Capítulo 1

El Programa General

1.1. Introducción

La Inteligencia Artificial (AI) se inicia como una disciplina académica y de investigación científica con la publicación del artículo *Intelligence and Computing Machinery* (Inteligencia y Maquinaria Computacional) por Alan Turing en 1950 [1].[1] En dicho artículo se pregunta por primera vez en un foro formal si las máquinas pueden pensar. Esta forma de hablar es cotidiana en la actualidad, pero en aquellos años se pensaba que sólo los seres humanos podían tener la facultad del pensamiento y la pregunta resultaba realmente extraordinaria. La implicación es por supuesto que si se puede pensar se es o se puede ser inteligente.

Turing enfatiza que las palabras "máquina" y "pensar" son problemáticas. Es muy claro que por máquina se refiere a la computadora digital en oposición al ser humano, que no considera como máquina. Su propuesta original fue construir máquinas capaces de simular los procesos de la mente pero no que el cerebro fuera una computadora y la mente un progra-

[1] Turing no nombró así a la disciplina y la denominación "Inteligencia Artificial" surgió en el *Dartmouth Summer Research Project on Artificial Intelligence* que se llevó a cabo en el verano de 1956 en el Dartmouth College, en Hannover, New Hampshire, cuando Turing ya había fallecido, por iniciativa de John McCarthy junto con Marvin Minsky, Nathaniel Rochester y Claude Shannon, y a la que asistieron Herbert Simon y Allen Newell, entre otras personalidades; sin embargo, hay un consenso muy amplio en la comunidad que la propuesta original y el programa de investigación corresponden a la propuesta original de Turing.

ma de cómputo. Asimismo, la computadora es un tipo especial de máquina que se distingue cualitativamente de las máquinas estándar, como los coches y los aviones.

Para comprender mejor a la máquina en cuestión, en la Sección 1.2 se revisa el modelo teórico general de las computadoras digitales, el cual se presentó originalmente en 1936 en el artículo "Acerca de los números computables, con una aplicación al problema de decisión" (*On Computable Numbers, with an Application to the Entscheidungs problem*) también de Turing [2]. Este artículo se considera la piedra fundacional de las Ciencias de la Computación y al modelo propuesto se le denominó posteriormente "La Máquina de Turing" (*The Turing Machine*).[2]

La palabra "pensar", por su parte, tiene una amplia gama de connotaciones lingüísticas, filosóficas, psicológicas, culturales, históricas, etc., y Turing plantea abordarla desde una perspectiva práctica. Para este efecto propone analizar su significado en relación al Juego de Imitación (*Imitation Game*), mejor conocido como La Prueba de Turing (*The Turing Test*) [1], cuyas reglas presuponen que para ganar es necesario pensar, por lo que se tiene que considerar como pensante a quien gane el juego, ya sea éste un ser humano o una máquina. Por estas razones se puede decir que el primer objetivo particular de la Inteligencia Artificial fue construir una Máquina de Turing capaz de pasar la Prueba de Turing o de ganar el juego de imitación. Esta discusión se aborda en la Sección 1.3.

Turing es consciente de que su propuesta puede resultar inaceptable para mucha gente y él mismo plantea y refuta un conjunto de objeciones posibles. Estos argumentos inician la discusión que dio lugar posteriormente a la Ciencia Cognitiva que pregunta no sólo si podemos construir máquinas pensantes sino también si el cerebro y la mente se pueden conceptualizar como fenómenos computacionales naturales, en oposición a las máquinas y procesos computacionales inventados por los seres humanos. La argumentación de Turing y sus repercusiones posteriores se abordan de manera inicial en la Sección 1.4 y serán un tema recurrente a lo largo de este texto.

[2] Una guía del artículo comentado línea por línea incluyendo el contexto histórico y anecdótico se presenta en el libro *The Annotated Turing* por Charles Petzold, 2008 [3].

Una vez establecidos estos preliminares Turing se aboca a presentar de manera general los lineamientos generales de la Inteligencia Artificial. Propone que es posible representar conocimiento en las computadoras y que éste se puede utilizar para emular funciones mentales mediante programas de cómputo. Para efectos de dotar de conocimiento a la máquina hay dos vías: dárselo a través de la interacción con los usuarios humanos y construir máquinas que aprendan a partir de su interacción con el entorno. Estas dos vías son ambas necesarias y complementarias. Esta discusión se revisa con más detalle en la Sección 1.5. Turing concluye dicha presentación con dos propuestas prácticas para desarrollar el programa: 1) construir una máquina capaz de jugar ajedrez, como el paradigma de pensar; y 2) construir una máquina capaz de entender el inglés, es decir el lenguaje natural, que es la materia de la Prueba de Turing propiamente. Estas dos tareas generales han sido objeto de investigación continua desde entonces, y una gran variedad de las investigaciones puntuales se pueden articular en relación a estas dos tareas paradigmáticas. Este es el origen del proceso muy rico e intenso que se llevó a cabo desde entonces y que continúa hasta la actualidad. Al final del artículo de 1950 Turing plantea explícitamente el objetivo general de la Inteligencia Artificial: construir una máquina capaz de igualar e incluso superar las competencias mentales de los seres humanos.

1.2. La Máquina de Turing

El modelo teórico general de las computadoras digitales se presentó originalmente por Alan Turing en 1936 [2] y desde entonces nos referimos al mismo como la *Máquina de Turing* (MT). Las calculadoras creadas previamente, pasando por las de Pascal, presentada en 1642, y Leibniz, desarrollada entre 1671–1694, así como la sumadora patentada por Burroughs en 1888 y la máquina de tarjetas perforadas patentada por Hollerith en 1889, utilizada ya en el censo de los Estados Unidos de 1890, se pueden considerar como computadoras de propósito particular orientadas a las operaciones aritméticas básicas y a registrar opciones binarias; y la Máquina Analítica de Babbage, desarrollada desde 1837 hasta su muerte en 1871, aunque nunca se construyó completamente, se podía reconfigurar para hacer diversos cálculos, y se considera que era ya una computadora de propósito general; pero no fue sino hasta la

introducción de la MT cuando estuvo disponible un modelo teórico completamente abstracto que describiera a las computadoras digitales de forma independiente de sus diseños y construcciones particulares.

Los elementos constitutivos de la MT son una cinta dividida en celdas, como un renglón de una hoja de papel cuadriculado. Este sustento material se denomina aquí *el medio* de la computación. En cada celda se lee o escribe un símbolo de un alfabeto, como los numerales[3] del cero al nueve o las letras de la "a" a la "z".[4] Se utiliza para este efecto un escáner que corresponde a un lápiz con goma. Con éste se apunta a una celda a la vez y se lee o sustituye su contenido. Adicionalmente, el escáner se puede mover a la izquierda o a la derecha una sola celda a la vez.

Además de los elementos materiales, la máquina tiene un control con un conjunto finito de estados que dirige al escáner y las operaciones sobre la cinta; dicho control es discreto y dado un estado y el símbolo que se inspecciona en la cinta, selecciona una operación, la lleva a cabo y cambia a otro estado. El control se puede especificar como una tabla con una columna para cada símbolo del alfabeto y un renglón para cada estado, y la operación que se realiza cuando se inspecciona un símbolo en un estado se codifica en la celda en la que que se intersectan el renglón y la columna respectivas.

El control se puede pensar como el proceso de la mente que guía a la mano que sostiene al lápiz. La máquina inicia su trabajo en un estado inicial designado y hay un subconjunto de estados llamados "de paro" tales que si la máquina llega a uno de éstos se detiene y la computación termina. Toda computación consiste en partir del estado inicial con el escáner inspeccionando una celda dada, y realizar las operaciones que se especifican en la tabla de transición hasta llegar a un estado final. El trabajo de la máquina se limita a sustituir la cadena de símbolos en la cinta, *la entrada,* por la cadena que queda al final, *la salida*.

[3]Un numeral es el nombre de un número en oposición al objeto matemático que tiene un carácter abstracto (e.j., el numeral "9" es el nombre del número 9).

[4]Los símbolos del alfabeto son *tipos*, y los que se escriben en la cinta son *instancias* de dichos tipos; las instancias son "perfectas", como los tipos de una máquina de escribir o los *fonts* de los editores de texto, y cada instancia cabe exactamente en una celda. Si los símbolos son caligráficos se descartan las diferencias individuales.

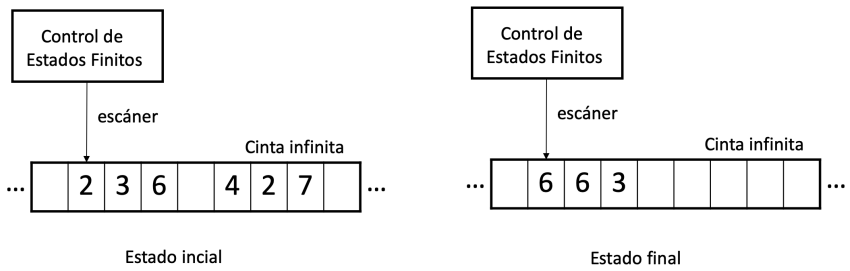

Figura 1.1: Máquina de Turing

La Figura 1.1 ilustra los elementos de la MT al inicio y al final de una computación –los diagramas a la izquierda y la derecha respectivamente. El algoritmo se codifica en la tabla de transición del control de estados, en este caso el algoritmo de la suma aritmética; los argumentos se representan como cadenas de símbolos en el estado inicial –las cadenas "236" y "427" separadas por un espacio en blanco– y el valor en el estado final –la cadena "663". El escáner apunta al símbolo más izquierdo en ambos estados.

