

<http://doi.org/10.21555/top.v0i58.1090>

Modelling Strategies in Alan Turing: Terms and Concepts of Machine

Estrategias de modelización en Alan Turing: términos y conceptos de máquina

Andrés Ilcic

Universidad Nacional de Córdoba
Argentina

ailcic@ffyh.unc.edu.ar

<https://orcid.org/0000-0003-3966-8924>

Pío García

Universidad Nacional de Córdoba
Argentina

piogarcia@ffyh.unc.edu.ar

<https://orcid.org/0000-0001-7450-6539>

Recibido: 05 - 09 - 2018.

Aceptado: 23 - 10 - 2018.



This work is licensed under a Creative Commons Attribution
-NonCommercial-ShareAlike 4.0 International License.

Abstract

In 1936, Alan Turing proposed the notion of an automated machine as a model of the computation performed by a human being while only being aided by mechanical resources. Still, it seems that much more can be said about Turing's own conception of machines in the scope of his later work, both terminologically and conceptually. In this paper we present the terms he used that refer to machines and that according to our understanding are important to give an account of Turing's concerns and the problems he tackled with after his first recourse to the notion of machine during the late nineteen-thirties. Exploring his usage of such terms we show how it is possible to see an enlargement or extension towards a general notion of machine from the very first automated machine. At the same time, we identify in his work a *modelling attitude or stance* in which machines can be used as a way to understand and explain a natural or abstract phenomenon.

Keywords: Alan Turing; machine; modelling; computation; epistemology.

Resumen

En 1936 Alan Turing propuso sus máquinas automáticas como modelo de la computación realizada por un ser humano asistido sólo con recursos mecánicos. Sin embargo, tanto terminológicamente como conceptualmente, habría más para decir acerca de la concepción de máquina en la obra de Turing. En el presente trabajo presentamos términos que, a nuestro entender, son relevantes para dar cuenta de las preocupaciones y problemas que Turing abordó luego de la propuesta inicial de fines de los años treinta. En el uso de dichos términos trataremos de mostrar la ampliación o extensión de la noción de máquina en general a partir de la original de "máquina automática". A su vez, identificamos en su obra una *intención o actitud modeladora*, en la que las máquinas se pueden usar como una forma de entender y explicar un fenómeno natural o abstracto.

Palabras clave: Alan Turing; máquina; modelización; computación; epistemología.

The successful construction of all machinery depends on the perfection of the tools employed; and whoever is a master in the arts of tool-making possesses the key to the construction of all machines... The contrivance and construction of tools must therefore ever stand at the head of the industrial arts.

—Charles Babagge, *The Exposition of 1851, or, Views of the industry, the science, and the government, of England.*

Introducción

Preguntar por las nociones de máquinas en la obra de Alan Turing parece una tarea sencilla con una respuesta directa: aquellas máquinas que hoy llevan su nombre. Como es bien conocido, en 1936 Turing propuso sus máquinas automáticas como modelo de la computación realizada por un ser humano asistido sólo con recursos mecánicos, a quienes en la época se les conocía como “computadoras”. Sin embargo, tanto terminológicamente como conceptualmente habría más para decir acerca de la concepción de máquina en la obra del mismo Turing, más allá de la máquina automática. Este investigador utilizó un conjunto de expresiones para referirse a diferentes *aspectos, problemas* y *contextos de aplicación*, todas en torno a las máquinas.

En el presente trabajo presentamos términos que, a nuestro entender, son relevantes para dar cuenta de las preocupaciones y problemas que Turing abordó luego de su propuesta inicial de fines de los años treinta. En el uso de dichos términos trataremos de mostrar la ampliación o extensión, en un cierto sentido que aclararemos, de la noción de máquina en general a partir de la de “máquina automática”. A su vez, este recorrido nos permitirá mostrar cierta evolución en el pensamiento de Turing, especialmente en lo que concierne a su manera de abordar el problema del conocimiento de los fenómenos del mundo natural.

Planteo del problema

Entre fines de los treinta y principios de los años cincuenta, Turing utiliza diferentes expresiones para hacer referencia a tipos de máquinas o mecanismos que permiten abordar o aproximarse de manera apropiada a problemas particulares. La primera propuesta de Turing, y sin duda la más conocida, es la que aparece en su trabajo de 1936. En dicho texto,

Turing comienza haciendo referencia a lo que puede ser descripto como “computable” en términos de aquello que es calculable por “medios finitos”, para finalizar con la propuesta (es decir, su *definición*) de que lo computable está vinculado con aquello que puede ser generado por una máquina.¹ Al intentar responder por la “máquina” que sería capaz de realizar esta tarea, Turing compara —y a su vez modeliza— el proceso de cálculo mecánico llevado adelante por un ser humano: “Podemos comparar a un hombre en el proceso de computar un número real con una máquina que sólo es capaz de un número finito de condiciones q_1, q_2, \dots, q_R , que serán denominadas como ‘m-configuraciones’” (Turing, 1936; Copeland, 2004, p. 59; Cooper 2013, p. 17). Por esta razón, la vinculación entre los procesos humanos y los de una máquina se realiza, en principio, a través de la explicitación de la finitud de los medios utilizados. A continuación, Turing presenta los “análogos” de los medios físicos utilizados (como el papel) y de los procedimientos, incluyendo los “estados de la configuración”, para concluir que su propuesta es “que estas operaciones incluyen todo aquello que es usado en la computación de un número. La defensa de esta propuesta será más fácil cuando la teoría de las máquinas le sea familiar al lector” (Turing, 1936; Copeland, 2004, p. 60; Cooper, 2013, p. 17).

