Modelos de Computación
Cuestionario 0

Sofía Fernández Moreno
Universidad de Granada
12 de octubre de 2015

Índice

1. El Entscheidungsproblem

El **Entscheidungsproblem** (*"Problema de decisión"*) es un reto propuesto por **David Hilbert** y **Wilhelm Ackermann** en 1928. El Entscheidungsproblem buscaba un algoritmo que tomase una fórmula lógica de primer orden y responda si es universalmente válida. Por el **teorema de complitud de Gödel**, una fórmula lógica es universalmente válida si y sólo si puede ser deducida a partir de los axiomas, por tanto, el **Entscheidungsproblem** puede ser visto como la búsqueda de un algoritmo que decida si una fórmula lógica dada se puede probar de los axiomas usando las reglas de la lógica. En 1936, **Alonzo Church** y **Alan Turing** demostraron, por independiente, que es imposible escribir tal algoritmo.

El origen del **Entscheidungsproblem** se remonta a **Gottfried Leibniz**, quién en el siglo XVII, tras haber construido con éxito una máquina mecánica de cálculo, soñaba con construir una máquina que puediera manipular símbolos para determinar si una frase en matemáticas era un teorema. Se dio cuenta de que lo primero era tener un lenguaje formal y claro, y mucho de su trabajo posterior estuvo orientado a ese objetivo.

2. Church, Turing y el Entscheidungsproblem

Antes de poder responder a la pregunta, se necesitaba definir formalmente la noción de *"algoritmo"*. Esto lo hizo **Alonzo Church** en 1936, con el concepto de *"calculabilidad efectiva"* basado en su *cálculo λ*, y **Alan Turing** en el mismo año, con su concepto de *máquinas de **Turing***. **Turing** inmediatamente reconoció que ambos eran modelos de computación equivalentes.

La respuesta negativa al **Entscheidungsproblem** fue dada por **Alonzo Church** entre 1935 y 1936, y de manera independiente muy poco tiempo después por **Alan Turing** en 1936. **Church** probó que no hay ninguna función computable que decida si dos expresiones de *λ-cálculo* son equivalentes o no, basándose en el trabajo previo de **Stephen Kleene**. **Turing** lo resolvió basándose en el *Problema de la Parada* para *máquinas de **Turing***. El trabajo de ambos autores estuvo fuertemente influenciado por el trabajo previo de **Kurt Gödel** sobre el teorema de incomplitud, especialmente en el método de asignar números(un *numerado de **Gödel***) a fórmulas lógicas para reducir la lógica a la aritmética.

El **Entscheidungsproblem** está relacionado con el décimo problema de **Hilbert**, que busca un algoritmo que decida cuando una ecuación diofántica tiene solución. La no existencia de ningún algoritmo, establecida por **Yuri Matiya Sevich** en 1970, también implica una respuesta negativa al **Entscheidungsproblem**.

Es importante notar que si se restringe el problema a una teoría de primer orden específica con constantes, predicados constantes y axiomas, es posible que exista un algoritmo de decisión para la teoría. Algunos ejemplos de teorías decidibles son: la *aritmética de* ***Presburger*** y los sistemas estáticos de tipos de los lenguajes de programación.

Sin embargo, la teoría general de primer orden para los números naturales conocida como la aritmética de Peano no puede ser decidida con ese tipo de algoritmo.

3. Vida y contribución de Turing a la computabilidad

Alan Mathison Turing, (Paddington, Londres, 23 de junio de 1912 - Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la computación, criptógrafo y filósofo británico.

Es considerado uno de los padres de la ciencia de la computación siendo el precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la *máquina de* ***Turing***. Formuló su propia versión de la **Tesis de Church-Turing**.

3.1. Estudios

Turing dio muestras ya desde una edad muy temprana del ingenio que más tarde mostraría prominentemente. Desde el principio mostró un gran interés por los números y los rompecabezas. Sus padres lo inscribieron en el colegio *St. Michael* cuando tenía seis años. Su profesora se percató enseguida de la genialidad de **Turing**, tal como ocurrió a sus profesores posteriores.

