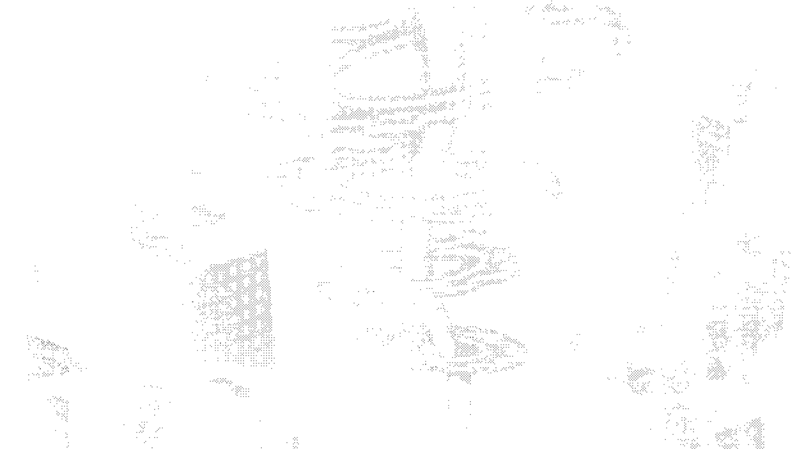

# Hipercomputación:

## La próxima generación de la computación teórica


Andrés Sicard
Mario Vélez



### 1. RESUMEN

Se construye un contexto adecuado para hablar de hipercomputación a partir de la tesis de Church-Turing y la tesis M. Se presentan las características generales de tres de modelos de hipercomputación (máquinas con oráculos, máquinas de Turing aceleradas y redes neuronales recurrentes análogas) y se señalan algunas ideas en relación con su posible o imposible implementación.


### 2. INTRODUCCIÓN

La noción de 'computabilidad' es considerada usualmente como una noción absoluta. Un objeto (función, número, proceso, etc.) es computable si y sólo si es Turing-computable, es decir, si es computado por una máquina de Turing (MT). En este sentido es posible hablar de objetos no computables, es decir, de objetos no Turing-computables; quizás el ejemplo más conocido sea el problema de la parada (*halting problem*), que consiste en determinar la finalización o la no finalización de la ejecución de una MT cualquiera, con una entrada cualquiera. Si por el contrario, se considera la noción de 'computabilidad' una noción relativa -tal como es considerado por los autores- es posible hablar de modelos de computación que computen objetos no Turing-computables. Estos modelos de computación inicialmente fueron referenciados como *'super-Turing computation'*, *'computing beyond the Turing limit'*, *'breaking the Turing barrier'*, etc.; esta refencia es errónea, debido a que el mismo Alan Turing propuso uno de los primeros de tales modelos (Copeland y Sylvan, 1999, Turing, 1939). Estos modelos reciben hoy el nombre de hipermáquinas o hipercomputadores (Copeland y Proudfoot, 1999).

La hipercomputación es la teoría de los hipercomputadores. Aunque uno de los modelos de hipercomputación fue propuesto sólo dos años después que las máquinas de Turing, y aunque existen modelos de hipercomputación propuestos desde muy diferentes consideraciones, el área de la hipercomputación está en proceso de gestación y consolidación.

*The Turing Project* dirigido por Jack Copeland y Diane Prooufoot, en la Universidad de Canterbury, Nueva Zelandia (www.alanturing.net) es uno de los grupos lideres a nivel mundial de esta temática y organizó en el mes de mayo de 2000, el primer *Hipercomputation Workshop* en *University College*, Londres, Inglaterra (www.alanturing.net/conference/hyper/hypercom.html). Por otro lado, en la dirección www.hypercomputation.net es posible encontrar información actualizada sobre el área.

### 3. POSIBILIDAD DE LOS MODELOS DE HIPERCOMPUTACIÓN

Podría pensarse que la imposibilidad de hipercomputación está sustentada en la


ANDRÉS SICARD. Ingeniero de Sistemas, Universidad EAFIT; Máster en Ingeniería Informática, Universidad EAFIT. Profesor del departamento de Ciencias Básicas, Universidad EAFIT.
E-mail: asicard@eafit.edu.co
MARIO VÉLEZ. Físico, Universidad de Antioquia: Máster en Física, Universidad de Antioquia. Profesor del departamento de Ciencias Básicas, Universidad EAFIT.
E-mail: mvelez@eafit.edu.co

tesis de Church-Turing que usualmente es entendida como la tesis que afirma que cualquier tarea efectiva de manipulación de información realizada por una máquina puede ser realizada por una máquina de Turing, sin embargo, lo que afirma esta tesis es:

> *"Thesis Church-Turing: Any procedure than can be carried ouy by an idealised human clerk working mechanically with paper and pencil can also be carried out by a Turing machine".* (Copeland y Sylvan, 1999, p. 50).