Una máquina automática será aquella en la cual cada movimiento “esté completamente determinado por su configuración” (Turing, 1936; Copeland, 2004, p. 60; Cooper, 2013, p. 17). Si esta máquina automática tiene la configuración que Turing ya ha presentado en su modelización del proceso humano de cálculo mecánico, entonces se la puede llamar una “máquina de cómputo”. La “máquina” en el trabajo de 1936 aparece, entonces, en directa analogía con las computadoras humanas y se expone como un “modelo mínimo” de aquello que podría estar ocurriendo en el proceso de realizar un cálculo. Lo esencial para que dicho modelo funcione es que las operaciones se puedan llevar a cabo de manera mecánica y autónoma, es decir que, una vez iniciada una secuencia la misma pueda dar un resultado sin futuras intervenciones de un usuario. Pero, ya en este trabajo, Turing concibe máquinas no automáticas que requieren la entrada de parámetros auxiliares. Sin embargo, se concentra en las máquinas automáticas ya que son el recurso

¹ Un “número es computable si sus decimales pueden ser escritos [*written down*] por una máquina” (Turing, 1936; Copeland, 2004, p. 58). Todas las traducciones son nuestras.

necesario para resolver el problema particular que estaba atacando: el llamado de la decisión (o *Entscheidungsproblem*), que se desprende del núcleo del programa de Hilbert de fundamentación de la matemática.

Si bien la máquina de Turing *per se* es un modelo mínimo de un tipo de operaciones, como resultado es un “modelo máximo” ya que define los límites de lo que se puede computar, al menos en la interpretación clásica de computable. La noción central y canónica de computadora depende de esta modelización.² Es una cuestión controvertida dar cuenta de las razones por las cuales se aceptó la propuesta de Turing como una caracterización de lo computable. Kurt Gödel fue uno de los primeros en defender la solución de Turing (cfr. Sieg, 1994, p. 89). La equivalencia extensional de los formalismos fue un argumento de peso. Gödel escribía en 1946:

Tarski ha destacado en su conferencia la gran importancia (y creo justamente) del concepto de recursividad general (o computabilidad de Turing). Me parece que esta importancia se debe en gran parte al hecho de que con este concepto se ha logrado dar una definición absoluta de una noción epistemológica interesante, es decir, una que no depende del formalismo elegido (Gödel, 1990, p. 150).

Esta idea parece dejar fuera las caracterizaciones intensionales de computación. Sin embargo, por esta época ya había otras interpretaciones que intentaban explicar el acuerdo en torno a la tesis Church-Turing desde otra perspectiva. Emil Post, por ejemplo, ya en 1941 comentaba que “establecer [la tesis] no es una cuestión de prueba matemática sino de análisis psicológico de los procesos mentales involucrados en procesos matemáticos combinatorios” (Post, 1941, citado por Shapiro, 2006, p. 44).

A pesar de la preeminencia de la versión extensional de la tesis, varios filósofos han defendido caracterizaciones de computación de tipo intensional. Así, por ejemplo, Piccinini (2007) defiende una noción de

² Como veremos, ya desde la primera postulación de las máquinas de Turing se puede notar una tensión que será explorada en profundidad por la filosofía de la computación entre la noción de computable o computación y la de computadora.

computación en términos de mecanismos caracterizados en términos de sus componentes y de su organización. Shapiro (2006) sugiere que hay una noción intensional asociada con la computación –más específicamente con la efectividad de los procedimientos– que debería entenderse como un concepto epistémico o como una manera de hacer referencia a nuestro acceso epistémico a la computación. Sieg (2002) ha destacado que, a pesar del éxito en el desarrollo de formalismos, la discusión acerca de los fundamentos de la computación requiere explicitar el alcance de la tesis Church-Turing. Para ello se requiere discutir hasta dónde está atada la caracterización de computación al análisis de las operaciones mecánicas de los seres humanos y de las operaciones de cierto tipo de máquinas. De manera semejante Copeland y Sylvan (1999), presentan lo que los autores llaman una noción relativa de computación. El alcance de lo computable dependería de los recursos disponibles. Esta es la razón de la insistencia en diferenciar entre computación que puede realizar un ser humano y computación realizada por una máquina. A su vez Shagrir (2002) dice que si esta tesis, defendida por Sieg y por Gandy, es correcta, entonces la noción de computación que utilizamos hoy es diferente a la propuesta por Turing.

En este sentido, si nos quedamos con una caracterización extensional de computación no parece que los usos terminológicos ampliados de máquina que se muestran en la obra de Turing tengan un significado conceptual sustantivo. Evidentemente todas estas nociones de máquina son reductibles, en principio, a la máquina automática (*máquina de cómputo*) o, como la conocemos ahora, máquina de Turing, en tanto ésta es la descripción de los pasos básicos que debe llevar adelante un sistema para realizar una tarea de manera mecánica. Además, la estrategia de Turing al tomar la actividad mecánica de un ser humano puede ser considerada una analogía útil. En este sentido, la actitud modelizadora de Turing podría ser vista como accidental, secundaria y prescindible.