En 1926, ingresó en el internado de *Sherborne* en Dorset. Su primer día de clase coincidió con una huelga general en Inglaterra, pero su determinación por asistir a clase en su primer día era tan grande que recorrió en solitario con su bicicleta las más de 60 millas que separaban Southampton de su escuela, pasando la noche en una posada. Tal hazaña fue recogida en la prensa local.

La inclinación natural de **Turing** hacia las matemáticas y la ciencia no le forjó el respeto de sus profesores de *Sherborne*, cuyo concepto de educación hacía mayor énfasis en los clásicos. Pero a pesar de ello, **Turing** continuó mostrando una singular habilidad para los estudios que realmente le gustaban, llegando a resolver problemas muy avanzados en 1927 sin ni siquiera haber estudiado cálculo elemental.

En 1928, **Turing** descubrió los trabajos de **Albert Einstein** y, no sólo pudo comprenderlos, sino que además infirió las críticas de **Einstein** a las **Leyes de Newton** de la lectura de un texto en el que no estaban explícitas. Durante su edad escolar, **Turing** fue un joven cuyo optimismo y ambiciones se vieron acrecentados debido en gran parte a su intensa unión con su amigo Christopher Morcom, cuya muerte, aún joven, afectaría a Turing profundamente. La fe religiosa de **Turing** se hizo pedazos, volviéndose ateo. Adoptó la convicción de que todos los fenómenos, incluyendo el funcionamiento del cerebro humano, deben ser materialistas. Sin embargo siguió creyendo en la supervivencia del espíritu después de la muerte.

Debido a su falta de voluntad para esforzarse en el estudio de los clásicos, **Turing** suspendió sus exámenes finales varias veces y tuvo que ingresar en la escuela universitaria que eligió en segundo lugar, *King's College*, Universidad de Cambridge, en vez de en la que era su primera elección, *Trinity*. Recibió las enseñanzas de **Godfrey Harold Hardy**, un respetado matemático. En 1935 **Turing** fue nombrado profesor del *King's College*.

3.2. La máquina de Turing

Turing reformuló los resultados obtenidos por **Kurt Gödel** en 1931 sobre los límites de la demostrabilidad y la computación, sustituyendo al lenguaje formal universal descrito por **Gödel** por el concepto de *Máquina Universal (de **Turing**)*, con la tesis de que dicha máquina podría realizar las mismas tareas que cualquier otro tipo de máquina. **Turing** demostró que dicha máquina era capaz de implementar cualquier problema matemático que pudiera representarse mediante un algoritmo. Su estudio también introduce el concepto de números definibles.

La mayor parte de 1937 y 1938 la pasó en la Universidad de Princeton, estudiando bajo la dirección de **Alonzo Church**. En 1938 obtuvo el doctorado en Princeton. En su discurso, introdujo el concepto de hipercomputación, en el que ampliaba las *máquinas de **Turing*** con las llamadas *máquinas con oráculos*, las cuales permitían el estudio de los problemas para los que no existe una solución algorítmica.

3.3. Análisis criptográfico

Durante la **Segunda Guerra Mundial** fue uno de los principales artífices de los trabajos para descifrar los códigos secretos nazis. Sus observaciones matemáticas contribuyeron a romper los códigos de la máquina **Enigma** y de los codificadores de teletipos *FISH*. Sus estudios del sistema *FISH* ayudarían al desarrollo posterior de la primera computadora programable electrónica digital llamada **Colossus**. Dicha computadora se utilizó para descifrar los códigos *FISH* (en concreto las transmisiones de la máquina **Lorenz**).

