

The Impossibility of a Perfect Clock

Sami Al-Suwailem

December 15, 2022

Abstract

A Perfect Clock is an algorithm that can perfectly identify programs with a regular output from those with an irregular one. We argue that such an algorithm may not logically exist.

Definitions: A “Perfect Clock” is defined as an algorithm that can identify computer programs whose output is a regular string (henceforth “regular programs”) from those with an irregular one (“irregular programs”). A regular string is a sequence of digits that follows a perfectly repeating pattern.

Theorem: A Perfect Clock is logically impossible.

Proof: Deciding whether the output of an arbitrary computer program is regular or not, or the “Regularity Problem,” is a direct application of the Halting Problem. We first show the unsolvability of the Halting Problem, based on which we show the unsolvability of the Regularity Problem.

The Halting Problem

Suppose we can classify the universe of all computer programs \mathcal{P} into two groups: Those that halt and those that don’t. A program halts when it executes its commands and produces its output then stops processing the input.

Let $h(x)$ be a function that identifies *a priori* halting programs from non-halting ones.

Let:

- $H_1 = \{x : h(x) = 1\}$ be the set of all programs that halt.
- $H_2 = \{x : h(x) = 0\}$ be the set of all programs that do not halt.

Note that H_1 and H_2 are mutually exclusive such that:

- $H_1 \cap H_2 = \emptyset$, and

- $H_1 \cup H_2 = \mathcal{P}$.

However, H_1 and H_2 are themselves programs that must halt to be able to identify each group. This implies that $H_2 \in H_1$. But this contradicts the assumption that the two are mutually exclusive. It follows that $h(x)$ cannot exist. Hence, there is no universal algorithm that can identify *upfront* programs that halt from those that don't.

The Regularity Problem

Following the same reasoning above, suppose we can classify the universe of all programs into two groups: those with a regular output, and those with an irregular output.

Let $\rho(x)$ be a function that identifies *a priori* regular programs from irregular ones.

Let:

- $R_1 = \{x : \rho(x) = 1\}$ be the set of all regular programs.
- $R_2 = \{x : \rho(x) = 0\}$ the set of all irregular programs.

As before, R_1 and R_2 are mutually exclusive.

However, R_1 and R_2 are themselves regular programs because the output of each follows a perfectly repeated pattern. This implies that $R_2 \in R_1$, which contradicts the assumption that the two are mutually exclusive. It follows that $\rho(x)$ cannot exist. Hence, there is no universal algorithm that can identify upfront regular programs from irregular ones. Consequently, a Perfect Clock cannot logically exist. \square