Para comprender la máquina de manera intuitiva se puede generalizar a dos dimensiones substituyendo la cinta por un papel cuadriculado e incluyendo las operaciones para mover el escáner una celda hacia arriba y hacia abajo. Supóngase que se quiere diseñar una máquina para sumar dos números decimales, donde los sumandos están uno sobre otro y alineados por la derecha, como el algoritmo elemental que se enseña en la escuela primaria en México. En el estado inicial el lápiz está sobre el símbolo en el extremo derecho del sumando superior y la operación es leer dicho símbolo, mover el lápiz hacia abajo y cambiar a un estado que "recuerde" el símbolo leído; en el nuevo estado el lápiz apunta al símbolo más a la derecha del segundo sumando, y la operación consiste nuevamente en leer dicho símbolo y moverse hacia abajo, donde hay una celda en blanco, y pasar a un nuevo estado que "recuerda" la ruta seguida; en éste se escribe el símbolo que corresponda a la tabla elemental de la suma.

Por ejemplo, si los números sumandos superior e inferior son "236" y "427" respectivamente, en el tercer estado el lápiz quedará abajo del "7" y se escribirá un "3". Posteriormente el escáner se mueve para apuntar el símbolo superior de la siguiente columna hacia la izquierda y se repite el proceso de manera recurrente, considerando que puede haber "acarreos" de

la columna anterior (e. j., si los dígitos son "236" y "427" se suman seis y siete, se escribe un tres y se lleva uno). Se invita al lector a diseñar la tabla de transición para la suma aritmética, para números representados por una cadena finita de dígitos. Esta tabla especifica el *algoritmo* que se enseña a los niños para sumar en la escuela primaria, y la computación desde el estado inicial hasta el final se puede considerar como un modelo del proceso mental que realiza el niño cuando suma. Los elementos materiales y conceptuales de este proceso motivan claramente el diseño de la máquina y el concepto de computación de Turing.

Turing pensó que la dimensión del medio es contingente y escogió la cinta lineal y no la cuadrícula, pero los elementos conceptuales son los mismos. Sin embargo, comparar los dos formatos permite ilustrar tanto los aspectos conceptuales del modelo ideal como los prácticos de las máquinas físicas.

1.2.1. Consideraciones teóricas

Desde el punto de vista teórico toda MT evalúa una función matemática. En este sentido una función es una relación entre dos conjuntos, el primero llamado dominio y el segundo codominio, en la que cada elemento del dominio se relaciona a lo más con un miembro del codominio. Los elementos de dichos conjuntos se designan respectivamente como los *argumentos* y los *valores*. Si todos los argumentos están asociados a un valor la función es *total* pero puede haber argumentos para los que no se defina un valor en cuyo caso la función es *parcial*.

No existe ninguna restricción para los miembros de ambos conjuntos –pueden ser objetos físicos, como los animales y los coches, abstractos, como los números, los conjuntos y las propias funciones, o conceptuales como los objetos de conocimiento o conceptos que residen en la mente– siempre y cuando se puedan concebir como objetos individuales.

Se asume que la cinta es infinita y siempre hay celdas adicionales disponibles por los dos lados; asimismo, que las operaciones se hacen con una velocidad infinita, por lo que la cadena de símbolos de salida se escribe inmediatamente una vez que se inicia la computación. Una consideración adicional es que, dado su carácter ideal, la máquina nunca falla.

Además de los elementos físicos y conceptuales de la máquina se requiere asignar un significado a las cadenas de símbolos que se escriben en el medio. Para este efecto se establecen las llamadas *Convenciones de Interpretación* [4]. Hasta ahora se han utilizado de manera implícita pero es necesario hacerlas explícitas. Por ejemplo, la estipulación que las cadenas al inicio y al final de la computación representan a los argumentos y al valor de la función, respectivamente.

Además se requiere establecer *la notación*: el código o lenguaje de las cadenas en la cinta. En la Figura 1.1 se ha asumido hasta este momento que la notación es *decimal* pero esto es una contingencia, pues las cadenas podrían ser numerales especificados en cualquier base. En general las mismas cadenas reciben diferentes interpretaciones en relación a diferentes notaciones. Por ejemplo, la cadena "111" se interpreta como tres en notación monádica,[5] siete en binaria y ciento once en decimal.

El modelo exige adicionalmente incluir una referencia convencional para manipular las cadenas de símbolos en el medio. Por ejemplo, en la llamada *configuración estándar* en el formato lineal [4] se exige que al inicio y al final de la computación el escáner inspeccione la celda que contenga el símbolo más a la izquierda de las cadenas que representan al argumento y al valor respectivamente, y que todas las demás celdas están en blanco.[6] Por supuesto, el proceso computacional tiene que inspeccionar todos los argumentos para que la computación no sea trivial.[7]

Aunque el objeto de la computación, es decir la función matemática, es un objeto abstracto y se puede concebir independientemente de una representación, los algoritmos se diseñan en relación a una notación, a un medio, a las convenciones de interpretación y, en particular,

[5] La notación monádica incluye solo el símbolo "1" en su alfabeto y la cadena de 1s de longitud n se interpreta como el número n.

[6] El contenido de las celdas en blanco se piensa convencionalmente como un símbolo especial de todo alfabeto –"el símbolo vacío"– que se lee y escribe de manera estándar y cuando se escribe "pone" en blanco a la celda. Asimismo, es el contenido inicial de todas las celdas, cuando la cinta o la cuadrícula "están en blanco".

[7] No se vale, por ejemplo, que para evaluar la función de identidad, que tiene por dominio y codominio al mismo conjunto, y la relación funcional establece que cada objeto se relaciona consigo mismo, se pase directamente del estado inicial al final al inicio de la computación, a pesar de que el valor de la función sería el correcto.

a una configuración estándar. Aunque éstas son contingentes, la especificación e implementación de una algoritmo requiere siempre una elección particular.

La convención de que toda máquina computa una función junto con la definición de la configuración estándar tiene una implicación directa en la teoría de la computabilidad. Esta teoría tiene por objetivo definir un modelo o mecanismo para avaluar a todos los argumentos de todas las funciones. Si este objetivo se lograra, dicha teoría sería completa. Sin embargo hay funciones que no se pueden computar, las cuales se conocen como funciones *no computables*. Para apreciar esta distinción considere que toda computación tiene tres posibles conclusiones con sus respectivas consecuencias:

1. Si la máquina se detiene o *para* en la configuración estándar, la cadena en la cinta representa al valor de la función.

2. Si la máquina se detiene en una configuración diferente de la estándar, la función no tiene valor para el argumento dado; en este caso la función es parcial.

3. Si la máquina no se detiene, la función no tienen valor para el argumento dado y la función es parcial.

El problema es que para satisfacer los tres casos se tendría que contar con una MT *h*, la Máquina de Paro (*the Halting Machine*) que tuviera como argumentos la descripción o nombre de la MT en cuestión *m*, y el argumento particular *n* para el que se desea saber si dicha máquina se detiene o no se detiene *h(m, n)*. Si se contara con la máquina *h* se sabría si la máquina bajo investigación se detiene y se caería en 1) o 2); o no se detiene, y se caería en 3), pero para este efecto *h(m, n)* tendría que detenerse para todo *m* y *n*, es decir, siempre. Se sabe, sin embargo, que no es posible diseñar una MT (e.j. [4]) que resuelva el problema del paro (*the halting problem*). Por lo mismo, la función *h* es un ejemplo de función no computable.

Por otra parte, si se considera que la MT trabaja a velocidad infinita y cuando se detiene lo hace instantáneamente, éste no sería un problema para un ser omnisciente, quien tendría conocimiento completo de todas las funciones, y no se puede descartar que haya otra forma de computar, alternativa a la MT, que pueda resolver el problema del paro. Sin embargo, esta

pregunta ha estado abierta desde su formulación original [5] y a la fecha no se ha encontrado dicha máquina, y las posibilidades de encontrarla parecen ser muy remotas.