3.4. Inteligencia artificial y Test de Turing

De 1945 a 1948 **Turing** trabajó en el **NPL**(Laboratorio Nacional de Física) en el diseño del **ACE**(Motor de Computación Automática). En 1946 presentó un estudio que se convertiría en el primer diseño detallado de un computador automático. Aunque el diseñar el **ACE** era factible, el secretismo que reinaba durante la guerra desembocó en retrasos para iniciar el proyecto por lo que **Turing** se sintió desilusionando. En 1947 se tomó un año sabático en Cambrdige, tiempo durante el cual produjo el germen sobre la *Inteligencia Artificial* que no fue publicada en vida. Mientras se encontraba en Cambrdige el piloto del **ACE** estaba siendo construido a pesar de su ausencia. **ACE** ejecutó su primer programa informático en mayo de 1950. Aunque la versión completa de la **ACE** de **Turing** jamás fue construida, un gran número de computadoras alrededor del mundo le debe muchísimo a ésta.

En 1950, **Turing** trató el problema de la *Inteligencia Artificial* y propuso un experimento que hoy se conoce como **Test de Turing**, con la intención de definir una prueba estándar por el que una máquina podría catalogarse como "*sensible*" o "**sintiente**". En el documento, **Turing** sugirió que en lugar de construir un programa para simular la mente adulta, sería mejor producir uno más simple para simular la mente de un niño y luego someterlo a educación.

Entre 1948 y 1950 en conjunto con un antiguo colega, **D.G. Champernowne**, empezó a escribir un programa de ajedrez para un ordenador que aún no existía. En 1952 trató de implementarlo en el **Ferranti Mark 1**, pero a falta de potencia el ordenador no fue capaz de ejecutar el programa. En su lugar, **Turing** jugó una partida en la que él simuló al ordenador, tomando alrededor de hora y media en efectuar un movimiento. Una de las partidas llegó a registrarse. El programa perdió frente a un colega de **Turing**, **Alick Glennie**. Su test fue significativo, provocativo y una gran contribución para empezar el debate alrededor de la *Inteligencia Artificial* que aún hoy continua.

Trabajó junto a **Norbert Wiener** en el desarrollo de la cibernética. Esta rama de estudios se genera a partir de la demanda de sistemas de control que exige el progresivo desarrollo de las técnicas de producción a partir del siglo XX. La cibernética pretende establecer un sistema de comunicación entre el hombre y la máquina como premisa fundamental para administrar los sistemas de control. Sus estudios profundizaron en esta relación estableciendo el concepto de interfaz y cuestionando los límites de simulación del razonamiento humano.

3.5. Biología matemática

Turing trabajó desde 1952 hasta que falleció en 1954 en la biología matemática, concretamente en la *morfogénesis*. Publicó un trabajo sobre esta materia en 1952. Su principal interés era comprender la *filotaxis de **Fibonacci***, es decir, la existencia de los **números de Fibonacci** en las estructuras vegetales. Utilizó ecuaciones de reacción-difusión que actualmente son cruciales en el campo de la formación de patrones.

3.6. Muerte

La carrera profesional de **Turing** se vio truncada cuando lo procesaron por su homosexualidad. En 1952, Arnold Murray, el amante de **Turing**, ayudó a un cómplice a entrar en la casa de **Turing** para robarle. **Turing** acudió a la policía a denunciar el delito. Durante la investigación policial, **Turing** reconoció su homosexualidad, con lo que se le imputaron los cargos de *"indecencia grave y perversión sexual"*.

Convencido de que no tenía de qué disculparse, no se defendió de los cargos y fue condenado. Dos años después del juicio, en 1954, **Turing** falleció debido a la ingestión de una manzana contaminada con cianuro en un contexto que indica un posible suicidio.