Sin embargo, como hemos visto, hay varias propuestas de la literatura filosófica que cuestionan este punto de vista. Nuestra propuesta va en esa dirección, pero con algunas diferencias importantes. A nuestro entender habría al menos dos aspectos en los cuales la reducción extensional en principio no es adecuada para comprender la manera en la que Turing utiliza la noción de máquina en sus trabajos posteriores. En primer lugar, la caracterización de estos otros tipos de máquina permite comprender la manera en la cual Turing abordaba problemas específicos. En segundo lugar, los tipos de máquina que sugiere Turing

involucran un enriquecimiento y sofisticación de la noción misma de máquina automática. Esto último es posible si aceptamos, como parece hacerlo el mismo Turing, que hay una noción no trivial de máquina que se construye a partir de la relación con problemas abordados y con las clases de caminos que se estima que son convenientes para atacar dichos problemas. Es decir que las máquinas *extendidas* tendrían un doble rol: por un lado, el de proveer una heurística para abordar un problema (tal y como ya había sucedido con la formulación original) y, a su vez, el de funcionar como un modelo mediante el que se puede obtener una “descripción de trabajo” de lo que ocurre en aquello que se pretende estudiar, ya no meramente en un contexto formal o matemático.³

Antes de continuar es necesario aclarar que existe otra forma de extender y volver más rica la noción de máquina de Turing, centrándose en la máquina automática y en los procesos computacionales. Comentamos brevemente dos posibilidades que han surgido en la literatura. La primera, y que debería considerarse como la primera extensión propia de la máquina de Turing en tanto pretende considerarla desde el punto de vista de su implementabilidad física, es la que propuso Alonso Church en su breve reseña de 1937 del artículo original de Turing. A su vez, en esta primera interpretación ya puede verse el peso que tendrán las nociones meramente extensionales de computación. Church comienza la reseña comentando que

El autor [Turing] propone como criterio de que una secuencia infinita de dígitos 0 y 1 sea “computable” que sea posible diseñar [*devise*] una máquina de computar, ocupando un espacio finito y con partes operativas de tamaño finito que escribirán la secuencia al número deseado de términos si es que se le permite funcionar por un tiempo lo suficientemente largo. Por conveniencia, se le imponen ciertas otras restricciones al carácter de la máquina, pero éstas son de una naturaleza tal que no obviamente no causan pérdida alguna de generalidad – en particular, un calculador humano provisto con lápiz

³ Curiosamente, este último movimiento puede leerse como el inverso de la formulación de la máquina automática, al menos en su sentido original, en la que un modelo de algo que efectivamente ocurre en el mundo es usado para describir una propiedad matemática abstracta.

y papel e instrucciones explícitas puede ser considerado como una clase de máquina de Turing (Church, 1937, pp. 42-43).

En el trabajo de Turing, la importancia de la máquina estaba puesta en su carácter de modelo de la computación llevada a cabo por las computadoras humanas y el énfasis en el carácter finito de la máquina sólo aparecía en términos abstractos, mediante la restricción a que su tabla de comportamiento sólo pueda contener un número finito de términos, aunque sea capaz de usarlos para producir un número infinito de resultados (como imprimir un decimal, por ejemplo). Church, por el contrario, le da una entidad mucho más física a la máquina de Turing y el aspecto de modelo de un proceso computacional queda relegado a un caso particular de dicha máquina. En Gandy (1980) encontramos un análisis detallado de la propuesta de Church, que incluye también una definición precisa de lo que se entiende por “máquina”.⁴

Otra posibilidad de extender la noción original de Turing radica en concentrarse más en sus capacidades que en su estructura, esto es, en el hecho de que lo que puede hacer una máquina de Turing es, después de todo, implementar un algoritmo. Las “Máquinas de Estados Abstractos” (*Abstract State Machines* o ASM) propuestas originalmente por Yuri Gurevich (1995; 2018) son un caso ejemplificador de una extensión en dicha dirección. En estas máquinas, cada estado es una estructura arbitraria de datos, entendiendo estructura en el sentido matemático de un álgebra o un conjunto no vacío con funciones y relaciones definidos sobre el mismo. El funcionamiento de la máquina está definido por las reglas de transición entre estados que tienen una forma condicional para detectar el estado (descrito en lógica de predicados sin variables libres) y aplicar la actualización, que es el consecuente del condicional y es un conjunto finito de asignaciones que cambia en paralelo el valor

⁴ Una “Máquina de Gandy” es aquella que respeta los siguientes cuatro principios, ninguno de los cuales podría ser violado sin violar la tesis de Church-Turing: i) las máquinas son descritas usando conjuntos hereditarios finitos con etiquetas para sus partes; ii) el rango conjunto teórico de las máquinas es acotado, es decir que no hay ningún ensamblado que contenga cadenas infinitas de subensamblados; iii) todas pueden ser ensambladas con partes de tamaño finito; y iv) los estados sucesivos de dichas máquinas pueden ser reconstruidas sólo recurriendo a causas y efectos locales (es decir, dentro de una vecindad determinada y finita).

de los argumentos de las funciones especificadas. Una de las ventajas de estas máquinas es que su descripción abstracta de los algoritmos permite utilizarlas tanto para evaluar a los modelos matemáticos de computación –como la máquina de Turing– como para evaluar los métodos de especificaciones formales para programas de computación, cubriendo así la distancia teórica que existía entre dichos campos.⁵

