Dialogues, Proofs and Programs

Tadeusz Litak (Informatik 8, FAU Erlangen-Nürnberg)

How to understand logical connectives? Which sentences are tautologies? And, actually, what is logic all about? There are several ways to answer these questions other than good old truth tables. One can tie logic to analysis of *argumentation* and *dialogue strategies*. On this view, logical tautologies are those sentences which the Prover can defend against any attack from the Skeptic. Shifting the focus just a little bit, one can concentrate on *judgements*, *evidence* and *proofs*, designing a *natural deduction* system and defining each connective in terms of suitably matching *introduction* and *elimination* rules. One can also go about it in different ways, like the fashionably *computational* approach (which identifies logical connectives with operations on types in programming languages and blurs the distinction between proofs and programs) or via the workings of logical connectives in *categories* and *toposes* ... but we needn't go there in this tutorial.

Curiously, at least on the propositional level, all these approaches are consistent with each other, and allow to defend against skeptics many formulas we learned to venerate as tautologies in basic logic courses. Curiouser and curiouser, though, not quite all the classical tautologies have the same status. Justifying laws like the Excluded Middle, the Double Negation, the Peirce Syllogism and their relatives would require significant tweaks to each of these perspectives (like enforcing full symmetry between the Prover and the Skeptic in dialogues or putting *sequent systems* rather than natural deduction at the heart of philosophically inclined proof theory). Those laws that do *not* require these changes and are valid under *any* of these interpretations are known to the world as the *Intuitionistic Propositional Calculus* (IPC).

We will try to understand slightly better this formalism and its amazing journey from the esoteric Kantian roots to the roots of modern proof assistants, tools for formalization of mathematics and program verification. We will see some useful semantics employed to investigate the IPC as a formal system, in particular a suitable variant of the Kripke semantics of possible worlds. If time allows (which looks like a pretty big counterfactual), we may also have a look at various closer and distant relatives of the IPC.