

AGI: WHERE EVENTS COME FROM

Mykola Rabchevskiy

THE INTERNAL REPRESENTATION OF THE EVENT DETECTOR



In the previous chapters, the concept of an **event** appeared a few times. Intuitively, in each specific context, it is pretty clear what this means. The requirement for the universality of information representation in the AGI system requires the universalization of the way of representing and operating with events.

Without giving a formal definition, we note that an event occurs at a particular ***moment in time***; this distinguishes an event from a ***process***: the process has a certain ***extent in time***, and the **event** corresponds to a ***time point***. Accordingly, events are naturally ordered in time.

AGI engineering is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

An event is essentially a signal that the situation has changed significantly, and the system must somehow react to this. For example, the temperature has reached the value at which it is required to turn on the air conditioner, two objects collided, a new object appears in the field of view, etc.

The current system situation is represented by an internal description that contains a set of attributes of two types: quantitative (temperature, coordinate, speed) and logical (on/off,

observable/unobservable). Therefore, a universal internal representation of an event is a **logical function** with an arbitrary number of arguments, including quantitative and logical parameters. Note that we are talking about the **mathematical** concept of a function; in code, this can be done in many ways.

Such a function is constructed from operations on numbers and logical values (Boolean or three-valued logic values). The presence of numeric variables obviously requires comparison operators that produce a logical value. The **event happens when this function changes its value**. The time intervals between events of the corresponding type are **intervals of constancy** of the value of the function that serves as a detector of events of this type. Detection is reduced to calculating the value of the function and comparing the current value with the previous one. The current value of the event detector function can naturally be considered a logical attribute of the situation description that can change at any moment. Accordingly, there is always some **delay** in detecting an event, and the amount of delay is determined by the **frequency** of checks.

In a continuum environment, the situation` description may include functions that predict changes in the quantitative attributes of the situation description. Each new data from sensors makes it possible to update the values of quantitative attributes, update the function of forecasting attribute changes over time, and determine **how much the actual value differs from the prediction**. A difference that exceeds a certain threshold signals a significant change in the situation and can serve as a detector of the corresponding event. At its core, this detector catches the **moment when smooth attribute changes are interrupted**.

So the event detector is another place where something inside an intelligent system is represented as a function. Other cases are invariant functions that describe cause-and-effect relationships, object detection, and predicting changes in quantitative attributes.

Since the AGI system must be able to detect objects **unknown** to it, and the presence of an object implies the **existence of events** related to it, the **construction of event detection functions during the operation of an intelligent system becomes inevitable**.

AGI engineering is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.