kind of rewrite systems to define possibly non-strict non-deterministic functions. The semantics adopted for non-determinism is call-time choice, whose combination with non-strictness is not a trivial issue that has been addressed from a semantic point of view in the Constructor-based Rewriting Logic (*CRWL*) framework.

In a recent serie of works we have investigated how to express call-time choice and non-strict semantics from a point of view closer to classical rewriting. We propose a notion of rewriting which uses an explicit representation for sharing with *let*-constructions and is proved to be equivalent to the *CRWL* approach.

During this talk we will address the following issues:

- We will briefly review some of the main existing approaches for describing the semantics of functional logic languages: *CRWL*, traditional rewriting and Curry's *à la* Launchbury operational semantics.
- We will present the *let*-rewriting relation for first order functional logic programs, and its semantic equivalence with *CRWL* logic.
- We will discuss the relation between all these semantic approaches, and present some technicall results concerning this relations.
- We will see some interesting properties of the combined framework *CRWL*/*let*-rewriting, and show by a case study (correctness of bubbling) the usefulness of the achieved combination of semantic and reduction notions.
- We will extend our *let*-rewriting relation providing a notion of *let*-narrowing which is adequate for call-time choice as proved by soundness and completeness results of *let*-narrowing with respect to *let*-rewriting. Completeness is based on a lifting lemma for *let*-rewriting similar to Hullot's lifting lemma for ordinary rewriting and narrowing.