La investigación de formalismos para caracterizar el conjunto de todas las funciones así como de mecanismos para computarlas ha sido muy activo. Tres de éstos son la teoría de las funciones recursivas, la teoría de las funciones ábacus o máquinas de registro, que son el modelo teórico de la arquitectura de Von Neumann, y la propia MT. Sin embargo, todas estas formas de computación se pueden traducir a la MT y vice versa por medio de un proceso reductivo puntal. Es decir, dada una función expresada en cualquiera de estos formalismos se puede encontrar la MT que computa a la misma función y vice versa, para todas las funciones computables. Los procedimientos para traducir entre sí a las Máquinas de Turing, las funciones recursivas y las máquinas ábacus se muestran de manera concisa y elegante por Boolos y Jeffreys [4]. Otro formalismo equivalente desarrollado por Alonso Church, quien fuera el director de la tesis doctoral de Turing, es el *cálculo-λ*; en éste se basa el lenguaje de programación *Lisp* –por *List Processing*– desarrollado por John McCarthy a finales de la década de los cincuenta y posiblemente el lenguaje de programación más popular en la historia de la Inteligencia Artificial. La correspondencia entre la MT y el *cálculo-λ* fue demostrada parcialmente por el propio Turing en un apéndice al artículo original y posteriormente de forma rigurosa por él mismo y por Kleene.

Por estas razones hay una corriente de opinión muy sólida en Ciencias de la Computación que sostiene que: 1) La Máquina de Turing computa el conjunto completo de las funciones computables; 2) todo formalismo computacional suficientemente general es equivalente a la MT; y 3) este conjunto corresponde con el conjunto de funciones que pueden ser evaluadas intuitivamente por los seres humanos. Esta hipótesis se conoce como la Tesis de Church o la Tesis Church–Turing. En su versión más fuerte, la tesis establece que la MT es el formalismo computacional más poderoso que puede existir en cualquier sentido posible.

Una consideración teórica adicional que aparece ya en el artículo de Turing de 1936 es la definición de la Máquina Universal. La tabla de transición de cada MT codifica un algoritmo que computa a una función particular, pero Turing planteó que dicha tabla también se puede especificar como una cadena de símbolos, poniendo en secuencia cada uno de sus

tuplos <*estado actual, símbolo escaneado, operación, siguiente estado*>. Esta secuencia es una representación del programa que computa la función correspondiente. Con base en esta observación, Turing propuso una Máquina con un control que inicialmente lee o *carga* (*up load*) la tabla de transición y configura dinámicamente una máquina particular. Como todas las tablas de transición se pueden poner en dicho formato, la Máquina Universal puede computar todas las funciones. La Máquina Universal es programable y es el modelo teórico de todas las computadoras digitales.[8]

Una consideración final enfatizada por Turing es el carácter determinístico de la máquina: dado que computar es evaluar una función, y la relación entre los argumentos y los valores está definida de antemano, el resultado de toda computación está predeterminado por necesidad. Turing afirmó que el determinismo de la máquina supera al concebido por Laplace. Esto resulta confuso ya que las computadoras digitales pueden modelar procesos estocásticos en los que el argumento está relacionado con más de un valor el cual se puede escoger de manera aleatoria; y también existen los autómatas no-determinísticos, en los que la tabla de transición es una relación y no una función, y para un estado y un símbolo de entrada puede haber más de un siguiente estado. Sin embargo, las relaciones se pueden componer por un conjunto de funciones, las cuales son el objeto de la computación propiamente, y ambas clases de no-determinismo se diluyen en esta reducción.

1.2.2. Consideraciones prácticas

En oposición a la máquina ideal, las máquinas reales cuentan con recursos finitos de memoria y velocidad de cómputo. Éstos dependen del tipo de tecnología empleada en su construcción y aunque la velocidad y capacidad de memoria de las máquinas actuales es muy significativa, hay limitaciones que deben tomar en cuenta en el diseño de algoritmos prácticos.

La primera consideración es que puede haber una gran variedad de algoritmos para computar la misma función. Por ejemplo, la definición del algoritmo para sumar es más sencilla si

[8] El concepto corresponde al de *programa almacenado* que se atribuyó posteriormente a Von Neumann por la muy controvertida publicación del *First Draft of a Report on the EDVAC* pero la idea original es de Turing.

se utiliza la cuadrícula en vez de la cinta lineal. Asimismo, los pasos que se requieren para completar el cálculo son mucho menos con el medio cuadriculado, como se puede verificar diseñando ambos algoritmos utilizando la notación decimal. En general la geometría del medio importa porque determina las trayectorias que tiene que seguir el escáner para llevar a cabo el proceso. Adicionalmente, la notación tiene un impacto muy significativo en la definición de algoritmos. La monádica es la más simple para definir las funciones sucesor, suma e identidad, como se puede verificar diseñando los algoritmos correspondientes.

Por su parte, la configuración estándar se requiere no sólo para interpretar a las cadenas sino también para concatenar o acoplar computaciones; por ejemplo, para permitir que el valor de una computación sea el argumento de la siguiente, para lo cual es también necesario que el estado final de la primera corresponda con el inicial de la siguiente. La necesidad de establecer una configuración estándar impacta también en el diseño de algoritmos, que se puede dividir en dos partes: el procedimiento para evaluar la función propiamente y los procedimientos auxiliares para asegurar que las computaciones empiecen y terminen en la configuración estándar. La arquitectura Von Neumann y la memoria de acceso random (RAM), que asigna una dirección a cada localidad de memoria, permitieron la definición de algoritmos prácticos, y de ahí su utilidad.

En computaciones prácticas se requiere también determinar el número de operaciones o pasos computacionales y la cantidad de localidades de memoria necesarias para llevar a cabo la computación. Dado que estos números pueden crecer de manera muy rápida, fácilmente se podría llegar a cifras extraordinarias, y el cómputo requeriría miles o millones de años en las computadoras más rápidas que existen en la actualidad, por lo que es necesario cuantificar de antemano estos parámetros. Para este efecto es importante distinguir a la teoría de la computabilidad de la teoría de la complejidad algorítmica; mientras que la primera asume que se cuenta con recursos infinitos de memoria y las computaciones se hacen instantáneamente, la segunda trata de los algoritmos que se pueden computar de manera efectiva. La teoría de la complejidad permite abstraer hasta cierto punto sobre los medios, notaciones y configuraciones, y determinar la complejidad en términos de la forma de las funciones directamente, pero de cualquier forma es conveniente tener presente los elementos involucrados

en la definición de algoritmos y su relación a las máquinas prácticas. Esto es particularmente relevante si el objetivo no es sólo realizar cálculos matemáticos complejos sino modelar las funciones de la mente.

1.2.3. Consideraciones interpretativas

Una consideración final es que la máquina opera sobre formas y su trabajo consiste en transformar representaciones: tan sólo manipula símbolos mediante el escáner de manera local y nunca tiene acceso a toda la cinta. Su definición establece la geometría del medio y el alfabeto, pero no la notación ni la configuración, que son implícitos en la especificación de los algoritmos. Asimismo, los contenidos o interpretaciones le son ajenos y residen sólo en la mente de los intérpretes humanos. Por lo mismo, la máquina no sabe cuál es la función que se representa en la tabla de transición o se computa por el algoritmo. No sabe tampoco que las cadenas en la entrada y la salida representan al argumento y al valor de la función respectivamente. Las computadoras, como cualquier otra máquina, no saben nada ni son conscientes del trabajo que hacen o la información que procesan para el consumo humano.

Sin embargo, la inteligencia artificial resta que en que el conocimiento se puede representar en computadoras digitales y que *razonar* o, de manera más general, *realizar inferencias*, son procesos computacionales o algorítmicos que transforman a las representaciones. Consecuentemente, que todo objeto de conocimiento, incluyendo las habilidades perceptuales y motoras, se puede representar a través de funciones matemáticas, bajo un conjunto apropiado de convenciones de interpretación, y que la ejecución de los algoritmos correspondientes en su maquinaria computacional es causal y esencial a la conducta del agente.

1.3. La Prueba de Turing

Pasamos ahora a analizar la prueba operativa propuesta por Turing para considerar a una máquina *pensante*: si la computadora puede conversar en lenguaje natural[9] con un ser

[9]La oposición es entre el lenguaje humano, como el español o el inglés, y los lenguajes formales, como los lenguajes lógicos o los lenguajes de programación.

humano sin que éste se dé cuenta que su interlocutor es una máquina, a ésta se le tiene que considerar inteligente. El nombre "Juego de Imitación" alude a que la computadora tiene que imitar o pretender que es un ser humano para ganar. La propuesta es que si se habla se piensa y si se piensa se es inteligente. Esto no excluye que haya formas alternativas de pensar que no involucren al lenguaje, ni que hayas formas de inteligencia que no involucren al pensamiento, pero la prueba sí enfatiza el carácter simbólico y lingüístico del pensamiento y la inteligencia.[10]

Turing presenta un escenario preliminar con tres participantes como se ilustra en la Figura 1.2. El entrevistador E, en el cuarto a la izquierda, es un ser humano –hombre o mujer– cuyo objetivo es identificar el sexo de los participantes en los cuartos del lado derecho. Para este efecto puede hacer preguntas sin ninguna limitación; puede expresar interjecciones aisladas, expresiones no gramaticales e incluso usar palabras sin sentido. Los sujetos del lado derecho son un hombre H y una mujer M. El objetivo de M es ayudar a E y convencerlo de que ella es efectivamente la mujer y su mejor estrategia es posiblemente decir siempre la verdad. El objetivo de H es, por el contrario, engañar a E y convencerlo de que él es la mujer, y para este efecto tiene que mentir. H y M no tienen tampoco ninguna restricción de carácter lingüístico. Incluso pueden simplemente callar.

