A first-order inquisitive witness semantics

Jeroen Groenendijk & Floris Roelofsen

ILLC/University of Amsterdam

Abstract

A central notion in inquisitive semantics is that of *support*. A common way to formulate the semantics is to start with a recursive definition of when a state supports a sentence, and then define the proposition expressed by a sentence as the set of all (maximal) states supporting that sentence. This approach is similar to the one that is usually taken in classical logic. There we start with a recursive definition of truth, and then define the proposition expressed by a sentence as the set of all worlds where the sentence is true. Thus, the role of support in inquisitive semantics is comparable to that of truth in classical logic.

However, Ivano Ciardelli has shown that there are certain sentences in the language of first-order logic, the so-called *boundedness formulas*, which are equivalent in terms of support, even though, intuitively, they invite different types of responses. Ciardelli concludes from this observation that a support-based inquisitive semantics is not fine-grained enough in the first-order setting.

In the semantics we will propose in this paper, states do not only contain information, but also a set of *witnesses*. The main feature of the semantics, then, is that an existentially quantified sentence like $\exists x.Px$ is only supported in a state if there is a specific witness in that state which is known to have the property P. As a result, an inquisitive sentence may not only request a response that provides certain information, but also a response that introduces a certain witness. Thus, the notion of inquisitiveness is richer than in the basic first-order system. Because the notion of inquisitiveness is enriched in this way, the semantics is able to make more fine-grained distinctions. In particular, it suitably assigns different semantic values to the boundedness formulas. At the same time, unlike the semantics that Ciardelli proposed to avoid the boundedness problem, the semantics developed here is still support-based.