> **Si por el contrario, se considera la noción de 'computabilidad' una noción relativa -tal como es considerado por los autores- es posible hablar de modelos de computación que computen objetos no Turing-computables. Estos modelos de computación inicialmente fueron referenciados como *'super-Turing computation´, 'computing beyond the Turing limit´, 'breaking the Turing barrier´*, etc.; esta refencia es errónea, debido a que el mismo Alan Turing propuso uno de los primeros de tales modelos (Copeland y Sylvan, 1999, Turing, 1939). Estos modelos reciben hoy el nombre de hipermáquinas o hipercomputadores (Copeland y Proudfoot, 1999).**

Existe entonces una diferencia entre la tesis cuyos objetos son las máquinas y la tesis de Church-Turing cuyos objetos son los *computors ("idealized human calculating in a purely mechanical fashions")* (Soare, 1996, p. 292). La primera tesis es más amplia y fue presentada inicialmente por Robin Gandy (quien fue alumno de Turing) bajo el nombre de tesis M (Gandy, 1980). Esta tesis es presentada como:

> *"Thesis M: Whatever can be calculated by a machine (working on finite data in accordance with a finite program of instructions) is Turing machine computable".* (Copeland, 1996, p. 5).

Como es señalado por Copeland, la tesis M admite dos posibles interpretaciones:

> *"Thesis M itself admits of two interpretations, according to whether the phrase 'can be calculated by a machine' is taken in the narrow sense of 'can be calculated by a machine that conforms to the physical laws (if not to the resource constrains) of the actual world', or in a wide sense that abstracts from the issue of whether or not the notional machine in question could exist in the actual world. The narrow version of the thesis M is an empirical proposition whose truth-value is unknown. The wide version of the thesis M is known to be false. Various notional machines have been described which can calculate functions that are not Turing machine computable ... "* (Copeland, 1996, p. 6).

Modelos usados para demostrar que la versión amplia de la tesis M es falsa son modelos de hipercomputación. Para construir un modelo de hipercomputación con frecuencia se parte de un modelo equivalente a una máquina de Turing y bajo algunas modificaciones éste viene a ser un modelo de hipercomputación. Para demostrar que una máquina que computa es un modelo de hipercomputación se puede proceder de dos formas: la primera es demostrar que la máquina es equivalente computacionalmente a otro modelo de hipercomputación conocido, o se demuestra que la máquina puede resolver un problema que una máquina de Turing no puede resolver (por ejemplo, el problema de la parada). Hoy en día, existen modelos de hipercomputación desde diferentes teorías como la indica la siguiente sección.

## 4. MODELOS DE COMPUTACIÓN

- **Máquinas con oráculos**

Dos años después de presentar las máquinas de Turing, el mismo Turing al interior de su tesis de Ph.D. (la cual fue supervisada por Alonso Church) presenta la noción de máquina con oráculo (MO) (Turing, 1939). Una MO puede ser imaginada como una MT con una propiedad adicional: la máquina es asistida por un oráculo, a quien puede realizar preguntas y de quien recibe respuestas binarias a las preguntas realizadas. Infortunadamente, Turing no presentó ninguna característica matemática o física de los oráculos, lo único que él mencionó fue:

> *"Let us suppose that we are supplied with some unspecified means of solving number-theoric problems; a kind of oracle as it were. We shall not go any further into the nature of this oracle apart from saying that it cannot be a (Turing) machine".* (Turing, 1939, p. 172-173).

La presentación de una MO en el lenguaje de las MT, requiere un pequeño cambio en el formato de instrucciones de éstas (Davis, 1982, Rogers, 1992). El oráculo puede ser pensado como un conjunto con un procedimiento de decisión sobre sus elementos, de forma tal que la máquina puede preguntar al oráculo si un determinado elemento sobre su cinta pertenece o no pertenece al conjunto, y con base en la respuesta, la máquina realiza una posible acción. Entonces, si el conjunto es recursivo (su función característica es Turing-computable), la MO es una MT, pero si el conjunto no es recursivo, la MO es un modelo de hipercomputación.

