

syn.1 Languages and Connectives

mvl:syn:con:
sec Classical propositional logic, and many other logics, use a set supply of *propositional constants* and *connectives*. For instance, we use the following as primitives:

1. The propositional constant for **falsity** \perp .
2. The propositional constant for **truth** \top .
3. The logical connectives: \neg (negation), \wedge (conjunction), \vee (disjunction), \rightarrow (**conditional**), \leftrightarrow (**biconditional**)

The same connectives are used in many-valued logics as well. However, it is often useful to include different versions of, say, conjunction, in the same logic, and that would require different symbols to keep the versions separate. Some many-valued logics also include connectives that have no equivalent in classical logic. So, we'll be a bit more general than usual.

Definition syn.1. A *propositional language* consists of a set \mathcal{L} of *connectives*. Each connective \star has an *arity*; a connective of arity n is said to be *n-place*. Connectives of arity 0 are also called *constants*; connectives of arity 1 are called *unary*, and connectives of arity 2, *binary*.

Example syn.2. The standard language of propositional logic \mathcal{L}_0 consists of the following connectives (with associated arities): \perp (0) \neg (1), \wedge (2), \vee (2), \rightarrow (2). Most logics we consider will use this language. Some logics by tradition an convention use different symbols for some connectives. For instance, in product logic, the conjunction symbol is often \odot instead of \wedge . Sometimes it is convenient to add a new operator, e.g., the determinateness operator Δ (1-place).

Photo Credits

Bibliography