Una segunda forma de ir más allá de la máquina automática –y un tanto más cercana a la que queremos defender desde la obra posterior del mismo Turing– consiste en extender dichas máquinas bajo otra clase de restricciones físicas, ya no considerando su implementación sino la capacidad de la misma para simular un proceso físico en el mundo y, al mismo tiempo, su capacidad de servir como instrumento de mediación en nuestro conocimiento de dicho proceso del mundo. Una de las primeras sugerencias en esta otra dirección es la de Brosl Hasslacher (1988). En dicho artículo, plantea la necesidad de cambiar la perspectiva sobre los dispositivos computacionales y sobre los agentes epistémicos, considerando a estos como parte del mundo físico, pudiendo así “ignorar lo que es en principio posible con arquitecturas computacionales hipotéticas con recursos ilimitados” pasando a examinar “sólo las preguntas que uno puede hacerle a una máquina física de computar, con una cierta precisión, en un tiempo razonable” (Hasslacher, 1988, p. 418). Estos supuestos lo obligan a pensar a un físico acotado a modelar la complejidad de un fenómeno del mundo con algoritmos eficientes en tiempo polinómico, de manera tal que puedan generar cierto *insight* sobre la estructura interna detrás del mecanismo que produce tal fenómeno. Según Hasslacher, en el análisis clásico de las máquinas de computar basado en las máquinas de Turing y en la teoría de la complejidad computacional, el foco está puesto en la relación entre las entradas y las salidas, mientras que prácticamente se ignora a la máquina de estados finitos que es el control finito (es decir, el programa) de la máquina de Turing, considerando esta parte interna como una caja negra. Aquí es precisamente donde uno podría pensar una extensión de la máquina automática y Hasslacher lo identifica precisamente en estos términos, ya que:

⁵ Además de los trabajos del mismo Gurevich, Börger & Stark (2003) es un excelente recurso teórico sobre las ASM y sus aplicaciones posibles para el diseño de software.

(...) bajo ciertas condiciones bien definidas, se puede considerar que tal MSF [máquina de estados finitos] codifica una función unidireccional cuyo estado es imposible de invertir bajo el supuesto de tiempo P. Semejante control finito complejo no puede ser reducido a una MSF más simple con una cinta de entrada más larga, o bien hacerlo conllevaría recursos [en tiempo] NP. Esta es la estrategia básica que adoptamos para construir máquinas que están funcionalmente “cerca” de la estructura de sistemas complejos. Llamaremos máquinas de Turing extendidas o dispositivos Gödel (1) a tales máquinas, con la máquina de Turing convencional en el lugar Gödel (0) en esta escala (Hasslacher, 1988, p. 420).

Si bien esta dirección en la Hasslacher plantea una extensión de la máquina de Turing bien puede verse como una analogía computacional del trabajo científico, y de la práctica de generar modelos de fenómenos complejos del mundo natural –y se trata de una analogía muy fértil–, todavía es posible ver otra forma de extender la noción de máquina pensada como un elemento heurístico para la comprensión de un fenómeno que es todavía más general y cuyos orígenes pueden observarse en la obra del mismo Turing, considerando cómo el abordó el problema de modelar ciertos fenómenos naturales.

Términos y conceptos de máquina en la obra de Alan Turing

A continuación, realizaremos un breve recorrido por las expresiones utilizadas por Turing para referirse a diversos tipos de máquina. En la sección siguiente rescataremos los aspectos sustantivos que parecen estar supuestos en este uso. Este último paso estará guiado tanto por las expresiones como por el contexto en el cual han sido utilizadas.

Como ya hemos visto, la primera noción que presenta Turing es la de “máquina automática”, que incluye a la de “cómputo”. Pero en el texto de 1936 también se habla de “máquinas universales” –noción central para su concepción de computación–, de “máquinas de elección” (*choice-machines*), de “máquinas circulares y no circulares” (*circle-free machines*). Cada una de estas máquinas tenía el propósito de aclarar algún aspecto de las máquinas de computar. Por ejemplo, las máquinas

de elección serían una instancia de máquinas parcialmente determinadas o parcialmente mecanizadas.

En los años siguientes Turing comienza a utilizar diversas expresiones que evidencian una evolución en su estrategia modelizadora. Así, por ejemplo, para poner de manifiesto los límites de lo computable presenta a las “máquinas oráculo” (Turing, 1939), las cuales a través de algún tipo de artificio podrían solucionar problemas que vayan más allá de lo que puede hacer una máquina de computar. También comienza a evidenciarse en Turing una creciente preocupación por los límites de la implementación. Las “máquinas computacionales universales prácticas” (Turing, 1948), que mostrarían la diferencia entre una abstracción y su realización física, son una primera expresión de este cambio en sus intereses. Las distinciones entre “máquinas discretas” y “continuas” (Turing, 1948) podrían ser vistas desde esta última perspectiva. Además, las “máquinas de papel” (*paper machines*) mostrarían una focalización creciente en el problema de la programación de las máquinas.

A fines de los años '40 Turing intenta extender la estrategia de modelización a otros tipos de comportamiento humano, en particular aquellos señalados como inteligentes. Hay varios ejemplos de esta preocupación: las “máquinas-niño” (Turing, 1950), las “máquinas parcialmente aleatorias” (Turing, 1948), las máquinas desorganizadas (Turing, 1948), las máquinas modificables y automodificables (Turing, 1948), máquinas controladoras y máquinas activas (Turing, 1948).

Ejemplo de máquina extendida: Inteligencia mecánica

Veamos a continuación, a modo de ejemplo, una descripción tentativa de la forma en la cual Turing utilizaba las máquinas para abordar el problema de la modelización de la inteligencia. El trabajo de Turing en el que más claramente se puede notar esta forma de pensar a las máquinas es en la conferencia sobre el *Automatic Computing Engine* de 1947 para la Sociedad Londinense de Matemática, y en “Intelligent Machinery”, texto que consiste en un reporte escrito por Turing en 1948 para el *National Physical Laboratory* acerca de la posibilidad de que las máquinas puedan mostrar comportamiento inteligente.⁶ Es el mismo tema que enfrentará después en el famoso trabajo de 1950 “Computing

⁶ Como reporta Copeland (2004), hacia 1941 ya Turing había comenzado a desarrollar la idea de máquinas que piensan.