Aunque desde el punto de vista lógico-matemático es posible hablar de la existencia de una MO, desde el punto de vista de su implementación, o si se prefiere, desde el punto de vista físico, la situación es muy diferente. Aunque Turing no mencionó como implementar ni una MT ni una MO, es conocido que los computadores actuales son implementaciones (parciales) de una MT universal y cada programa que se ejecuta sobre éstos es una implementación de una MT particular. Sin embargo, en la actualidad no se sabe como podría ser la implementación de un MO, inclusive es poco factible que dicha implementación sea posible (Copeland y Proudfoot, 1999). De ser esto último cierto, se estaría hablando entonces de máquinas de computación teóricas sin ningún referente físico, pero no por ello carentes de interés, como será ampliado en las conclusiones.

- **Máquinas de Turing aceleradas**

Los modelos clásicos de computación (máquinas de Turing, funciones parciales recursivas, sistemas de Post, etc.) asumen que cada operación realizada por ellos, es ejecutada en una unidad de tiempo finita y discreta. Sin embargo, es posible pensar en un modelo de computación que ejecute su primera operación en una unidad de tiempo, su segunda operación, en media unidad de tiempo, su tercera operación, en un cuarto de unidad de tiempo y así sucesivamente. Si el modelo de computación es una máquina de Turing, entonces se habla de una máquina de Turing acelerada (MTA) (Copeland, 1998b, Copeland, 1998a).

El término 'supertarea' se refiere a la ejecución de un número infinito de acciones en una cantidad finita de tiempo (Laraudogoitia, 1999). Dado que $\sum_{i=0}^{n=\infty} \frac{1}{2^n} = 2$, una MTA puede ejecutar un número infinito de instrucciones en dos unidades de tiempo, es decir, puede realizar una supertarea que le permita el cómputo de un problema no computable (por ejemplo, el problema de la parada), constituyéndose así, en un modelo de hipercomputación (existe un variante de máquinas de Turing que realizan supertareas, denominada *infinite time Turing machines* (Hamkins y Lewis, 2000)). De forma similar a una MO, la existencia lógico-matemática y la existencia física de una MTA es diferente. Dejando a una lado importantes consideraciones filosóficas sobre la ejecución de supertareas (Laraudogoitia, 1999), no es claro en la actualidad como llevar a cabo la implementación de una MTA. En este caso sin embargo, el pensar en diferentes concepciones espacio-temporales, en particular concepciones relativistas, acercan un poco más la noción de MTA a un posible referente físico.

- **Redes neuronales recurrentes análogas**

No todos los modelos de hipercomputación actuales son construidos como variaciones de las máquinas de Turing. A manera de ejemplo se presenta un modelo de hipercomputación desde la teoría de las redes neuronales. Estas redes se pueden clasificar de acuerdo a su arquitectura en redes *feedforward* (redes de alimentación hacia adelante o por niveles) o en redes *feedback* (redes de alimentación hacia atrás o recurrentes).

Hava Sielgelmann y Eduardo Stong proponen un red neuronal *feedback* llamada *Analog Recurrent Neural Network* (ARRN). Bajo una función de activación apropiada; una ARRN es equivalente a un autómata de estado finito si los pesos entre las neuronas son restringidos a los números enteros; por otra parte, si los pesos entre las neuronas son restringidos a números racionales, la ARRN es equivalente a una MT; finalmente, la ARRN es un modelo de hipercomputación si los pesos entre neuronas son números reales (Siegelmann, 1999, Siegelmann y Sontag, 1994).

## 4. CONCLUSIONES

Desde un punto de vista físico, toda operación de manipulación de información se debe realizar sobre un sistema físico de alguna naturaleza, por lo tanto el «mundo» físico subyacente al modelo de computación elegido, establece las propiedades y la limitaciones inherentes a éste. Es decir, las características temporales (discreto, continuo), espaciales (discreto, continuo), evolutivas ((determinista, probabilista), (reversible, irreversible)), y las diferentes interrelaciones entre ellas, establecen los límites de la computación (existe una propuesta en una dirección inversa a la anterior llamada *general-Timed X-machine*, que consiste en proponer un modelo de computación que reciba como parámetros las características del mundo físico sobre el cual va a operar (Stannett, ??)).