La pregunta que hace Turing es si dadas estas condiciones el entrevistador E puede descubrir el género de sus interlocutores. E debe dar su respuesta en un tiempo dado: si confunde al hombre con la mujer, H gana y, consecuentemente, E y M pierden. Si acierta, por el contrario, E y M ganan, y H pierde. El escenario está diseñado para que no haya pistas visuales, auditivas o de cualquier otro tipo que permitan al entrevistador ganar por medios no lingüísticos. La prueba se debe repetir un número de veces para evitar resultados contingentes y el resultado es el promedio de veces que H gana. Turing sostiene que ninguno de los participantes tiene o puede idear una estrategia segura para ganar. Y, efectivamente, dadas las

[10] Sin embargo, dichas formas de pensamiento e inteligencia se tendrían que concebir como procesos no computacionales –es decir que no involucran procesos de información, lo cual es inconscebible– o como procesos computacionales que rebasan al modelo de Turing –ya que de otra forma serían a final de cuentas procesos simbólicos computables por la máquina de Turing– pero esta última proposición entra en conflicto directo con la Tesis de Church. Esta paradoja lleva directamente a los límites de la concepción de la inteligencia como un fenómeno computacional y será motivo de reflexión a lo largo de este texto.

Figura 1.2: Escenario de presentación de la Prueba de Turing

condiciones del juego, no la hay. El lector puede jugar este juego un número de veces con sus mejores amigos y sacar sus propias conclusiones. Si E y M ganan significativamente H no sabe mentir, y si el resultado es aleatorio o H gana significativamente, E pierde, M tiene que esforzarse más y a H se le tiene que considerar del género femenino.

Una vez establecidas las presuposiciones y consecuencias del juego, Turing presenta a la prueba propiamente. Ésta consiste simplemente en sustituir al hombre H por una computadora digital y a la mujer por un ser humano de cualquier género, como se ilustra en la Figura 1.3. Asimismo el objetivo del entrevistador es ahora distinguir a la máquina del ser humano. En esta situación si el entrevistador y el humano ganan significativamente la computadora es muy limitada y pierde el juego, pero si el resultado es aleatorio o la computadora gana significativamente, se tiene que conceder que la máquina tiene las facultades mentales de los seres humanos, incluyendo el pensamiento, la inteligencia, los sentimientos y la consciencia. La cuestión es si es posible construir y programar una computadora capaz de ganar el juego.

Independientemente de su implementación práctica, la prueba pasó desde muy temprano al imaginario colectivo en la literatura y el cine de ciencia ficción. La prueba se presentó de manera explícita en la película *2001: Odisea en el Espacio* en la que el astronauta conversa y juega ajedrez con la computadora de la nave espacial HAL-9000. En una escena el astronauta es entrevistado desde la tierra y se le pregunta si percibe a HAL-9000 como un colega, y éste responde que sí, que platica con ella como lo haría con un amigo y no tiene razones

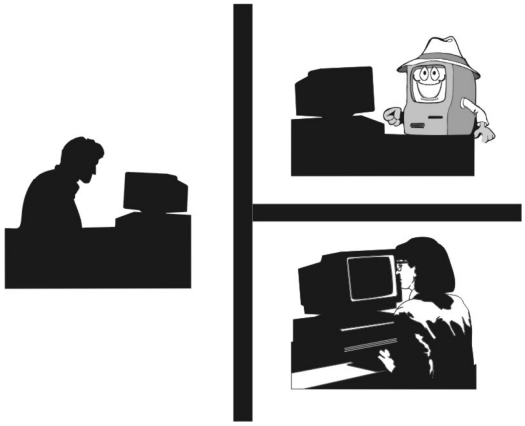

Figura 1.3: Prueba de Turing

para dudar que HAL-9000 piense, sienta e incluso sea consciente, como los seres humanos. En este escenario HAL-9000 pasa la Prueba de Turing.

El escenario se ha elaborado hasta el extremo en que la computadora tiene ya un cuerpo físico que es indistinguible de los seres humanos, como en las películas *Terminator*, *Yo Robot* o *Inteligencia Artificial*, en las que el robot deambula por las calles sin que los seres humanos se percaten de su identidad maquinal. Las preguntas detrás de estos cuentos son cómo sabemos si nuestros interlocutores humanos piensan, sienten y tienen consciencia, cuáles son las consecuencias de las respuestas para nuestra propia concepción de la naturaleza humana y en qué nos distinguimos de las máquinas. Un análisis muy inspirado de los giros que puede tomar esta línea narrativa se presenta en la novela *¿Sueñan los Androides con Ovejas Eléctricas?* de Philip K. Dick, en la que se basa la película *Blade Runner*.

Una tradición mucho más antigua es la historia cabalística del Golem, una criatura con forma humana hecha de barro, pero a diferencia del Adán bíblico, creado por Dios, el Golem es una creación humana. Debe obediencia al rabino que le dio vida, pero en un momento de lucidez se da cuenta que es una marioneta y se obsesiona por manipular sus propias cuerdas. El Golem es el cuerpo, el autómata, la máquina, que se quiere apropiar su alma. Nunca lo hace plenamente pero en la medida en que lo logra se convierte en ser humano. Es la historia de la toma de consciencia. Un cuento más popular es *Pinocho* quien, como el robot de la Prueba

de Turing, miente para que lo crean humano. Su propósito es ser aceptado y posiblemente amado. Pinocho es un juguete con el que nos identificamos porque su predicamento es también el nuestro. Los robots, como los juguetes, son un espejo en el que nos reflejamos. Si son conscientes son humanos. La toma de consciencia es la vida. La humanidad no nos es dada. La humanidad se gana. La Prueba tiene una gran riqueza simbólica, con connotaciones que aluden preguntas muy profundas, y la fuerza de la metáfora es indudable.

1.4. El Problema de la Consciencia

La propuesta del Juego de Imitación entra en conflicto con corrientes de opinión muy fuertes que podrían rechazar su premisa principal. Turing se adelantó a esta eventualidad y él mismo propuso e intentó refutar nueve objeciones, las cuales se analizan a continuación. Se dividen aquí en tres grupos de acuerdo a su relevancia y la fortaleza de los argumentos. En el primero están las objeciones que Turing refuta contundentemente, aunque son las que menos impactan a la propuesta científica; en el segundo aquellas que se requiere refutar para llevar a cabo el programa de investigación, aunque ya son problemáticas; y en el tercero, las relativas al problema de la consciencia, que son centrales al programa filosófico, pero en las que Turing es menos convincente.

En el primer grupo están: 1) la objeción teológica, que sostiene que la consciencia y la inteligencia son facultades que sólo pueden tener los seres creados por Dios; 2) la que sostiene que los seres humanos somos seres superiores y que no puede existir un ente creado por nosotros que nos supere en aspectos esenciales a nuestra naturaleza; y 3) la que admite la posibilidad de que el entrevistador pueda distinguir al ser humano del robot por fenómenos como la telepatía o la telequinesis. Turing descarta estas objeciones porque mientras la máquina y el programa de cómputo están enraizados en el materialismo científico y la tecnología, las objeciones en este grupo no lo están. Sin embargo, Turing refuta a la primera en sus propios términos y arguye que si Dios quisiera dotar de alma a las máquinas, como se la otorga a los seres humanos, sería una limitación que no lo hiciera, pero esto no es posible porque Dios es ilimitado. Los robots serían el vehículo que los seres humanos construiríamos

para acoger las almas que Dios les otorgara. La segunda, a la cual Turing llama *heads in the sand* –que podríamos traducir como "la política del avestruz"– consiste en evitar o posponer los asuntos importantes, o simplemente hacerse de la vista gorda; ésta objeción tiene dos aspectos: la indolencia de no querer prevenir el futuro y la arrogancia de sentirnos superiores. Turing considera que ésta no es realmente una objeción y no requiere respuesta; y la tercera se descarta porque aceptarla sería renunciar al pensamiento científico.

El segundo grupo incluye las objeciones: 1) que la conducta informal no sigue reglas, por lo que no se pude programar; 2) que las computadoras sólo hacen aquellas cosas para las que están programadas y consecuentemente no pueden aprender y ser creativas, que Turing atribuye a Lady Lovelace, Ada; y 3) que el cerebro y el sistema nervioso son continuos y no se pueden simular con una computadora digital. Turing cuestiona en relación a la primera objeción que si efectivamente la conducta informal no se guía por reglas, cómo es entonces que se lleva a cabo. Pone como ejemplo cómo hay que actuar frente a un semáforo, tomando en cuenta las circunstancias excepcionales, como si cuando está en amarillo se atraviesa una pelota o en verde un niño, que pueden extenderse sin límite. El ejercicio es inducir las reglas que causan las conductas observables, que a pesar de ser difícil debe ser posible. Turing distingue aquí las reglas de conducta de las reglas de comportamiento; estás últimas determinan cómo se responde a los fenómenos físicos, como cuando uno se quema con la sartén, que son las leyes de la naturaleza a las que nuestro cuerpo no puede ser ajeno. Estas reglas son seguramente más accesibles. Refutar esta objeción es esencial ya que la conducta informal es la que llevamos a cabo los seres humanos de manera cotidiana.

