Errors in Infinite Computations

Ramón Casares

ORCID: 0000-0003-4973-3128

When the error rate is not absolutely zero, infinite computations are necessarily erroneous, rendering them uninformative. This restricts the practical value of those hypercomputers that perform infinite computations.

Keywords: *physical hypercomputation, infinite computation, error rate, no free of errors law, computing limitations*

§1 Introduction

¶1 · For Shagrir & Pitowsky (2003): "A hypercomputer is a physical or an abstract system that computes functions that cannot be computed by a universal Turing machine." Then they present "the only known physical *digital* hypercomputer. This device [HM] performs a supertask, that is, it can carry out an infinite number of steps in a finite time-span." And finally, they "consider three objections to [their] contention that HM is a physical system that computes the function h", where "h characterizes the self-halting states of Turing machines." This note presents another objection.

§2 Memory

¶1 · The hyper-machine HM is "made up of a pair of communicating standard computers, T_A and T_B." By using a relativistic spacetime somewhat resembling that of the "twin paradox" fame, T_B can compute an infinite number of steps and can communicate with T_A, for which the elapsed time is finite, in such a way that T_B works as an oracle for T_A on infinite computations; see Turing (1938) on oracles.

¶2 · Turing (1936) machines are mathematical idealizations that abstract away some practical limitations of real computing: Turing machines are unbounded in time and tape, and they are error-free, see Casares (T). The hyper-machine HM resolves the limitation of time, but it is silent on the other two limitations: memory and errors.

¶3 · Though not all non-halting computations require unbounded memories, some will expand beyond any finite memory, and then, in resolving the halting problem, T_B will sometimes meet memory limitations. This is, in fact, objection #1 to which Shagrir & Pitowsky (2003) reply starting in page 88. In any case, regarding memory limitations, HM could not compute the function h in every case, because the computation would be aborted whenever T_B's memory were exhausted.

This is DOI: 10.6084/m9.figshare.13686439.v3, version 20220704.

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

§3 Errors

§3.1 The no free of errors law

¶1 · The same way that in mechanics there is a fundamental law negating perpetual motion, in computing another "no free lunch" law states that *the probability of an erroneous computation is never zero*, meaning that error-free computing is an idealization. One can base that law on the probabilistic nature of quantum mechanics, or on the laws of thermodynamics. However, it is better to present it as an empirical fact of engineering, because in science empirical facts trump theories.

¶2 · We can decrease the error probability by redundancy, that is, by performing the same computation more than once. However, though reduced, the new probability cannot be zero, showing that the *no free of errors law* in computing holds in this case, too. This is because, when performing a computation T times, there is a non-zero probability that all T computations are wrong (where p_j is the probability that computation j is erroneous):

$$0 < p_j \leq 1 \implies 0 < \prod_{j=1}^{T} p_j \leq 1.$$

¶3 · We are dealing here with bounded situations. Repeating infinitely or, more generally, with unbounded redundancy ($T \to \infty$), we could reach some certainty:

$$0 < p_j \leq P_\varepsilon < 1 \implies \prod_{j=1}^{\infty} p_j \leq \prod_{j=1}^{\infty} P_\varepsilon = \lim_{T\to\infty} {P_\varepsilon}^T = 0.$$

Sadly, computers are bounded systems.

§3.2 Constant case

¶1 · Therefore, in practice, the error rate P_e cannot be absolutely zero, and then the error rate is always positive: $0 < P_e \leq 1$. Then, assuming a constant error rate, the probability of an error-free computation of n steps is $(1 - P_e)^n$, and then it is zero when the number of steps grows to the infinite:

$$0 < P_e \leq 1 \implies \lim_{n\to\infty} (1 - P_e)^n = 0.$$

§3.3 General case

¶1 · In the general case, if $P_e(i)$ is the probability that a machine makes an error in step i, so $0 \leq P_e(i) \leq 1$ for every i, then the probability of an error-free infinite computation P_∞ is:

$$P_\infty = \prod_{i=1}^{\infty} (1 - P_e(i)).$$

¶2 · If for every i, $P_e(i) \geq P_\epsilon > 0$, that is, if the error rate of the machine cannot be less than P_ϵ, meaning that P_ϵ is the absolute error rate limit of the machine, then:

$$P_\infty = \prod_{i=1}^{\infty} (1 - P_e(i)) \leq \prod_{i=1}^{\infty} (1 - P_\epsilon) = \lim_{i\to\infty} (1 - P_\epsilon)^i = 0 \quad \implies \quad P_\infty = 0.$$

§3.4 Unbounded case

¶1 · The only way to circumvent this conclusion, $P_\infty = 0$, would be to build a computer with no absolute error rate limit, that is, with $P_\epsilon = 0$, which is not possible under the *no free of errors law* in computing. However, as seen above, this law applies only to bounded systems, so it might be possible with unbounded redundancy.

¶2 · Therefore, Andréka et al. (2018), page 213, propose that

> *instead of implementing the computer C [= T_B] by a single Turing machine, the programmer can use a fleet of self-reproducing robots which carry out the task of C [= T_B] in a massively parallel self-correcting fashion. Each robot lives for a finite time, and in each time instant there are finitely many robots. However, both numbers are unbounded, as our fleet of robots follow the strategy that sometime after each time, the computation is independently started from the beginning by more and more new robots with more and more life-expectancy.*

¶3 · This proposal by Andréka et al. (2018) deserves some comments:

- The proposal reads nicely, but they have not yet proved that their unbounded fleet could achieve, at least theoretically, a $P_\infty > 0$.
- It is out of the scope of Shagrir & Pitowsky's hyper-machine HM, because T_B (our computer C) is a standard computer, which is then bounded, and therefore it is not an unbounded system, as the unbounded fleet of robots is.
- And, of course, Andréka et al. will also need an unbounded budget to feed their ever burgeoning fleet of more and more new robots. In the case of infinite computations, at some point in time, their plan will necessarily collapse.

