The Impossibility of a Perfect Clock

Sami Al-Suwailem[1]

[1]Affiliation not available

December 2, 2022

Abstract

A Perfect Clock is defined as a Turing machine that can identify cyclical strings from acyclical ones. We argue that such a machine cannot logically exist.

The Impossibility of a Perfect Clock

Sami Al-Suwailem

December 2, 2022

Abstract

A Perfect Clock is defined as a Turing machine that can identify cyclical strings from acyclical ones. We argue that such a machine cannot logically exist.

Definition: A "Perfect Clock" is a Turing machine T the input of which is a stream of data S that commingles two types of strings: cyclical and acyclical strings. The cyclical string is a string that has a pattern the repeats perfectly throughout S. The output of T is the cyclical string only.

Theorem: A Perfect Clock is logically impossible.

Proof: Let τ be the characteristic function based on which the Perfect Clock can decide if a set of string is cyclical or not. Let:

- $S_1 = \{x : \tau(x) = 1\}$, the set of *all* cyclical strings in S, and
- $S_2 = \{x : \tau(x) = 0\}$, the set of *all* acyclical strings.

$S_1$and S_2 are exhaustive and mutually exclusive such that:

- $S_1 \cap S_2 = \emptyset$, and
- $S_1 \cup S_2 = S$.

Since S_1 includes *all* cyclical strings of S, then the set S_1 itself is acyclical because, by construction, it does not repeat within S. If S_1 is acyclical, it follows that $\tau(S_1) = 0$, which implies that $S_1 \in S_2$. But this contradicts the assumption that $S_1 \cap S_2 = \emptyset$. It follows that the function $\tau(x)$ does not exist, and therefore the Perfect Clock cannot exist as well. $\square$