

Putnam's Rocks Are Clocks

Ramón Casares

ORCID: [0000-0003-4973-3128](https://orcid.org/0000-0003-4973-3128)

Putnam proved that “every ordinary open system is a realization of every abstract finite automaton”, showing that computing is meaningless. Analyzing a simpler version of his proof, we conclude that giving a meaning to a computation requires computing, which is meaningless, starting a recursion.

Keywords: Putnam, implementation, computing, meaning.

§1 Putnam's theorem

¶1 · Putnam (1988), pages 121–125, proved the following theorem:

Every ordinary open system is a realization of every abstract finite automaton.

¶2 · An abstract finite automaton, also known as finite-state automaton, is a mathematical concept, so it does not need further specifications, see for example Mealy (1955), but an ordinary open system requires some additional physical stipulations in order to fit it into a mathematical theorem. So before the proof, Putnam explains the two Physical Principles that he needs: the Principle of Continuity and the Principle of Noncyclical Behavior. Putnam assures us that the Principle of Noncyclical Behavior is true for any physical system affected by a clock, and that every open system is affected.

¶3 · We will proceed otherwise. Instead of assuming that every ordinary open system is affected by a clock, we will use a clock as system. This way we can completely ignore the physics of the theorem, including the Principle of Noncyclical Behavior, and we can instead focus on its mathematical content. So we will restate Putnam's theorem in a simpler way by using a simple clock instead of the ordinary open system, we will prove it following his trail, and we will examine the consequences.

¶4 · Putnam's theorem shows that computing by itself is meaningless, and our examination shows that giving a meaning to a computation requires computing, which is meaningless as shown by Putnam's theorem. Therefore, Putnam's theorem is just a step in a recursion. Finally, I will propose a way out of the recursion.

This is DOI: [10.6084/m9.figshare.5450278](https://doi.org/10.6084/m9.figshare.5450278), version 20170928.

© 2017 Ramón Casares; licensed as cc-by.

Any comments on it to papa@ramoncasares.com are welcome.

§2 Simpler theorem

§2.1 Theorem

¶1 · We will prove the simpler theorem:

Every simple clock is a realization of every abstract finite automaton.

§2.2 Definition

¶1 · A simple clock is one that implements the successor function, so its starting state is 0, and the next state is the next natural number. So for any simple clock $s_0 = 0$, $s_1 = 1$, $s_2 = 2$, and in general $s_n = n$.

§2.3 Proof

¶1 · Any finite string of states, generated by any abstract finite automaton, can be realized by any simple clock using the following rule: assign the disjunction of the positions where it appears to each state in the string. For example, for string $ABABABA$, define $A = 0 \vee 2 \vee 4 \vee 6$ and $B = 1 \vee 3 \vee 5$. Using these definitions, the clock realizes the string. Q.E.D.

§2.4 Discussion

¶1 · As you can see, our proof has distilled the essence of Putnam's proof, avoiding any possible distraction. The analysis is now easier.

¶2 · The simple clock is perhaps the simplest clock, but the only requirement that any clock has to fulfill is that it cannot repeat any state. If, for example $s_j = s_n$, where $j \neq n$, that is, if state at time j were repeated at time n , not necessarily consecutive, then it could not realize every finite-state automaton, but only those that were at the same state at times j and n . The simple clock does not repeat any state, thus providing a unique index. In Putnam's proof, the Principle of Noncyclical Behavior grants a unique index.

¶3 · Using the simple clock, the rule used in our proof defines a partial function f that assigns a state of the finite-state automaton to the first natural numbers. Following the example, $f(0) = A$, $f(1) = B$, $f(2) = A$, $f(3) = B$, $f(4) = A$, $f(5) = B$, and $f(6) = A$. We will call this function the interpreting function.

¶4 · So the simple clock generates a series $0, 1, 2, 3, 4, 5, 6$, which is transformed by the interpreting function into the series $f(0), f(1), f(2), f(3), f(4), f(5), f(6)$ that realizes the string generated by the finite-state automaton $ABABABA$.

¶5 · By the simple clock definition in §2.2, we know that the next state of the simple clock will be $s_7 = 7$, but we cannot use the simple clock to foresee what will be the next state of the finite-state automaton. To make that prediction we will need to know the value of the interpreting function for number 7, that is, $f(7)$. And in general, to know the state of the finite-state automaton at any instant, we will need to know the total version of the interpreting function, that is, the value of $f(n)$ for any $n \in \mathbb{N}$.

¶6 · What indeed realizes the finite-state automaton is not the simple clock, which only provides a temporal index, but the interpreting function, which is not just a device of the proof, as it seemed to be. In other words, the proof puts all the information about the finite-state automaton into the interpreting function.

¶7 · Since Turing's (1937) proof, we know that realizing a function is equivalent to computing the function, where 'realizing' means 'effectively calculating' as in Church's (1935) thesis. As this is true for any function, it applies also to every interpreting function: realizing any interpreting function is computing it.

§3 Conclusion

¶1 · An idea behind Putnam’s theorem is that any computation is meaningless, except when it is interpreted by a mind. When you press, in this order, the four buttons labeled ‘2’, ‘+’, ‘3’, and ‘=’, of your calculator, its screen lights up some segments that you interpret as the number 5. You can say that your calculator is adding numbers, but it is only because you are giving some meanings to the labeled buttons and to the lighted up screen segments. In this example, you are the mindful agent who interprets that the physical ordinary open system that is your calculator is realizing the mathematical abstract finite automaton that defines addition. The same happens to a full programmable computer, as shown by Searle (1980) with his Chinese room argument.

¶2 · In the case of our proof, the interpreting function plays the the rôle of the interpreting mind. What we have found in our analysis is that the interpreting mind has to perform computations, too. Then Putnam’s theorem and Searle’s Chinese room argument show that computing by itself is meaningless, and our analysis shows that giving a meaning to a computation requires computing, which is meaningless, starting a recursion.

¶3 · To me, this recursion halts whenever a computation resolves a problem, because then the problem provides the meaning, so I would say that any computation is meaningful for its resolver, regardless whether it is mindful or mindless. Therefore, for those who can agree with me in this, all brain computations resolving the survival problems of living beings are meaningful.



References

- Church (1935): Alonzo Church, “An Unsolvable Problem of Elementary Number Theory”; in *American Journal of Mathematics*, vol. 58, no. 2, pp. 345–363, April 1936, DOI: [10.2307/2371045](https://doi.org/10.2307/2371045). Presented to the American Mathematical Society, April 19, 1935.
- Mealy (1955): George H. Mealy, “A Method for Synthesizing Sequential Circuits”; in *Bell System Technical Journal*, vol. 34, no. 5, pp. 1045–1079, September 1955, DOI: [10.1002/j.1538-7305.1955.tb03788.x](https://doi.org/10.1002/j.1538-7305.1955.tb03788.x).
- Putnam (1988): Hilary Putnam, *Representation and Reality*; The MIT Press, Cambridge, MA, 1988, ISBN: 978-0-262-66074-7.
- Searle (1980): John R. Searle, “Minds, Brains, and Programs”; in *The Behavioral and Brain Sciences*, vol. 3, no. 3, pp. 417–424, September 1980, DOI: [10.1017/S0140525X00005756](https://doi.org/10.1017/S0140525X00005756).
- Turing (1937): A. M. Turing, “Computability and λ -Definability”; in *The Journal of Symbolic Logic*, vol. 2, no. 4, pp. 153–163, December 1937, DOI: [10.2307/2268280](https://doi.org/10.2307/2268280).