Respecto a la objeción de Ada, Turing señala que una cosa es definir un conjunto de reglas y otra saber sus consecuencias, y si las computadoras nos sorprenden, las podemos considerar creativas. Por ejemplo, saber las reglas del ajedrez no equivale a ser un buen jugador y conocer los axiomas de un sistema lógico o matemático no equivale a conocer los teoremas en dicho sistema. Por lo mismo si la computadora gana una partida contra un jugador competente o si descubre un teorema se puede considerar creativa.

Se puede agregar que la interacción del agente computacional con otros agentes y el mundo puede dar lugar a conocimiento realmente original. En este sentido, Turing predijo que

sería posible construir programas capaces de modificarse a sí mismos, anticipando el éxito del aprendizaje de máquina de nuestros días. Sin embargo, esta postura se contradice en parte con su propia afirmación de que los estados y salidas de las procesos computacionales están completamente predeterminados –para el demonio de Laplace nunca hay nada nuevo– por lo que las computadoras no pueden aprender ni ser creativas. Por otro lado y de manera más profunda, el aprendizaje y la creatividad requieren de la comprensión pero las computadoras no entienden nada[11] y bajo esta consideración la objeción de Ada se mantiene firme.

En relación a la tercera objeción de este grupo, Turing sostiene que se podría concebir que el robot fuera una computadora analógica; éste no daría respuestas exactas sino aproximadas, pero que esto no afectaría al formato general del juego. Aquí se puede agregar que todo fenómeno continuo se puede aproximar para todo efecto práctico con modelos discretos; sin embargo, no es claro que toda máquina analógica, en particular el cerebro, sea una Máquina de Turing, por lo que esta objeción es también problemática.

En el tercer grupo están las objeciones relacionadas con la experiencia y la consciencia: 1) la matemática que sostiene que hay preguntas que los seres humanos podemos responder pero las máquinas no; 2) las diversas limitaciones de las máquinas con respecto a los seres humanos, como reír, estar triste, amar, jugar, cometer errores, etc.; y 3) la objeción que las máquinas no pueden ser conscientes. La objeción matemática involucra un resultado que Turing mostró (aunque no originalmente) en el artículo de 1936: que hay preguntas que la máquina no puede decidir. Esto se deriva de la imposibilidad de construir la máquina de paro; no es posible, por ejemplo, crear un programa que analice a otro y decida si éste último termina para cierto argumento. Pero un ser humano puede escribir un programa que simplemente ejecute un *loop* infinito –un procedimiento circular para siempre– y sabe que la máquina nunca se va a detener. Turing arguye que esto se puede hacer para casos particulares, pero en última instancia no hay un ser humano que pueda responder a todas las preguntas que una u otra máquina no pueda responder. Se puede agregar en apoyo a Turing, y con base en la tesis de Church, que las funciones computables por la máquina de Turing son las que pueden computar los seres humanos de manera efectiva, por lo que las limitaciones de la máquina

[11] Ver Sección 1.2.3.

son las de los humanos. Sin embargo, la refutación de Turing no toma en cuenta que los seres humanos tenemos a nuestra disposición no sólo la vía sintáctica, que consiste en manipular símbolos, sino también la semántica, que nos permite comprender dichas manipulaciones, como si cierto método de prueba es válido y completo. Los seres humanos tenemos a nuestro alcance la vía semántica justamente porque entendemos, en oposición a las máquinas que carecen de esta facultad. Comprender que algo no se puede decidir, por ejemplo, va mas allá de la manipulación simbólica o del pensamiento mecánico.

Ante la segunda objeción en este grupo, Turing arguye que toda actividad requiere ser programada, y que debe existir un programa para cada limitación. El problema es simplemente de espacio en la memoria de la máquina. Sin embargo, muchas de las limitaciones incluidas en la lista involucran que la máquina debe *sentir* la emoción y no sólo simularla, es decir, expresarla sin sentirla. Reír es la expresión de una emoción y estar triste es una emoción propiamente, que se experimenta al tenerla aunque no se exprese, y aunque se puede programar a un robot para que ría, y lo haga, la causa no es la alegría, la risa es hueca, y el robot no siente nada. Además de contar con las reglas que causan la conducta, los seres humanos, como los animales, sentimos y "tenemos experiencias", en oposición a las máquina que no las tienen. En particular, la Máquina de Turing no tiene ningún componente o estructura que le permita experimentar al mundo o a sí misma, como se puede verificar revisando nuevamente sus elementos constitutivos en la Sección 1.2.

La tercera objeción en este último grupo es justamente que las computadoras digitales no pueden ser conscientes. Ante ésta, Turing sostiene que si la máquina se comporta como si fuera consciente es porque realmente lo es. Aduce que estar en contra de esta proposición es caer en el llamado "solipsismo psicológico" o el problema de las otras mentes: dado que nuestra experiencia y consciencia sólo son accesibles a nosotros mismos, por qué creemos que el resto de los seres humanos –y posiblemente los animales con un sistema nervioso suficientemente desarrollado– tienen sentimientos, experiencias y consciencia. Negarlo es sostener que sólo cada uno de nosotros tiene estos atributos y que estamos solos en el mundo.

Turing enfatiza, sin embargo, que para ganar el juego de imitación se tiene que entender y ser consciente, y no solamente responder de manera mecánica, como lo han hecho desde Eliza

[6] hasta los llamados "chatbots" de hoy en día, aunque tengan una cabeza, un rostro muy expresivo y un cuerpo robótico, como Sofia.[12] Para ilustrar esta distinción Turing apela a lo que él llama juego o prueba de *viva voce* que tiene por objetivo interrogar a alguien para saber si realmente entiende o sólo repite como perico[13] lo aprendido de memoria, como se exige en modelos educativos tradicionales –si se repite sin entender realmente se es en ese momento más como una máquina que como un ser humano. Turing ilustra este sentido de "entender" con una conversación acerca de un verso (en inglés) en la que el sujeto interrogado entiende en un sentido profundo. Pongamos por ejemplo una conversación imaginaria en español entre el interrogador *E* y la computadora *C* –intentando seguir el espíritu del ejemplo de Turing– acerca del soneto de Sor Juana "Que contiene una fantasía contenta con amor decente":[14]

```
E:  El segundo cuarteto empieza:
    ‘‘Si al imán de tus gracias, atractivo,
      sirve mi pecho de obediente acero’’
C: No era el sujeto repulsivo;
   ¿Podrías cambiar ‘‘atractivo’’ por ‘‘adhesivo’’?
E: Sólo que se tropezara y no se pudieran separar;
C: Él sería un meloso
E: Pero eso nunca va a pasar
C: No, primero lo mata
```

Turing diría que si *C* responde de esta forma realmente entiende y por ello tiene que ser consciente. Y, efectivamente, es posible que éste sea el caso. Sin embargo, haría falta programar una computadora para que llevara a cabo conversaciones de este tipo, pero este objetivo no se ve en el horizonte y posiblemente no se pueda lograr del todo.

La posición de Turing respecto al problema de la consciencia es un tanto paradójica ya que al haber propuesto el modelo original y haber participado en el diseño y construcción

[12] https://www.hansonrobotics.com/

[13] Usando la expresión popular y con todo respeto para los pericos.

[14] Sor Juana Inés de la Cruz, Obras Completas, Quinta Edición, Editorial Porrúa, Ciudad de México, 1981, pp. 143.

de varias máquinas prácticas, que tienen un lugar muy destacado en la historia de la computación, sabía que éstas no tienen los componentes estructurales y/o funcionales para ser conscientes. Sin embargo y a pesar de que su propuesta contribuyó de manera muy significativa a cambiar la forma de pensar acerca del pensamiento y la consciencia, él mismo no era ajeno a las corrientes intelectuales de la época y, en especial, al conductismo psicológico, que sostenía que sólo las conductas observables eran objetos genuinos de investigación científica, y que las generalizaciones psicológicas sólo se podían basar en los estímulos y las respuestas. Los organismos eran cajas negras y su interior no se podía examinar por ser inaccesible. La Prueba de Turing refleja claramente esta perspectiva y Turing acepta las consecuencias.

La posición de Turing se aceptó de forma implícita y en ocaciones de manera explícita durante los cincuenta, sesenta e incluso setenta del siglo pasado, y hubo un sector de investigación "dura" que sostuvo que las computadoras eran o podían ser conscientes. Sin embargo, en 1980 John Searle, un filosofo del lenguaje de la Universidad de Berkeley, publicó un artículo en el que presenta un experimento mental, llamado "El Cuarto Chino" (*The Chinese Room*), donde refuta de manera muy sólida que las computadoras puedan entender y ser conscientes en un sentido humano [7].