Machinery and Intelligence” (Turing, 1950). Sin embargo, el problema es atacado en estos primeros textos de una manera distinta. El reporte de 1948 señala, desde el principio, que la investigación se llevará a cabo utilizando la analogía de las máquinas con el cerebro humano como recurso guía. Esto lleva a Turing a clasificar las máquinas disponibles y concebibles de manera de determinar si alguna puede ser utilizada para lograr con ella lo que el cerebro es capaz de hacer. Lo importante en el uso de la analogía es que el cerebro no *necesita* ser, efectivamente, una máquina (si bien probablemente lo sea). Si *puede* ser entendido como una máquina, entonces en principio incluso una máquina *distinta* puede ser usada para lograr lo que el cerebro hace. Como lo señalaría Turing algunos años más tarde, la posibilidad que tiene el cerebro de realizar tareas que en principio parecen muy difíciles de lograr para cualquier máquina en un tiempo finito, es una razón para apostar por la modelización:

En mi opinión, este factor temporal es la cuestión que supondrá toda la dificultad técnica real. Si uno no supiera ya que estas cosas pueden ser hechas por los cerebros dentro de un tiempo razonable, uno pensaría que es inútil intentarlas con una máquina. El hecho de que el cerebro puede hacerlo parece sugerir que las dificultades pueden no ser tan malas como lo parecen ahora (Turing, 1952; Cooper 2013, p. 675).

La apuesta por la analogía en este caso, tiene como supuesto que el funcionamiento de los cerebros no cae fuera de los límites planteados por el funcionamiento de las máquinas. Evidentemente Turing no tiene manera de mostrar esta relación entre límites de funcionamiento, pero, como destacaremos en la sección siguiente, esta manera de pensar el problema y de abordarlo evidencia una estrategia de trabajo particular para enfrentarse con un fenómeno tan complejo.

Según la clasificación de máquinas, el cerebro es “probablemente una máquina continua y controladora, pero es muy similar a muchas máquinas discretas” (Turing, 1948; Cooper, 2013, p. 503).⁷ Determinar

⁷ Una máquina controladora, para Turing, es aquella que sólo maneja información. En este sentido contrasta con una “máquina activa” la cual produce algún efecto físico.

qué tan similares *pueden* ser es parte del objetivo de la investigación de Turing. Las máquinas que van a servir a este propósito son, principalmente, las máquinas desorganizadas, las “máquinas-niño” y las máquinas de papel. De estas máquinas, las “niño” –es decir, las máquinas que aprenden– serán centrales para la investigación de Turing.

La conferencia del 1947 es uno de las primeras referencias al problema de modelizar la inteligencia. Aquí las máquinas que aprenden y las máquinas-niños se constituyen en una guía para la elucidación de tipos de mecanismos apropiados. Hacia el final de la *lecture*, Turing escribe un breve apartado sobre inteligencia y máquinas en el que comienza parafraseando a Ada Lovelace: “Se ha dicho que las computadoras sólo pueden producir aquello para lo que han sido instruidas” (Turing, 1947; Cooper, 2013, p. 392). Así, todo aquello que no sea producido por una “instrucción” será visto como un error. Según Turing, las máquinas han sido utilizadas, hasta ahora, como “esclavas” y el trabajo que realizan parece que debe ser comprendido por sus usuarios desde el comienzo hasta el fin. ¿Qué ocurriría cuando la guía ahora es una máquina que aprende o una máquina-niño? ¿Sigue valiendo el *dictum* de Lovelace? Debemos comenzar, continúa Turing, por ver de qué manera se pueden modificar las instrucciones de tal forma que su resultado no previsto sea algo más que un mero error. De hecho, es precisamente éste el punto que se vuelve central en los trabajos posteriores sobre inteligencia, que elaboran sobre por qué una máquina de la que se espera que sea infalible no puede ser, a la vez, inteligente. Las máquinas niño deberían tener, además de disciplina, iniciativa. La iniciativa para Turing está relacionada con varios conceptos. Por un lado, está el aspecto ya citado de la posibilidad del error. Además, está la relevancia de la variabilidad del comportamiento y finalmente la manera en la cual se resuelven los problemas. A este último punto lo retomaremos en la próxima sección.

Una de las máquinas que más le permitirán acercarse al objetivo de la variabilidad del comportamiento son las máquinas desorganizadas –consideradas desde el punto de vista de observador, ya que el nivel de desorganización no es objetivo. Éstas son máquinas que se logran construir haciendo interactuar elementos estándares de una manera no sistemática, es decir, sin un propósito definido más allá del de meramente “ser máquinas”. Una de las preocupaciones básicas para Turing va a ser la manera en la cual se puede automatizar el proceso de organización de las máquinas. Aquí está operando bajo la guía de las máquinas-niño porque Turing ve al proceso de organización como

semejante al del de aprendizaje. A pesar de esto, se puede ver a las máquina-niño como una subclase de las máquinas desorganizadas: la subclase de dichas máquinas que ha sido reorganizada por interferencias apropiadas del medio para lograr un comportamiento específico, tal como el de “responder con reacciones definidas a ciertos comandos” (Turing, 1948; Cooper, 2013, p. 509). Es decir que, una vez definido un propósito, cualquier máquina desorganizada que tenga una complejidad mínima para modificar de alguna manera su configuración puede ser reorganizada por interferencias para producir dicho propósito. Probablemente la máquina resultante sea bastante distinta a la original, e incluya muchas conexiones redundantes, lo cual podría no ocurrir en una máquina diseñada *ad hoc* para cumplir el mismo rol.