§3.5 Infinite computations are erroneous

¶1 · Assuming a constant error rate is assuming that the computer works in a stable environment, which is the most sensible assumption. Other assumptions are possible but, as long as the error rate would not decay to zero (asymptotically, since it cannot be zero), the conclusion will be the same, see §3.3: the probability of an error-free infinite computation is zero, $P_\infty = 0$. Button (2009; §2.3) argues similarly and he also reaches this conclusion, which is endorsed by Piccinini (2011; page 759), too.

¶2 · This means that T_B's answers as an oracle for T_A on infinite computations are as bad as those provided by a random source. In other words, regarding infinite computations, T_B's answers are completely uninformative. And, since finite computations do not need any oracle, the conclusion is that, when taking into account errors, the hyper-machine HM is not better than a universal Turing machine.

¶3 · This is not only applicable to the hyper-machine HM, because this conclusion can be extended to any non-error-free hyper-machine performing infinite computations.

§4 Discussion

¶1 · The hyper-machine HM presented by Shagrir & Pitowsky (2003) is a bounded system that completes infinite computations.

- HM is bounded because it is "made up of a pair of communicating standard computers, T_A and T_B", where the communication channel is not demanding: from T_A to T_B a program is transmitted, and from T_B to T_A a single signal is sent when that program halts, and none otherwise.
- HM is purported to calculate the halting function: it prints '1' if the transmitted program halts and it prints '0' if that program does not halt. For the last case, HM would have to complete an infinite computation.

¶2 · Both points are crucial. Unbounded idealizations simplify the theories by disregarding some real limitations. However, when it is proved that an unbounded idealization cannot do something, then we are sure that the corresponding bounded implementation will be even less able to do it. And in order to evaluate the limits of computing, the incumbent theory is the one by Turing (1936), which abstracts away time and memory limitations and errors that should be taken into account in addition to other common engineering limitations that affect all machinery, such as for example energy, volume, or resources. Now we can better appreciate the claim by Shagrir & Pitowsky (2003) on HM: even though HM is bounded, also in time, it can label infinite in time computations, which a universal Turing machine, unbounded in time, cannot do!

¶3 · Our analysis here was not concerned with the physical possibility of deforming the spacetime in such a way that we could see the end of time, but instead we benefited from the fact that real computing has other boundary conditions. And we focused on errors because the claim by Shagrir & Pitowsky was about infinite in time computations, and we showed that, under the *no free of errors law* in computing, the probability of an error-free infinite in time computation is zero, $P_\infty = 0$. There is no way to overcome this conclusion without breaking a boundary condition, as it is the case of the proposal by Andréka et al. (2018), which would require an unbounded quantity of resources to implement unbounded redundancy.

§5 Conclusion

¶1 · Contrary to Shagrir & Pitowsky's "contention that HM is a physical system that computes the function h" (page 88), our analysis shows that, either HM is not a physical system, but an idealization that ignores its errors, or, if we take into account its errors, then HM does not compute the function h.

Acknowledgements

The author thanks the two anonymous reviewers of the *Mind and Machines* journal who justly rejected the first version of this paper, making later ones a little better, I hope.

References

Andréka et al. (2018): Hajnal Andréka, Judit X. Madarász, István Németi, Péter Németi and Gergely Székely, "Relativistic computation", DOI: 10.1017/9781316759745.010. In *Physical Perspectives on Computation, Computational Perspectives on Physics*, (editors Michael E. Cuffaro and Samuel C. Fletcher), Cambridge University Press, Cambridge, UK, 2018, pp. 195–216, ISBN: 978-1-107-17119-0.

Button (2009): Tim Button, "SAD Computers and Two Versions of the Church-Turing Thesis"; in *The British Journal for the Philosophy of Science*, vol. 60, no. 4, pp. 765–792, December 2009, DOI: 10.1093/bjps/axp038.

Casares (T): Ramón Casares, "On Turing Completeness, or Why We Are So Many"; DOI: 10.6084/m9.figshare.5631922.

Piccinini (2011): Gualtiero Piccinini, "The Physical Church-Turing Thesis: Modest or Bold?"; in *The British Journal for the Philosophy of Science*, vol. 62, no. 4, pp. 733–769, December 2011, DOI: 10.1093/bjps/axr016.

Shagrir & Pitowsky (2003): Oron Shagrir and Itamar Pitowsky, "Physical Hypercomputation and the Church-Turing Thesis"; in *Minds and Machines*, vol. 13, pp. 87–101, 2003, DOI: 10.1023/A:1021365222692.

Turing (1936): A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem"; in *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937, DOI: 10.1112/plms/s2-42.1.230. Received 28 May, 1936. Read 12 November, 1936.

Turing (1938): A. M. Turing, "Systems of Logic Based on Ordinals"; in *Proceedings of the London Mathematical Society*, vol. s2-45, no. 1, pp. 161–228, 1939, DOI: 10.1112/plms/s2-45.1.161. Received 31 May, 1938. Read 16 June, 1938.