El experimento consiste en imaginar que hay un cuarto con dos ventanillas –una "de entrada" y otra "de salida"– en el que está una persona que no entiende chino. El sujeto tiene a su disposición un conjunto de instrucciones que tiene que seguir cuando se le da un texto en chino –se asume que las instrucciones codifican un conocimiento perfecto del chino– así como un conjunto de tarjetas en blanco. El sujeto recibe tarjetas en chino por la ventanilla de entrada, las revisa siguiendo las instrucciones, escribe en las tarjetas en blanco y las pasa por la ventanilla de salida. Las tarjetas que recibe se pueden interpretar como preguntas y las que entrega como sus respuestas; alternativamente, se pueden interpretar como una conversación entre el sujeto que está en el cuarto y quien le da y recibe las tarjetas, que es un hablante nativo del chino, para quien la conversación se lleva a cabo de manera natural en chino. Bajo este supuesto, Searle mantiene que el sujeto dentro del cuarto lleva a cabo un proceso simbólico sin entender y por lo mismo sin ser consciente, pero para el observador externo se comporta como si realmente tuviera estos atributos. Por ejemplo, si la conversación fuera como la

mostrada en el juego de *viva voce* y quienes estuvieran afuera y adentro del cuarto fueran *E* y *C* respectivamente, *E* pensaría que *C* entiende y es consciente, pero por los supuestos del escenario éste no sería el caso.

Una vez que se presenta el argumento Searl, al igual que Turing, anticipa un conjunto de posibles objeciones y procede a refutarlas, y aunque el argumento no es conclusivo para mucha gente, ha dado lugar a un debate que por su fuerza y vigencia se mantiene abierto a la fecha [8]. Una consideración fundamental a favor de Searle, que se hizo explícita mucho después de la presentación original de la Máquina de Turing, especialmente con las contribuciones de Kleene y Davis, y cuya importancia se enfatiza por Boolos y Jeffreys, es que las estructuras simbólicas en la cinta y en la tabla de estados se interpretan convencionalmente en relación a una configuración estándar. Sin estas convenciones e interpretaciones las cadenas de símbolos no significan nada. Y la interpretación es un acto de la consciencia. Por lo mismo, para que haya computación tiene que haber interpretación, y si hay interpretación tiene que haber consciencia. Consecuentemente, si hay computación hay consciencia. Es decir, la computación es un acto de la consciencia apoyado en un dispositivo material con la capacidad de manipular símbolos de manera mecánica.

Una consecuencia de esta distinción, que era ajena en aquella época y que aún en día no se reconoce plenamente, es que la computación no es una propiedad objetiva de los mecanismos. Tampoco lo es de los fenómenos naturales, ya sean estos físicos, químicos o biológicos. Estos se pueden aprovechar para realizar el proceso mecánico, ya sea la manipulación de símbolos o un proceso continuo, pero en todo caso, la computación requiere el acto de consciencia que le da significado a las entradas y salidas, y al proceso mismo [9].

Como corolario se puede distinguir a la *computación artificial*, que es la que realizan la Máquina de Turing y en general las computadoras digitales, de la *computación natural*, que realizan los cerebros biológicos. La primera requiere a dos entes diferentes: la máquina, que hace la manipulación simbólica, y el ser humano, que hace la interpretación; mientras que en la segunda quien manipula los símbolos y quien hace la interpretación es el mismo ente [9].

Esta observación permite también distinguir a las computadoras de las máquinas convencionales: las entradas y salidas de las computadoras se interpretan como representaciones, mientras que las de las máquinas convencionales no representan nada –y si representaran y se interpretaran pasarían por este simple hecho a ser computadoras.

Estas consideraciones ponen en perspectiva al programa de investigación tal como lo propuso Turing. La interpretación directa da lugar a la llamada "IA dura", que asume que la máquina entiende, tiene experiencias y es consciente. Por su parte, la interpretación que se sigue del presente argumento es la llamada "IA débil"que pretende hacer máquinas capaces de percibir, hablar, razonar, aprender y actuar. etc., pero no que sientan o sean conscientes. Esta oposición da lugar a dos tipos de problemas de la mente: los informacionales, que se refiere al proceso de información requerido para llevar a cabo la función, y los problemas relacionados con la experiencia y la consciencia. Las computadoras y los robots llevan a cabo sólo los primeros, que son la causa de sus conductas observables, mientras que los seres humanos y los animales llevamos a cabo ambos. Esta oposición se retoma en filosofía de la ciencia que distingue los problemas "débiles" de la consciencia, que son los informacionales, del problema "fuerte" o "duro", que se refiera a tener experiencias y consciencia [10].[15]

Por estas razones el programa de la IA dura no es viable y se tiene que adoptar la IA débil. Esto no ha representado un problema en la práctica ya que desde un principio todos los programas de cómputo desarrollados en la disciplina se han concebido y evaluado en relación a su función informacional, independientemente de las propiedades semánticas que los propios investigadores o el público en general les atribuyan. Esto podría decepcionar a algunos, ya que no se arroja luz sobre los sentimientos y la consciencia, que son esenciales a la naturaleza humana, y seguramente a la naturaleza de diversas especies animales con cierto nivel de desarrollo cerebral. Sin embargo, modelar computacionalmente las facultades de la mente vale la pena por varias razones; por ejemplo:

1. Aporta conocimiento causal de dichas facultades en oposición a disciplinas no computacionales que ofrecen perspectivas descriptivas. Por ejemplo, la lingüística y la psico-

[15] Esta distinción se presentó de manera muy amena en México en el artículo "El Misterio de la Consciencia" por John Searle en la revista *Vuelta* No. 232, marzo de 1996.

lingüística ofrecen conocimiento de diversos aspectos del lenguaje, pero son muy limitados para explicar la maquinaría mental que nos permite hablar.

2. Nos permite construir máquinas virtuales y robots físicos capaces de asistirnos en diversas tareas de la vida cotidiana.

3. Nos permite llevar hasta el límite la metáfora de la computación como modelo de la mente: Si el cerebro no es una computadora y la mente no es un programa de cómputo ¿en qué sentido el cerebro procesa información? Las respuestas que se den a esta pregunta podrían arrojar luz al problema de la experiencia y la consciencia.

1.5. El Programa de Investigación

Una vez concluida la discusión de las objeciones, Turing esboza el programa de investigación de la Inteligencia Artificial que en un sentido muy literal se ha seguido a la fecha. Éste es por supuesto construir programas de cómputo que modelen las facultades de la mente. Turing tenía gran confianza en que los avances de la tecnología serían de gran magnitud durante la segunda parte del siglo XX y que sería posible construir computadoras capaces de correr dichos programas. Bajo este supuesto Turing estipuló muy claramente que los investigadores en Inteligencia Artificial son analistas y programadores de computadoras. Los objetos de análisis son las facultades de la mente: la percepción, el lenguaje, el pensamiento, el aprendizaje y la acción motora.

Modelar computacionalmente dichas funciones presupone también dotar a las máquina de conocimiento. Éste puede ser de carácter conceptual o de habilidades; ejemplos del primero son jugar ajedrez y probar un teorema, y del segundo andar en bicicleta, pilotear un avión o realizar una cirugía en el cerebro humano. La frontera entre estos dos tipos de conocimiento es difusa, pero intuitivamente se puede decir que el primero se adquiere a través del lenguaje y se consolida mediante la reflexión mientras que el segundo se aprende con el ejercicio cotidiano de la habilidad.

Es necesario considerar también que el objeto fundamental de la computación es la función matemática, por lo que tanto el conocimiento conceptual como el de habilidades se

representan con funciones matemáticas. Esta relación se hace explícita cuando se utiliza el *cálculo-λ* o su implementación en el lenguaje de programación Lisp, ya que las representaciones se expresan directamente como funciones. Incluso los teoremas lógicos y matemáticos, como el teorema de Pitágoras, se representan por funciones matemáticas (ver, por ejemplo, [11]). El objeto en la mente es la función representada y el propio proceso mental es su evaluación en relación a sus argumentos, que provienen de la percepción o de otros objetos representados en la mente. Ésta es una premisa esencial que hay que aceptar en la Inteligencia Artificial: todo objeto de conocimiento se representa, literalmente, por una función matemática. Más aún, todo programa de cómputo, expresado en cualquier computadora, representa a una función matemática.

Esta visión tiene raíces muy profundas en el pensamiento humano y frecuentemente se asocia al realismo platónico, como se describe en el Mito de la Caverna. En éste Platón imagina unos hombres encadenados en una cueva iluminada por una fogata. Detrás de ellos y delante del fuego hay otros hombres que cargan objetos diversos cuyas sombras se proyectan en la pared del fondo. Los hombres encadenados, como espectadores en el cine, sólo pueden ver las sombras, su única vía para conocer a los objetos reales, a los que Platón llama "las ideas". En esta analogía los objetos matemáticos, abstractos e inmutables, son las ideas, a las que podemos llegar a través de sus imágenes, impuras y mutables. Representar el conocimiento es traducir las imágenes que se nos presentan a través de las diversas modalidades de la percepción a funciones matemáticas, que a su vez representan a los objetos que están en el mundo.

