

THREE-VALUES LOGIC

Mykola Rabchevskiy

In many cases, boolean logic that reduces everything to a choice between `true` and `false` is less adequate than a three-valued logic that adds a third option, `unknown`. When constructing hypotheses (for example, about the presence of a causal relationship - see [AGI: CAUSALITY](#)), testing using the available data has just such options: hypothesis refuted, confirmed, or the available data are insufficient either to confirm or to deny. Suppose there are several variants of a hypothetical dependence that are incompatible with each other and, at the same time, do not contradict the available data. In that case, it is not known which of them is correct (and whether there is one among the tested variants).

Three-valued logic is also helpful in purely technological situations. In the case of real-time computing, operations often have a timeout limit. Suppose the result is not what is required. In that case, it is helpful to distinguish the fundamental impossibility of performing the operation (for example, the lack of a necessary resource) from the situation when the allotted time is exhausted, but retrying makes sense. In such a situation, the third option (it is not known if it is possible to do what is required because the time is up) simplifies the coding.

You can learn more about three-valued logic from Wikipedia: [Three-valued logic](#), and a C++ class implementation with a complete set of logic operations is available here: <https://github.com/mrabchevskiy/logic>.

Subscribe to AGI engineering

By Mykola Rabchevskiy · Launched 2 years ago

AGI: fundamentals, architecture, implementation, source code