Son varios los aspectos interesantes que encontramos en esta formulación de Turing. Por un lado, es claro el origen con la analogía del cerebro. Turing equipara a la corteza del cerebro de un niño con una máquina desorganizada que es organizada por los estímulos que recibe del medio. De hecho, no ve ninguna imposibilidad (más allá de las dificultades propias de lograrlo) de entrenar dicho cerebro para que se comporte como una máquina universal automática propiamente dicha. Lo que Turing presenta aquí es, efectivamente, la primera descripción en la literatura de lo que en la actualidad se conoce como *machine learning*, el tipo de máquinas o programas que puede aprender de su comportamiento pasado para mejorar el ajuste con una función de costo particular definida. La otra analogía a la que recurre Turing en esta formulación, aunque sin explicitarla, es claramente la de la selección natural, mecanismo que permite fijar mutaciones que se evalúan, a la larga, como “buenas”. Por otro lado, Turing es capaz de expresar con cierta claridad lo fundamental que es para que algo sea una máquina la integración entre los aspectos físicos y lógicos. Dada una arquitectura de una complejidad mínima para una máquina, su capacidad de generar comportamientos puede ser extendida mediante la adición de una memoria que tenga la suficiente velocidad de acceso para permitir que, en el tiempo de acción requerido, la parte netamente lógica o programable *compense* las deficiencias de la arquitectura propiamente dicha. Este balance entre arquitectura fija, compensabilidad por reprogramación y plasticidad general es lo que habrían logrado la evolución y la selección natural con el cerebro humano, y es lo que debería ser implementado de alguna forma si lo que se quiere es simular comportamiento inteligente. Uno de las características particulares de dicho comportamiento es

la posibilidad de resolver problemas. Este aspecto, sumado a lo que podríamos llamar su forma o estilo de modelización, permite percibir tanto una visión integral del trabajo de Turing desde sus primeros escritos como un cambio y evolución posterior.

Estilo de modelización y resolución de problemas

Ya desde la obra temprana se evidencia que para Turing abordar de manera ventajosa un problema equivale a construir una máquina que pueda modelizar el fenómeno en cuestión y la especificación de una estrategia de resolución de problemas. En este sentido, el desarrollo de sus mecanismos está asociado con una modificación en su manera de entender el proceso de modelización y con una forma de estructurar el proceso de resolución de problemas. Veamos ahora el primer aspecto. ¿Cómo entendía Turing el proceso de modelización a partir de sus mecanismos?

En primer lugar, podemos señalar la creciente preocupación de Turing por cuestiones de implementación:

He pasado un tiempo considerable de esta conferencia sobre este tema de la memoria ya que creo que la provisión de almacenamiento apropiado es el problema clave de la computadora digital y, sin duda, si han de ser persuadidas a mostrar cualquier clase de inteligencia genuina, se deberán proveer capacidades mucho más grandes que las disponibles en la actualidad (Turing, 1947; Cooper, 2013, p. 490).

Estas preocupaciones en cuestiones de implementación no necesariamente estaban relacionadas con cuestiones físicas. Hay al menos dos aspectos en los cuales se puede ver este punto. Primero, para Turing era importante la estructuración de la memoria a los fines de acelerar la velocidad de acceso a la misma:

Uno necesita alguna clase de memoria con la que se pueda alcanzar cualquier entrada [de registro] en un tiempo corto. Aparentemente esta dificultad solía preocupar a los egipcios cuando sus libros estaban escritos en rollos de papiro. Debe haber sido un trabajo muy lento buscar referencias en los mismos, y la manera

en la que en la actualidad ordenamos los datos impresos en libros es sin duda preferible. Podemos decir que el almacenamiento en una cinta y en rollos de papiro es un tanto inaccesible. Lleva un tiempo considerable encontrar una entrada determinada. La memoria en forma de libro es mucho mejor, y ciertamente es muy apropiada cuando ha de ser leída por el ojo humano. Podríamos incluso imaginarnos una máquina de computar que estuviera hecha para trabajar con una memoria basada en libros (Turing, 1947; Cooper, 2013, p. 487).

La estructura de la memoria aquí toma como modelo a la estructura de los dispositivos de lectura. La diferencia entre papiros y libros está vinculada con la velocidad de acceso y con las capacidades cognitivas humanas haciendo de la modelización un proceso que no se reduce meramente a la resolución de un problema técnico puntual.

En segundo lugar, la preocupación por la implementación no necesariamente física aparece asociada con las máquinas de papel. Turing estimaba que muchos problemas de la modelización por máquinas podían abordarse de manera satisfactoria por medio de la programación más que por el hardware. La expresión “máquina de papel” servía para destacar esta idea: “Es posible producir el efecto de una máquina de computar escribiendo una serie de reglas de procedimiento y pedirle a una persona que las lleve a cabo. Tal combinación de una persona con instrucciones escritas será llamada una ‘máquina de papel’” (Turing, 1948; Copeland, 2004, p. 416). Este acento está directamente vinculado con la importancia que Turing le va a dar a los programas en su construcción de computadoras. Sin embargo, más que una oposición a las soluciones de hardware –lo cual no era el caso como vimos a propósito de la velocidad de memoria– este interés puede ser visto, en algunos contextos, como una manera de enfatizar la relevancia del diseño y de la organización de las máquinas –esto es, de su arquitectura–. Como podemos notar, para Turing es claro que la modelización adecuada de un fenómeno no implica un compromiso ontológico con la naturaleza de aquello que es modelado. El cerebro puede ser visto como una computadora digital a pesar de que probablemente sea un sistema continuo.