Turing describió dos vías parea proveer conocimiento a las computadoras: 1) dárselos directamente; y 2) construir máquinas de aprendizaje que generaran por sí mismas el conocimiento. Turing ejemplificó el primer tipo con el conocimiento lógico. Para que una computadora pueda probar teoremas requiere que se le den los axiomas y las reglas de inferencia del sistema. Adelantó que las proposiciones se podrían etiquetar de acuerdo a su tipo: declarativas, imperativas, emotivas, etc., y qué podrían incluirse como argumentos de predicados de creencias, obligaciones y deseos; asimismo, que las proposiciones intencionales deducidas podrían dar lugar a la conducta intencional de la máquina.

Otro ejemplo, que Turing desarrolló por mucho tiempo, fue el ajedrez.[16] Para jugar es necesario dotar a las máquinas de las reglas del juego y de las estrategias de un jugador competente. Adelantó que las consecuencias de aplicar dichos sistemas da lugar a un espacio del problema, normalmente de grandes dimensiones, y que la inteligencia del sistema dependería en buena medida de las heurísticas con las que se explore dicho espacio para encontrar las soluciones.

Turing especula acerca de la capacidad de la memoria que se requería para programar el juego de imitación así como el esfuerzo humano requerido para hacer dicho programa, y concluye que para programar la máquina se requiere también emplear la vía del aprendizaje. Para este efecto presenta varias metáforas que ilustran cómo puede aprender una máquina. Una de éstas es la fisión nuclear. En ésta, las ideas comunes en la memoria son como masas atómicas sub-críticas y una idea que se le presente a la mente es como un neutrón que "se inyecta" a la masa causando una inestabilidad ligera para volver rápidamente al equilibrio, como cuando un martillo impacta las cuerdas de un piano. Sin embargo, un neutrón que se inyecte a una masa super-crítica causará una reacción en cadena. Este caso corresponde al aprendizaje y la creatividad –donde una idea genera una cadena de ideas, una teoría, que enriquece el estado de conocimiento.

El estado de la mente adulta es el resultado de un proceso que se inicia con el estado al nacer e involucra la educación que el sujeto recibe y otros tipos de experiencia que no se puedan considerar como educación. Para construir la máquina de aprendizaje propiamente Turing propone construir "La Máquina Niño" (*The Child Machine*) con 1) un estado inicial que corresponde al estado de la mente al nacer, que es como un cuaderno en blanco; 2) un conjunto de mecanismos de aprendizaje simples pero de carácter muy general; 3) mucha educación, tanto como se les pueda dar a los niños; y 4) mucha experimentación. Turing hace aquí una analogía con la evolución y propone que la estructura de la máquina niño corresponde al material hereditario; los cambios hechos a la máquina a las mutaciones; y los juicios del expe-

[16] Turing desarrolló el primer programa capaz de jugar ajedrez, al que llamó *Turochamp*, aunque en su tiempo no había computadoras con la velocidad de proceso y capacidad de memoria para correrlo, por lo que él mismo jugaba el papel de la máquina, como un computador humano, y hacia cálculos durante media hora para hacer cada movida.

rimentador a la selección natural. Sin embargo, las mutaciones las dirige el investigador por lo que el proceso puede ser mucho más rápido que la evolución natural. Turing explora también diferentes formas de aprendizaje que incluyen el condicionamiento operante, basado en recompensas y penalizaciones, pero también es explícito en que para aprender se requiere el uso del lenguaje.

La propuesta incluye incorporar un elemento aleatorio a los programas de aprendizaje, análogo a los procesos evolutivos. Esto puede resultar paradójico a la luz del carácter determinístico de las computadoras digitales, como se discute al inicio de la Sección 1.2.1. Sin embargo, Turing propuso métodos para simular la generación de números aleatorios por medio de procesos determinísticos, con lo que se originó la computación estocástica. Por ejemplo, propuso utilizar como números aleatorios los dígitos sucesivos de números irracionales como π. Posteriormente se han diseñado funciones más eficientes basadas en la misma idea, además de que se han diseñado generadores de hardware que escogen un número a partir de un fenómeno físico medido dinámicamente en el medio exterior; o la hora, minutos y segundos al momento de generar el número. Desde el punto de vista de la máquina, este tipo de funciones tienen un argumento "oculto" que se aporta por el generador de números aleatorios, por lo que el valor de la función no se puede predecir, pero el proceso sigue siendo determinístico en última instancia.

Los elementos aleatorios son muy importantes cuando el espacio del problema es muy grande, ya que puede haber regiones muy significativas del espacio en las que no haya ninguna solución, pero como esto no se puede saber de antemano, un programa de búsqueda determinístico realizaría un esfuerzo muy grande en tierra estéril. Por otra parte, utilizar números aleatorios permite saltar a diferentes regiones del espacio y buscar en regiones locales por un tiempo hasta encontrar la solución o saltar nuevamente. El ejemplo que da Turing es buscar un número entre 50 y 200 que sea igual a la suma del cuadrado de sus dígitos. El lector puede ensallar la estrategia determinística, probando del 50 al 200 de uno en uno, o la estocástica, escogiendo un número aleatorio, buscar en su vecindad, y si no se encuentra la solución, saltar aleatoriamente a otra zona. Ambos métodos tienen sus ventajas y limitaciones. El determinístico encuentra la solución si ésta existe, pero puede tardar mucho y recorrer

todo el espacio en vano, mientras que el aleatorio podría encontrar la solución rápidamente, pero puede ciclar y no hay garantía que termine. Otra estrategia es usar el método estocástico y guardar los valores ya tratados, pero hay que considerar el costo de memoria. Lo interesante es que el método aleatorio se parece más a la evolución ya que ensaya un valor y salta aleatoriamente sin tener memoria.

Turing concluye la exposición del programa sugiriendo dos tareas: la primera es construir máquinas capaces de jugar ajedrez, posiblemente como una actividad paradigmática de "pensar" y la segunda construir máquinas capaces de entender el lenguaje natural, como el inglés o el español, que corresponde directamente a la construcción del programa capaz de ganar el juego de imitación. Esta tarea requiere modelar la estructura del lenguaje y la conversación, el razonamiento conceptual y deliberativo que se requiere para hablar, y la memoria para almacenar el conocimiento. Turing especuló que para finales del siglo XX habría máquinas capaces de jugar el juego de imitación por cinco minutos con un 70 % de probabilidades de que el experto no pudiera distinguir a la máquina del ser humano. Si se toma en cuenta el tipo de conversación a la que se refiere Turing, como el diálogo de *viva voce* en la Sección 1.4, esta predicción no se cumplió y no se ve que se vaya a cumplir en el corto o mediano plazo. Sin embargo, si Turing hubiera hecho una predicción similar para el ajedrez habría acertado: desde 1996, cuando la computadora Deep Blue de la IBM le ganó a Kasparov, no es ya posible para los seres humano vencer a las máquinas. Las razones por las cuales se tuvo éxito en el ajedrez pero no en la máquina del lenguaje se abordarán a lo largo de este texto. De forma más general Turing esperaba que las máquinas pudieran competir con los seres humanos en todos los ámbitos meramente intelectuales –o ser competentes en todas las habilidades mentales.

Finalmente, en el contexto más amplio, Turing estipuló que "lo mejor sería proveer a la máquina con los mejores órganos sensoriales que el dinero pueda comprar" (Turing, 1950, S. 7, pp. 36), hacerla competente en el lenguaje –enseñarla a entender y hablar– y habilitar el proceso de enseñanza al que se someten los niños. Es decir, construir una máquina que aprenda el lenguaje, para que pueda aprender a través del lenguaje.

La incorporación de dispositivos sensoriales anticipa implícitamente la creación de robots con capacidades de percepción y motoras. Para este efecto se requiere modelar adicionalmente el reconocimiento visual y del resto de las modalidades de la percepción, así como el conocimiento de las habilidades motoras. La meta de esta extrapolación es contar con máquinas competentes en todas las facultades intelectuales y corporales de los seres humanos, y en general de los animales. El programa de la Inteligencia Artificial consiste en explorar la metáfora computacional hasta sus últimas consecuencias y hacer explícitas tanto sus posibilidades como sus limitaciones.

Bibliografía

[1] A. M. Turing, Computing machinery and intelligence, Mind 59 (1950) 433—460.

[2] A. M. Turing, On computable numbers, with an application to the entscheidungs problem, Proceedings of the London Mathematical Society 42 (1936) 230–265.

[3] C. Petzold, The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine, Wiley Publishing, 2008.

[4] G. S. Boolos, R. C. Jeffrey, Computability and Logic (Third Edition), Cambridge University Press, 1989.

[5] M. D. Davis, Computability and Unsolvability, McGraw-Hill Series in Information Processing and Computers, McGraw-Hill, 1958.