La idea del referente físico puede llevarse al extremo y afirmar que la computabilidad es un área de la física y que no tiene sentido hablarse de computabilidad sin un referente físico, tal como lo ha afirmado David Deutch (Deutsch et al., 1995, Brown, 2000) entre otros. Deutch es uno de los principales fundadores de la computación cuántica y se ha visto que su idea tienen gran aceptación entre las personas que trabajan en esta área.

Desde esta perspectiva, algunas funciones consideradas clásicamente como Turing-computables, deberían considerarse como no computables, pues su evaluación requiere recursos - ya sean temporales, ya sean espaciales (memoria)- mayores que los que físicamente pueden ser asignadados. Con mayor razón, la carencia de referente físico de los modelos de hipercomputación, eliminaría la posibilidad de considerar a ésta como un tipo de computación.

Sin embargo para los autores, la teoría de la computabilidad no requiere de un referente físico y mucho menos de una posible implementación, que determine la validez de sus modelos. La existencia de éstos, se equipara a la existencia de los objetos matemáticos, en cuyo caso es adecuado hablar de "realidad construida" en lugar de "realidad objetiva".

## 5. AGRADECIMIENTOS

Este artículo fue financiado por la universidad EAFIT, bajo el proyecto de investigación número 817424.

**Sin embargo para los autores, la teoría de la computabilidad no requiere de un referente físico y mucho menos de una posible implementación, que determine la validez de sus modelos. La existencia de éstos, se equipara a la existencia de los objetos matemáticos, en cuyo caso es adecuado hablar de «realidad construida» en lugar de «realidad objetiva».**

## BIBLIOGRAFÍA

Brown, Julián. (2000). Minds, Machines and the Multiverse: The Quest for the Quantum Computing. New York: Simon & Schuster.

Copeland, Jack. (1996). The Church-Turing Thesis. J. Perry and E. Zalta (Eds.) Stanford Encyclopedia of Philosophy plato. stanford.edu/entries/church-turing/.

Copeland, Jack. (1998a). Super Turing-Machines. En: *Complexity*, **No. 4,** pp. 30-32.

Eprint: www.phil.canterbury.ac.nz/philsite/people/jack/jack_copeland.html.

Copeland, Jack. (1998b). Unconventional Models of Computation, capítulo Even Turing Machines Can Compute Uncomputable Functions, páginas pp. 150-164. Singapure: Springer-Verlag.

Copeland, Jack y Proudfoot, Diane (1999). Un Alan Turing desconocido. En: *Investigación y Ciencia*, pp. 15-19.

Copeland, Jack y Sylvan, Richard. (1999). Beyond the Universal Turing Machine. En: *Australasian Journal of Philosophy*, **No. 77,** páginas 44-66. Eprint:ww.phil.canterbury.ac.nz/philsite/people/jack/jack_copeland.html.

Davis, Martin. (1982). Computability and Unsolvability. New York: Dover Publications, Inc.

Deutsch, David. et al. (1995). Universality in Quantum Computation. Eprint: arXiv.org/abs/quant-ph/9505018.

Gandy, Robin. (1986). Church's thesis and principles for mechanisms. En The Kleene symposium, pp. 123-148. The Kleene symposium, North-Holland.

Hamkins, Joel y Lewis, Andy. (2000). Infinite Time Turing Machines. En: *The Journal of Simbolic Logic*, **No. 65(2),** p. 567-604 (2000). Eprint: arXiv.org/abs/math.LO/9808093 (versión preliminar).

Laraudogoitia, Jon P., Perry, J. and Zalta, E. (1999). (Eds.) Stanford Encyclopedia of Philosophy plato.stanford.edu/entries/spacetime-supertasks/.

Rogers, Hartley. (1992). Theory of recursive functions and effective computability. Cambridge: The MIT Press, pág. ix + 482.

Siegelmann, Hava. (1999). Neural Networks and Analog Computation. Beyond the Turing Limit. Progress in Theorical Computer Science. Birkhäuser.

Siegelmann, Hava T. y Sontag, Eduardo D. (1994). Analog Computation Via Neural Networks. En: *Theoretical Computer Science*, **No. 131,** pp. 331-360.

Soare, Robert I. (1996). Computability and Recursion. En: *The Bulletin of Symbolic Logic,* pp. 284-321. Eprint: www.math.ucla.edu/ /,asl/bsltoc.htm.

Stannett, Mike. Computations over arbitrary temporals models. Eprint.

Turing, Alan. (1939). Systems of logic based on ordinales. En: *Proc. London Math. Soc,* No. 2(45), pp. 161-228.