Logics commonly have both a semantics and a derivation system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of derivation systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a derivation in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of sentences or formulas. Good derivation systems have the property that any given sequence or arrangement of sentences or formulas can be verified mechanically to be “correct.”

The simplest (and historically first) derivation systems for first-order logic were axiomatic. A sequence of formulas counts as a derivation in such a system if each individual formula in it is either among a fixed set of “axioms” or follows from formulas coming before it in the sequence by one of a fixed number of “inference rules”—and it can be mechanically verified if a formula is an axiom and whether it follows correctly from other formulas by one of the inference rules. Axiomatic derivation systems are easy to describe—and also easy to handle meta-theoretically—but derivations in them are hard to read and understand, and are also hard to produce.

Other derivation systems have been developed with the aim of making it easier to construct derivations or easier to understand derivations once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some derivation systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its derivations are essentially impossible to understand). Most of these other derivation systems represent derivations as trees of formulas rather than sequences. This makes it easier to see which parts of a derivation depend on which other parts.

So for a given logic, such as first-order logic, the different derivation systems will give different explications of what it is for a sentence to be a theorem and what it means for a sentence to be derivable from some others. However that is done (via axiomatic derivations, natural deductions, sequent derivations, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let’s write \( \vdash \varphi \) for “\( \varphi \) is a theorem” and “\( \Gamma \vdash \varphi \)” for “\( \varphi \) is derivable from \( \Gamma \).” However \( \vdash \) is defined, we want it to match up with \( \models \), that is:

1. \( \vdash \varphi \) if and only if \( \models \varphi \)

2. \( \Gamma \vdash \varphi \) if and only if \( \Gamma \models \varphi \)

The “only if” direction of the above is called soundness. A derivation system is sound if derivability guarantees entailment (or validity). Every decent derivation system has to be sound; unsound derivation systems are not useful at all. After all, the entire purpose of a derivation is to provide a syntactic guarantee of validity or entailment. We’ll prove soundness for the derivation systems we present.
The converse “if” direction is also important: it is called completeness. A complete derivation system is strong enough to show that \( \varphi \) is a theorem whenever \( \varphi \) is valid, and that \( \Gamma \vdash \varphi \) whenever \( \Gamma \models \varphi \). Completeness is harder to establish, and some logics have no complete derivation systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a derivation system of first-order logic in his 1929 dissertation.

Another concept that is connected to derivation systems is that of consistency. A set of sentences is called inconsistent if anything whatsoever can be derived from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiability: like unsatisfiable sets, inconsistent sets of sentences do not make good theories, they are defective in a fundamental way. Consistent sets of sentences may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different derivation systems the specific definition of consistency of sets of sentences might differ, but like \( \vdash \), we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that \( \Gamma \) is consistent if and only if it is satisfiable. Here, the “if” direction amounts to completeness (consistency guarantees satisfiability), and the “only if” direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

Photo Credits

Bibliography