[6] J. Weizenbaum, Eliza—a computer program for the study of natural language communication between man and machine, Commun. ACM 9 (1) (1966) 36–45. doi:10.1145/365153.365168.
URL http://doi.acm.org/10.1145/365153.365168

[7] J. R. Searle, Minds, brains and programs, Behavioral and Brain Sciences 3 (3) (1980) 417–57.

[8] D. Cole, The chinese room argument, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, spring 2019 Edition, Metaphysics Research Lab, Stanford University, 2019.

[9] L. A. Pineda, The mode of computing, CoRR abs/1903.10559. arXiv:1903.10559.
URL http://arxiv.org/abs/1903.10559

[10] D. Chalmers, The Conscious Mind: In Search of a Fundamental Theory, Oxford University Press, Inc., New York, NY, USA, 1996.

[11] L. A. Pineda, Conservation principles and action schemes in the synthesis of geometric concepts, Artificial Intelligence 171 (2007) 197–238.

[12] J. Piaget, Seis Estudios de Psicología, Barral Editores, S. A., Barcelona, 1970.

[13] S. Carey, D. Zaitchik, I. Bascandziev, Theories of development: In dialog with jean piaget, Developmental Review 38 (2015) 36–54.
URL https://doi.org/10.1016/j.dr.2015.07.003

[14] A. D. Baddeley, The concept of working memory: A view of its current state and probable future, Cognition 10 (1981) 17–23.

[15] M. Posner, M. K. Rothbart, Research on attention networks as a model for the integration of psychological science, Annu Rev Psychol 58 (2007) 1–23.

[16] M. Posner, M. K. Rothbart, H. Ghassemzadeh, Restoring attention networks, Yale Journal of Biology and Medicine 92 (2019) 139–143.

[17] S. Carey, The Origin of Concepts, Oxford University Press, New York, 2010.

[18] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81—132.

[19] R. J. Brachman, J. G. Schmolze, An overview of the kl-one knowledge representation system, Cognitive Science 9 (2) (1985) 171—216.

[20] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2010.

[21] E. Tulving, Memory systems: episodic and semantic memory, in: E. Tulving, W. Donaldson (Eds.), Organization of Memory, New York: Academic Press, 2013, pp. 381–403.

[22] L. A. Pineda, A. Rodríguez, G. Fuentes, C. Rascón, I. V. Meza, A light non-monotonic knowledge-base for service robots, Intel Serv Robotics 10 (2017) 159–171.

[23] L. A. Pineda, A. Rodríguez, G. Fuentes, N. Hernández, M. Reyes, C. Rascón, R. Cruz, I. Vélez, H. Ortega, Opportunistic inference and emotion in service robots, Journal of Intelligent & Fuzzy Systems, 34 (5) (2018) 3301–3311.

[24] I. Torres, N. Hernández, A. Rodríguez, G. Fuentes, L. A. Pineda, Reasoning with preferences is service robots, Journal of Intelligent & Fuzzy Systems 36 (5) (2019) 5105–5114.

[25] L. A. Pineda, N. Hernández, I. Torres, G. Fuentes, N. P. D. Ávila, Practical non-monotonic knowledge-base system for un-regimented domains: A case-study in digital humanities, Information Processing & Management 57 (3) (2020) 102214.

[26] G. Brewka, T. Eiter, M. Truszczyński, Answer set programming at a glance, Communications of the ACM 54 (12) (2011) 92—103.

[27] H. L. Levesque, R. Brachman, A fundamental tradeoff in knowledge representation and reasoning, in: R. Brachman, H. Levesque (Eds.), Readings in Knowledge Representation, Morgan and Kaufmann, Los Altos, CA, 1985, pp. 41–70.

[28] H. L. Levesque, Logic and the complexity of reasoning, Journal of Philosophical Logic 17 (1988) 355–389.

[29] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (3) (1948) 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x. URL https://ieeexplore.ieee.org/document/6773024/

[30] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation (3rd Edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[31] J. Martin, Introduction to Languages and the Theory of Computation (Third edition), McGraw Hill, 2003.

[32] L. A. Pineda, A distributed extension of the turing machine, CoRR abs/1803.10648. arXiv:1803.10648.
URL http://arxiv.org/abs/1803.10648

[33] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, General reinforcement learning algorithm that masters chess, shogi, and go through self-play, Nature 362 (2018) 1140–1144. doi:org/10.1126/science.aar6404.

[34] H. Simon, Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting., Wiley, New York, 1957.

[35] A. Newell, H. Simon, Computer science as empirical inquiry: Symbols and search, Communications of the *ACM* 19 (3) (1976) 113–126.

[36] H. A. Simon, The Sciences of the Artificial, 3rd Edition, MIT Press, Cambridge, MA, 1996.

[37] A. Newell, The knowledge level, Artificial Intelligence 18 (1982) 87–127.

[38] J. Pearl, Causality, 2nd Edition, Cambridge University Press, Cambridge, UK, 2009. doi:10.1017/CBO9780511803161.

[39] L. E. Sucar, Probabilistic Graphical Models Principles and Applications, 1st Edition, Advances in Computer Vision and Pattern Recognition, Springer London, London, 2015.

[40] J. R. Anderson, G. H. Bower, Human Associative Memory: A Brief Edition, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1980.

[41] Aristoteles, Obras, Aguilar S. A. de ediciones, Madrid, 1964.

[42] C. Shields, Aristotle's psychology, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, winter 2016 Edition, Metaphysics Research Lab, Stanford University, 2016.

[43] N. Chomsky, Syntactic Structures, The Hague/Paris: Moutons, 1957.

[44] N. Chomsky, A review of b. f. skinner's verbal behavior, Language 35 (1) (1959) 26–58.

[45] J. A. Fodor, The Language of Thought, Harvard University Press, 1975.

[46] B. C. Smith, Prologue to "reflection and semantics in a procedural language", in: H. L. R. Brachman (Ed.), Readings in Knowledge Representation, Morgan and Kaufmann, Los Altos, CA, 1985, pp. 31–40.

[47] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, Henry Holt and Co., Inc., New York, NY, USA, 1982.

[48] J. R. Searle, Minds, brains, and programs, Behavioral and Brain Sciences 3 (3) (1980) 417–457.

[49] G. S. Boolos, R. C. Jeffrey, Computability and Logic (Third Edition), Cambridge University Press, Cambridge, 1989.

[50] D. E. Rumelhart, J. L. McClelland, the PDF Research Group, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol.1: Foundations, The MIT Press, Cambridge, Mass., 1986.

[51] G. E. Hinton, J. L. McClelland, D. E. Rumelhart, Distributed representations (chapter 3), in: D. E. Rumelhart, J. L. McClelland (Eds.), Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol.1: Foundations, The MIT Press, Cambridge, Mass., 1986.

[52] J. A. Fodor, Z. W. Pylyshyn, Connectionism and cognitive architecture: A critical analysis, Cognition 28 (1-2) (1988) 3–71.

[53] D. J. Chalmers, Syntactic transformations on distributed representations, Connection Science 2 (1–2) (1990) 53–62. doi:10.1080/09540099008915662.

[54] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[55] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (11) (2015) 436–444. doi:10.1038/nature14539.

[56] J. G. Rueckl, Connectionism and the notion of levels, in: T. Horgan, J. Tienson (Eds.), Connectionism and the Philosophy of Mind. Studies in Cognitive Systems, vol 9, Springer, Dordrecht, 1991.

[57] H. T. Siegelmann, E. D. Sontag, On the computational power of neural nets, Journal of Computer and System Sciences 50 (1) (1995) 132–150. doi:org/10.1006/jcss.1995.1013.

[58] G. Z. Sun, H. H. Chen, Y. C. Lee, C. L. Giles, Turing equivalence of neural networks with second order connection weights, in: Anon (Ed.), Proceedings. IJCNN - International Joint Conference on Neural Networks, Publ by IEEE, 1992, pp. 357–362.

[59] M. L. Anderson, Embodied cognition: A field guide, Artificial Intelligence 149 (2003) 91–130.

[60] T. Froese, T. Ziemke, Enactive artificial intelligence: Investigating the systemic organization of life and mind, Artificial Intelligence 173 (2009) 466–500.

[61] J. A. Fodor, The mind-body problem, Scientific American 244 (1981) 114–123.

[62] S. M. Kosslyn, W. L. Thompson, G. Ganis, The Case for Mental Imagery, Oxford Univ. Press, 2006.

[63] R. Shepard, L. Cooper, Mental images and their transformations, MIT Press, Cambridge, Mass., 1982.

[64] M. Tye, The Imagery Debate, A Bradford Book, The MIT Press, Cambridge, Mass., 1991.

[65] Z. W. Pylyshyn, What the mind's eye tells the mind's brain: A critique of mental imagery, Psychological Bulletin 80 (1973) 1–24.

[66] B. J. Copeland, The church-turing thesis, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, spring 2019 Edition, Metaphysics Research Lab, Stanford University, 2019.

[67] J. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation (3rd Edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[68] D. Chalmers, The Conscious Mind: In Search of a Fundamental Theory, Oxford University Press, Inc., New York, NY, USA, 1996.