Ahora bien, el interés por la implementación también va a tener en Turing una faceta vinculada con el hardware. Cuando él comienza a

trabajar en el diseño de un “motor automático de computación” (ACE por sus siglas en inglés) a partir de 1946, su interés central en dichos trabajos es la implementación de estructuras físicas que permitan aumentar la velocidad y la capacidad (la memoria) de la computadora. Este interés llegaba al punto de proponer soluciones específicas acerca del tipo de hardware que podía ser utilizado. Al respecto comenta Copeland que:

De acuerdo con Wilkinson en 1955, Turing “fue el primero en darse cuenta de que era posible superar las dificultades del tiempo de acceso [a la memoria] con [...] líneas de mercurio [...] o con tambores magnéticos [*drum stores*] proporcionando una cantidad [de memoria] comparativamente menor [pero] de rápido acceso. Muchas de las máquinas comerciales en Estados Unidos y en este país hacen uso de este principio” (Copeland, 2004, p. 366).

En relación con el segundo aspecto que señalamos más arriba, la manera en la cual Turing entendía la relación entre máquinas y resolución de problemas, la comprensión de esta actividad como búsqueda es central. Al parecer, la manera más directa en la cual se podía encontrar un mecanismo para la resolución de problemas era entender este proceso como la búsqueda de una solución en un espacio de posibilidades. Cuando Turing describe los pasos que entrarían en juego en el momento en que una máquina inteligente intenta resolver un problema, uno está tentado de leer –y creemos que es correcto caer en dicha tentación– los mismos pasos que Turing utiliza para resolver un problema:

En la práctica, cuando se resuelven problemas de esta clase uno probablemente aplicará alguna clase de “transformación” muy compleja del problema original, lo que implica una búsqueda a través de muchas variables, algunas más análogas a la original, algunas más bien parecidas a una “búsqueda por todas las pruebas”. Las investigaciones posteriores acerca de la inteligencia de las máquinas probablemente se involucren en gran medida con las “búsquedas” de esta clase. Quizás podamos denominar tales búsquedas como “búsquedas intelectuales”. Se las puede definir en

pocos términos como “búsquedas de combinaciones con propiedades particulares llevadas a cabo por cerebros” (Turing, 1948; Cooper, 2013, p. 516).

Podríamos leer los pasos de dicha búsqueda intelectual de la siguiente manera: identificar un problema, transformarlo en alguno que sea más fácil de trabajar –es decir, crear un modelo del mismo– y explorar sus distintos comportamientos hasta obtener una solución adecuada para los propósitos que llevaron al problema. Turing ve directamente en su “búsqueda intelectual” una aplicación análoga del mecanismo de la selección natural, a la que ahora puede llamar “búsqueda genética”. Lo que también ve es la dificultad de aplicar una búsqueda intelectual sin un punto de partida y sin interacción con un medio ya rico en métodos y resultados. Cualquier búsqueda comienza con un conocimiento de fondo, que proviene de la capacidad interactiva con un entorno. Cualquier comportamiento intelectual es pues parte de una “búsqueda cultural” más grande, la de la comunidad como un todo (Turing, 1948; Cooper, 2013, p. 516).

Regresemos ahora a un aspecto que destacamos en varias partes de este trabajo: la relación entre la tarea de modelización y postulación de semejanza supuesta. El caso que presentamos en la sección anterior es el de la comprensión de la inteligencia en términos computacionales. Turing parece haber utilizado una estrategia habitual en ciencia: suponer como hipótesis de trabajo o como guía heurística la semejanza entre procesos que en principio parecen muy diferentes. El éxito en la capacidad descriptiva –y eventualmente predictiva– del modelo suele ser un aliciente para continuar en esa dirección. Ese parece ser el contexto mínimo –aunque quizás no el único– para comprender las afirmaciones ontológicas que Turing hace en relación con el cerebro y la mente. De esta manera, se podría describir el trabajo de Turing no tanto como intentando responder a la pregunta qué es el cerebro –o la mente– sino de qué manera se puede elaborar y desarrollar un programa de trabajo en relación con la inteligencia. No es que la cuestión ontológica sea irrelevante, sino más bien que ésta parece cumplir, en la perspectiva modelizadora de Turing, un rol asociado con el desarrollo de un plan de trabajo.

Consideraciones finales

En el presente trabajo hemos presentado un listado de la mayoría de los términos utilizados por Turing desde 1936 hasta 1952 para referirse a tipos de máquina más allá de la máquina automática. Estimamos que estos términos estaban asociados con una modificación y un eventual enriquecimiento de la concepción misma de máquina.

Entre las objeciones que se pueden presentar a esta propuesta hay una que sugeríamos ya la introducción de este trabajo; a saber, que finalmente todas las supuestas “innovaciones” se reducen en principio a la máquina automática propuesta en 1936. Responder a esta objeción implica explicitar el sentido y alcance de la extensión que sugerimos que hace Turing. Los tipos de comportamiento que en principio están supuestos por la operación de la máquina automática –y en particular de la máquina universal– no necesariamente solucionan las situaciones problemáticas presentadas por el propio Turing como un desafío. En este sentido, si bien estas “extensiones” pueden ser vistas, desde una perspectiva “en principio” como “explicitaciones” no obstante, desde la perspectiva de la comprensión del alcance y límites de la modelización computacional, están lejos de ser triviales. Y así parece haberlo entendido Turing, como lo muestra el tiempo y el esfuerzo dedicados en su vida y en su obra a la caracterización de estos tipos de máquina.