En una carta de esta época a su amigo Norman Routledge, **Turing** escribió en forma de falso silogismo una reflexión, relacionando el rechazo social que provoca la homosexualidad con el desafío intelectual que supone demostrar la posibilidad de inteligencia en los ordenadores. En particular, le preocupaba que los ataques a su persona pudieran oscurecer sus razonamientos sobre la *Inteligencia Artificial*:

- **Turing** cree que las máquinas piensan
- **Turing** yace con hombres
- Luego las máquinas no piensan

4. Tesis de Church-Turing. Importancia

En la década de 1930, uno de los problemas más estudiados por los matemáticos era el **Entscheidungsproblem** propuesto por **David Hilbert**. Por ello, se hicieron diversos intentos independientes de formalizar la noción de computabilidad:

- En 1933, el matemático **Kurt Gödel**, junto con **Jacques Herbrand**, crearon una definición formal de una *clase de funciones recursivas generales*.

- En 1936, **Alonzo Church** creó un método para definir funciones llamado λ-*cálculo*.
- En 1936, antes de aprender el trabajo de **Church**, **Alan Turing** creó un modelo teórico para las máquinas(*máquinas de **Turing***), que podría realizar cálculo a partir de la manipulación de los símbolos de una cinta.

Por otra parte, la **Tesis de Church-Turing** establece que estas tres definiciones formales de funciones computables coinciden entre ellas y con la noción informal de función "*efectivamente calculable*".

Aunque se asume como cierta, la **Tesis de Church-Turing** no puede ser probada ya que no se poseen de los medios necesarios. Esto es debido a que "*procedimiento efectivo*" y "*algoritmo*" no son conceptos dentro de ninguna teoría matemática y no son definibles fácilmente.

Se ha acordado que un procedimiento efectivo o algoritmo consiste en un número finito y preciso de pasos descrito en un número finito de símbolos que podría ser también ejecutado por un ser humano. En general, la ejecución de un algoritmo no requiere de mayor inteligencia que la necesaria para seguir las instrucciones.

Ejemplos de métodos efectivos o algoritmos abundan(suma, resta, multiplicación, división, el **algoritmo de Euclides**,...). Sin embargo, nada de esto ha sido una definición formal pues no es claro qué significa "*instrucción precisa*" o cuál es el tipo de inteligencia necesaria para seguir las instrucciones. Por esta misma razón, la idea abstracta de una máquina que funciona como parámetro para decidir cuándo algo es un algoritmo o procedimiento efectivo es de gran valor, esto es una *máquina de Turing*.

La **Tesis de Church-Turing** ha sido tan exitosa que la mayoría la supone verdadera. Los términos derivados de ella, como "*método efectivo*" y "*computable*" entre muchos otros conceptos y términos son comúnmente utilizados en la teoría de la computabilidad o funciones recursivas.

5. Tesis de Church-Turing. Variantes y detractores

Es claro que es más *"fácil"* probar la falsedad de la tesis que la verdad de la misma. Basta con exponer un método efectivo o algoritmo que no sea computable en el sentido de **Turing**(**Turing-computable**).

Aunque exponer un algoritmo que no sea **Turing-computable** no es tan fácil, pero, es más *"fácil"* que probar la verdad de la tesis, ya que parece imposible negar que sea verdadera a pesar de que eso es una posibilidad lógica.

Existe una tesis relativizada de **Church-Turing** que depende de los *grados de **Turing*** definidos por *máquinas de **Turing** con oráculos*. Los oráculos son medios formales que suponen que se le facilita cierta información a la *máquina de **Turing*** para resolver algún problema.

La **Tesis de Church-Turing** tiene además profundas implicaciones. Cuando la tesis es aplicada a la física tiene diversos significados:

- Que el universo es una *máquina de **Turing*** y, por lo tanto, no es posible construir físicamente una máquina con mayor poder computacional o que compute funciones no recursivas. A esto se le ha llamado **Tesis de Church-Turing** fuerte.

- Que el universo no es una *máquina de **Turing***, es decir, las leyes del universo no son computables pero no se puede crear una máquina más poderosa que una *máquina de **Turing***.

- Que el universo sea una hipercomputadora y entonces sea posible la construcción de máquinas más poderosas que las *máquinas de **Turing***.