First-order logic has a number of nice properties. We know it is not decidable, but at least it is axiomatizable. That is, there are proof systems for first-order logic which are sound and complete, i.e., they give rise to a derivability relation $\vdash$ with the property that for any set of sentences $\Gamma$ and sentence $Q$, $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$. This means in particular that the validities of first-order logic are computably enumerable. There is a computable function $f : \mathbb{N} \to \text{Sent}(\mathcal{L})$ such that the values of $f$ are all and only the valid sentences of $\mathcal{L}$. This is so because derivations can be enumerated, and those that derive a single sentence are then mapped to that sentence. Second-order logic is more expressive than first-order logic, and so it is in general more complicated to capture its validities. In fact, we’ll show that second-order logic is not only undecidable, but its validities are not even computably enumerable. This means there can be no sound and complete proof system for second-order logic (although sound, but incomplete proof systems are available and in fact are important objects of research).

First-order logic also has two more properties: it is compact (if every finite subset of a set $\Gamma$ of sentences is satisfiable, $\Gamma$ itself is satisfiable) and the Löwenheim–Skolem Theorem holds for it (if $\Gamma$ has an infinite model it has a denumerable model). Both of these results fail for second-order logic. Again, the reason is that second-order logic can express facts about the size of domains that first-order logic cannot.

Photo Credits

Bibliography