De manera esquemática podríamos sintetizar la extensión de la noción de máquina en Turing a partir de las siguientes dimensiones. En primer lugar, hemos señalado el vínculo entre máquinas y problemas puntuales. En segundo lugar, especificamos que, si bien en los primeros trabajos ya se habla de “máquinas”, el peso allí está puesto en la capacidad *abstracta* de trabajo, mientras que en los trabajos posteriores la capacidad y la factibilidad –a veces *física*– de las máquinas cobrarían un rol protagónico. En tercer lugar, entre la década del ‘30 y el trabajo posterior hay una continuidad directa en términos de un estilo de modelización que propone máquinas para abordar problemas. Este estilo de modelización explica la búsqueda posterior de Turing al especificar *tipos* de máquinas. Es el tipo de problema abordado en cada caso el que da cuenta de la necesidad de proponer máquinas particulares. En cuarto lugar, la concepción de resolución de problemas aparece asociada con un modo mecánico de búsqueda. Finalmente queremos

poner en primer plano la actitud epistémica, evidenciada incluso en su propuesta de 1936, en el cual dar cuenta de un fenómeno equivale a “construir una máquina”. Esta actitud, constante durante toda su vida intelectual, muestra una manera particular de comprender el proceso de modelización.

Agradecimientos

Agradecemos las sugerencias de los revisores anónimos de la revista que ayudaron a mejorar sustancialmente algunas secciones del trabajo. Esta investigación se realizó en marco de una beca doctoral de CONICET y del PICT 2016-1524 del Fondo para la Investigación Científica y Tecnológica (FONCyT), República Argentina.

Referencias

Las obras de Alan Turing

- ___ (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. (*Proc. Lond. Math. Soc.* (2) 42, 230–265).
- ___ (1939). Systems of Logic Based on Ordinals. (*Proc. Lond. Math. Soc.* (2) 45, 161–228).
- ___ (1947). Lecture to the London Mathematical Society.
- ___ (1948). Intelligent Machinery (Informe de Turing para el National Physical Laboratory).
- ___ (1950). Computing Machinery and Intelligence (*Mind*, 59, 433–460).
- ___ (1951a). Can Digital Computers Think? (BBC Third Programme radio broadcast (15 May 1951), transcript edited by B. J. Copeland).
- ___ (1951b). Intelligent Machinery: A Heretical Theory (Lecture given to *51 Society* in Manchester (c. 1951), transcript edited B. J. Copeland).
- ___ (1952). Can Automatic Calculating Machines Be Said To Think? (Broadcast discussion, BBC Third Programme (14 and 23 Jan. 1952), transcript edited B. J. Copeland)

Las referencias a las mismas citadas en el cuerpo del trabajo corresponden a la paginación de las compilaciones efectuadas por Copeland y Cooper:

- Cooper, S. B., y Leeuwen, J. van. (2013). *Alan Turing: His Work and Impact*. Waltham-Kidlington: Elsevier Science.
- Copeland, B. J. (2004). *The Essential Turing*. Oxford: Clarendon Press.

Bibliografía secundaria

- Börger, E., & Stärk, R. (2012). *Abstract State Machines: A Method for High-Level System Design and Analysis*. Berlín: Springer Science & Business Media.
- Church, A. (1937). On Computable Numbers, with an Application to the *Entscheidungsproblem*. *The Journal of Symbolic Logic*, 2(1), 42-43.
- Copeland, B. J., & Sylvan, R. (1999). Beyond the Universal Turing Machine. *Australasian Journal of Philosophy*, 77(1), 46-66.
- Gandy, R. (1980). Church's Thesis and Principles for Mechanisms. En J. Barwise (ed.), *The Kleene Symposium*. (pp. 123-148). Ámsterdam-Nueva York: North-Holland.
- Gurevich, Y. (2018). Evolving Algebras 1993: Lipari Guide. *arXiv:1808.06255 [cs, math]*. Recuperado de <http://arxiv.org/abs/1808.06255> (Trabajo original publicado en 1995).
- Gödel, K. (1990). *Collected Works. II: Publications 1938-1974*. S. Feferman, J. Dawson, S. Kleene, G. Moore, R. Solovay y J. van Heijenoort (eds.), Oxford: Oxford University Press.
- Hasslacher, B. (1988). Beyond the Turing Machine. En R. Herken, (ed.), *The Universal Turing Machine: A Half-Century Survey*. (pp. 417-433). Oxford: Oxford University Press.
- Herken, R. (1988). *The Universal Turing Machine: A Half-Century Survey*. Oxford: Oxford University Press.
- Piccinini, G. (2007). Computing Mechanisms. *Philosophy of Science*, 74(4), 501-526.
- Shagrir, O. (2002). Effective Computation by Humans and Machines. *Minds and Machines*, 12(2), 221-240.
- Shapiro, S. (2006). Effectiveness. En S. Rahman & J. Symons (eds.), *The Age of Alternative Logics*. (pp. 37-49). Dordrecht: Springer.
- Sieg, W. (1994). Mechanical Procedures and Mathematical Experience. En A. George (ed.), *Mathematics and Mind*. (pp. 71-117). Oxford: Oxford University Press.
- ____ (2002). Calculations by Man & Machine: Mathematical Presentation. En P. Gärdenfors, J. Wolenski y K. Kijania-Placek (eds.), *In the Scope of Logic, Methodology and Philosophy of Science: Volume One of the 11th International Congress of Logic, Methodology and Philosophy of Science, Cracow, August 1999*. (pp. 247-262). Dordrecht: Springer.

