A proof with side effects of Gödel's completeness theorem

Hugo Herbelin

As revealed by T. Griffin in 1990, classical reasoning is related to control operators along the proof-as-program correspondence. Classical reasoning can be simulated within intuitionistic logic via a negative translation, the same way that programming with control operators can be simulated in pure λ-calculus by reasoning in so-called continuation-passing style. Under this view, classical logic is *direct style* for proving intuitionistically within the target of a negative translation.

We shall consider a monotonic memory update effect and connect it to Kripke's forcing, seen as a dependent variant of an environment-passing style. We shall then interpret reasoning within the target of Kripke's forcing as indirect style for reasoning with a monotonic memory update effect (by monotonic update is meant that the memory can be updated only with a value which refines the previous value according to a given refinement order).

As an application, we shall consider the logical completeness of classical logic with respect to two-valued models and logical completeness of intuitionistic logic with respect to Kripke models. We shall interpret the former as a direct-style formulation of the latter using monotonic memory update. This provides with a proof with side effect of Gödel's completeness theorem whose computational content, when interpreted in a constructive meta-logic, is, like it is the case in intuitionistic logic, basically replicating the structure of a proof of classical validity into a proof of classical provability.

The first lecture will be about the proof-as-program correspondence for classical logic with an application to computing intuitionistically with Markov's principle.

The second lecture will be about the computational content of Henkin's proof of Gödel's completeness theorem.

The third lecture will be about proving Gödel's completeness theorem with side effects, properly speaking.